



## ZADÁNÍ DIPLOMOVÉ PRÁCE

<b>Název:</b>	Návrh uživatelského rozhraní ekosystému aplikací pro události
<b>Student:</b>	Bc. Marek Jan a
<b>Vedoucí:</b>	Ing. Jiří Hunka
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Webové a softwarové inženýrství
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	Do konce letního semestru 2017/18

### Pokyny pro vypracování

Cílem práce je provést teoretický rozbor user-centered návrhu uživatelského rozhraní (UI), navrhnout UI webové a mobilní aplikace ekosystému aplikací pro události a webovou aplikaci implementovat. Ekosystém dodá vedoucí práce.

1. Proveďte teoretický rozbor user-centered návrhu UI. Vysvětlete jeho vliv na vývoj aplikace.
2. Seznamte se se zadaným ekosystémem a podobnými aplikacemi na trhu.
3. Proveďte vhodné testování UI zadaných a vybraných podobných aplikací a využijte výsledky k dalšímu návrhu.
4. Proveďte návrh UI webové a mobilní aplikace s ohledem na teoretický rozbor a cílové platformy (iOS, Android, web).
5. Proveďte vhodné testování navržených UI.
6. Implementujte webovou aplikaci pomocí vhodných technologií.
7. Proveďte testování použitelnosti implementace.
8. Zhodnoťte výsledky návrhu, implementace a testování UI. Okomentujte vliv user-centered návrhu na vaši práci.

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.  
děkan

V Praze dne 19. prosince 2016



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA SOFTWAROVÉHO INŽENÝRSTVÍ



Diplomová práce

## Návrh uživatelského rozhraní ekosystému aplikací pro události

*Bc. Marek Janča*

Vedoucí práce: Ing. Jiří Hunka

5. května 2017



---

## Poděkování

Rád bych poděkoval rodině, za neomezenou podporu po celou dobu studia, Ing. Jiřímu Hunkovi, za rady a pomoc při psaní této práce, a všem zaměstnancům a vedení FIT ČVUT, za to, že umožňují existenci skvělé fakulty jejíž součástí mohu být.



---

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mé práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 5. května 2017

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2017 Marek Janča. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Janča, Marek. *Návrh uživatelského rozhraní ekosystému aplikací pro události*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.



---

# Abstrakt

Tato diplomová práce se zaměřuje na teoretický rozbor a praktické využití návrhu uživatelského rozhraní zaměřeného na uživatele, jejich zážitků a cíle.

Cílem diplomové práce je, na základě informací získaných v teoretické části, navrhnout uživatelská rozhraní mobilní a webové aplikace zadaného ekosystému pro události a webovou aplikaci implementovat.

Práce obsahuje sjednocení pojmů uživatelsky zaměřeného návrhu, vymezení základních pravidel, začlenění do cyklu vývoje a seznámení s technikami, které lze ve fázích vývoje využít. Práce dále obsahuje seznámení se zadaným ekosystémem pro události, návrh uživatelského rozhraní, implementaci webové aplikace a testování. Vše provedené se zřetelem na pravidla vymezená v teoretické části.

Výsledkem práce jsou navržená a řádně otestovaná uživatelská rozhraní mobilní a webové aplikace a implementovaná webová aplikace. Dále je výsledkem souhrn poznatků o vlivu uživatelsky zaměřeného návrhu na proces vývoje softwaru, jeho přínosy a nevýhody a závěry z toho plynoucí.

**Klíčová slova** User Experience, uživatelsky zaměřený návrh, web, mobilní platforma, události

# Abstract

This master's thesis focuses on theoretical analysis and practical usage of user interface design that aims at users, their experience and goals.

The goal of the thesis is to design user interfaces of a mobile and web application of given event's ecosystem and to implement the web application. All done accordingly to a knowledge acquired in the theoretical part.

The thesis contains unification of terms used in user-centered design, definition of basic rules, integration into a development cycle and introduction to techniques that can be used in phases of the development cycle. The thesis also contains information about given event's ecosystem, design of user interfaces, implementation of the web application and testing. Everything done with the knowledge acquired in the theoretical part.

The results of the thesis are designed and properly tested user interfaces of the mobile and web application and implemented web application. The result is also a summary of knowledge about influence of user-centered design at the process of software development and benefits and disadvantages of this influence and conclusions arising therefrom.

**Keywords** User Experience, user-centered design, web, mobile platform, events

---

# Obsah

Úvod	1
<b>1 User Experience design</b>	<b>3</b>
1.1 Návrh zaměřený na uživatele . . . . .	3
1.2 Návrh zaměřený na cíle . . . . .	7
1.3 Psychologie uživatele . . . . .	9
1.4 Shrnutí kapitoly . . . . .	15
<b>2 User-Centered engineering</b>	<b>17</b>
2.1 Vztah UCE, UxD a Goal-Directed designu . . . . .	18
2.2 Proces vývoje . . . . .	18
2.3 Analýza . . . . .	22
2.4 Návrh . . . . .	25
2.5 Testování . . . . .	29
2.6 Shrnutí kapitoly . . . . .	33
<b>3 Ekosystém aplikací pro události</b>	<b>35</b>
3.1 Architektura ekosystému . . . . .	35
3.2 Terminologie událostí . . . . .	36
3.3 Podporovaná funkcionalita . . . . .	37
3.4 Podporované druhy aplikací . . . . .	37
<b>4 Analýza</b>	<b>39</b>
4.1 Testování současné verze klientských aplikací . . . . .	39
4.2 Průzkum a testování konkurenčních aplikací . . . . .	44
4.3 Business cíle . . . . .	47
4.4 Uživatelský průzkum . . . . .	48
4.5 Persony a jejich cíle . . . . .	51
4.6 Požadavky a omezení . . . . .	52
4.7 Shrnutí kapitoly . . . . .	54

<b>5</b>	<b>Návrh uživatelského rozhraní</b>	<b>55</b>
5.1	Navrhované funkce . . . . .	55
5.2	Návrhová rozhodnutí . . . . .	58
5.3	Shrnutí kapitoly . . . . .	78
<b>6</b>	<b>Implementace webové klientské aplikace</b>	<b>81</b>
6.1	Výběr technologií . . . . .	81
6.2	Parametrizované generování webové klientské aplikace . . . . .	90
6.3	Shrnutí kapitoly . . . . .	92
<b>7</b>	<b>Testování</b>	<b>93</b>
7.1	Testování wireframů . . . . .	93
7.2	Testování mockupů mobilní klientské aplikace . . . . .	93
7.3	Testování použitelnosti webové klientské aplikace . . . . .	97
7.4	Shrnutí kapitoly . . . . .	100
<b>8</b>	<b>Vliv uživatelsky zaměřeného návrhu</b>	<b>101</b>
8.1	Nevýhody . . . . .	101
8.2	Přínosy . . . . .	101
8.3	Zhodnocení vlivu na praktickou část práce . . . . .	102
	<b>Závěr</b>	<b>105</b>
	<b>Literatura</b>	<b>107</b>
<b>A</b>	<b>Seznam použitých zkratk</b>	<b>111</b>
<b>B</b>	<b>Uživatelský průzkum</b>	<b>113</b>
B.1	Respondent č.1 . . . . .	113
B.2	Respondent č.2 . . . . .	114
B.3	Respondent č.3 . . . . .	115
B.4	Respondent č.4 . . . . .	117
B.5	Respondent č.5 . . . . .	119
<b>C</b>	<b>Konfigurace Webpacku pro generování webových klientských aplikací</b>	<b>121</b>
<b>D</b>	<b>Testování použitelnosti</b>	<b>125</b>
D.1	Testování prototypu mobilní klientské aplikace . . . . .	125
D.2	Testování implementace webové klientské aplikace . . . . .	128
<b>E</b>	<b>Finální verze uživatelských rozhraní</b>	<b>131</b>
<b>F</b>	<b>Obsah příloženého média</b>	<b>143</b>

---

## Seznam obrázků

1.1	Forma, obsah a chování produktu definují uživatelův zážitek s produktem. . . . .	6
1.2	Designér by se měl snažit vytvořit takový návrh, aby se designerský model co nejvíce podobal uživatelskému. V takovém případě bude použití produktu pro uživatele intuitivnější a jednodušší. . . . .	10
2.1	Metodiky typu vodopád. Zdroj: [4, str. 8] . . . . .	19
2.2	Fáze procesu vývoje podle Goal-Directed designu. Zdroj: [4, str. 8] . . . . .	20
2.3	Fáze jednoduchého iterativního modelu, do kterého lze zapojit UxD. . . . .	21
2.4	Rozdíl mezi různými druhy prototypů. Zdroj: [2, str. 43] . . . . .	29
3.1	Zjednodušená architektura ekosystému . . . . .	36
4.1	Testovaná verze mobilní klientské aplikace . . . . .	40
4.2	První testovaná webová aplikace: seznam performerů . . . . .	42
4.3	První testovaná webová aplikace: časová osa . . . . .	43
4.4	Druhá testovaná webová aplikace: úvodní stránka . . . . .	43
4.5	Mobilní aplikace festivalu Colours of Ostrava. . . . .	45
4.6	Mobilní aplikace festivalu Febiofest. . . . .	46
4.7	Mobilní aplikace festivalu Rock for People. . . . .	47
5.1	Mobilní klientská aplikace: loading screen . . . . .	59
5.2	Mobilní klientská aplikace: dnešní program . . . . .	60
5.3	Mobilní klientská aplikace: časová osa . . . . .	61
5.4	Mobilní klientská aplikace: můj program . . . . .	62
5.5	Mobilní klientská aplikace: seznam performerů . . . . .	62
5.6	Mobilní klientská aplikace: detail skupiny performerů . . . . .	63
5.7	Mobilní klientská aplikace: detail performerera . . . . .	64
5.8	Mobilní klientská aplikace: novinky . . . . .	65
5.9	Mobilní klientská aplikace: detail novinky . . . . .	65

5.10	Mobilní klientská aplikace: seznam tracků . . . . .	66
5.11	Mobilní klientská aplikace: detail tracku . . . . .	67
5.12	Mobilní klientská aplikace: informace . . . . .	67
5.13	Mobilní klientská aplikace: detail informace . . . . .	68
5.14	Mobilní klientská aplikace: mapy . . . . .	69
5.15	Mobilní klientská aplikace: dotazníky . . . . .	70
5.16	Mobilní klientská aplikace: diskuze . . . . .	70
5.17	Webová klientská aplikace: domovská stránka . . . . .	74
5.18	Webová klientská aplikace: časová osa . . . . .	75
5.19	Webová klientská aplikace: seznam performerů . . . . .	76
5.20	Webová klientská aplikace: mapy . . . . .	76
5.21	Webová klientská aplikace: informace . . . . .	77
E.1	Mobilní aplikace: loading screen, dnešní program, můj program, časová osa . . . . .	132
E.2	Mobilní aplikace: seznam performerů, list performerů pop-up, skupina performerů, detail performerera . . . . .	133
E.3	Mobilní aplikace: seznam tracků, detail tracku, interaktivní mapa, obrázková mapa . . . . .	134
E.4	Mobilní aplikace: seznam novinek, detail novinky, detail novinky nastavení jména, diskuze . . . . .	135
E.5	Mobilní aplikace: dotazníky, informace, detail informace . . . . .	136
E.6	Webová aplikace: domovská stránka . . . . .	137
E.7	Webová aplikace: časová osa . . . . .	138
E.8	Webová aplikace: seznam performerů . . . . .	139
E.9	Webová aplikace: mapy . . . . .	140
E.10	Webová aplikace: informace . . . . .	141

---

# Seznam zdrojových kódů

6.1	Použití JSX společně s knihovnou React-router ve webové klient- ské aplikaci. . . . .	86
6.2	Entry soubor pro Webpack z webové klientské aplikace. . . . .	91
6.3	Konfigurační soubor obsahu webové klientské aplikace. . . . .	92
6.4	Konfigurační soubor vzhledu webové klientské aplikace. . . . .	92
C.1	Konfigurace Webpacku pro generování webových klientských aplikací. . . . .	123





---

# Úvod

Tato práce se zabývá rozborem problematiky uživatelsky zaměřeného návrhu uživatelských rozhraní a jeho vlivem na vývoj softwaru. Autor získané poznatky následně aplikuje na návrh uživatelských rozhraní mobilní a webové aplikace zadaného ekosystému pro události, uživatelské rozhraní webové aplikace také implementuje pomocí vhodných technologií a všechna navržená uživatelská rozhraní vhodně otestuje.

Cílem této práce je tedy rozbor a pochopení významu uživatelsky zaměřeného návrhu a jeho praktické použití při tvorbě uživatelských rozhraní aplikací zadaného ekosystému. Autor se v práci šířeji nezabývá inženýrskými praxemi v oblasti implementace softwaru, jelikož to není pro splnění cílů práce důležité. Implementace provedená v rámci práce je důležitá zejména pro naznačení rozdílů mezi testováním prototypu a implementací uživatelského rozhraní.

Motivací k vytvoření diplomové práce na toto téma je autorův zájem o vývoj aplikací a obecně produktů, které poskytují svým uživatelům kvalitní zážitek při jejich použití a které pomáhají plnit cíle uživatelů, protože to by měl být smysl každého kvalitního produktu. Neméně důležitou motivací je také autorův zájem o vývoj moderních aplikací pro webovou platformu.

Rozborem uživatelsky zaměřeného návrhu a jeho vlivem na vývoj softwaru se zabývají kapitoly *User Experience design* a *User-Centered engineering*. Kapitola *Ekosystém aplikací pro události* seznamuje čtenáře se zadaným ekosystémem. Kapitola *Analýza* obsahuje seznámení s konkurencí, testování stávající verze aplikací z ekosystému a konkurenčních aplikací, uživatelský průzkum a určení požadavků na aplikace pro události. Kapitola *Návrh uživatelského rozhraní* rozebírá, jakým způsobem byla uživatelská rozhraní aplikací navržena. Kapitola *Implementace webové klientské aplikace* seznamuje čtenáře s rozhodnutími ohledně výběru technologií a způsobu tvorby webových aplikací pro nové události. Kapitola *Testování* shrnuje proces testování prototypů a implementace uživatelských rozhraní aplikací. Kapitola *Vliv uživatelsky zaměřeného návrhu* nakonec shrnuje vliv uživatelsky zaměřeného návrhu na tvorbu uživatelských rozhraní aplikací ekosystému a poznatky z toho plynoucí.



---

# User Experience design

*„Everything should be made simple as possible, but no simpler.“ Albert Einstein*

*Paradox technologie - „Každý nový vynález přináší něco užitečného. Zároveň však dochází k větší komplikovanosti, a tím i naší frustraci s uživatelsky nepřívětivými výrobky.“ [1, str. 55]*

Tato kapitola se zbývá návrhem zaměřeným na uživatele, jejich zážitek s použitím navrhovaného produktu a cíle, kterých tím chtějí dosáhnout. Pojmy zde zmíněné lze aplikovat obecně, nikoliv nutně pouze v kontextu informačních technologií (IT). Produktem je myšlena jakákoliv věc vyrobená pro použití člověkem. Může jím být telefon, dveře, software, automobil, autobusová zastávka či celé město.

Návrhem není myšlen pouze návrh vzhledu produktu. Jedná se o návrh produktu jako celku, jeho formy, obsahu a chování.

## 1.1 Návrh zaměřený na uživatele

Každý nový produkt přichází na trh do konkurenčního prostředí a jeho cílem je uspět. Pro úspěch v prostředí, kde může existovat více podobných produktů, je potřeba, aby nový produkt nějakým způsobem vyčníval nad ostatními. Je nutné si uvědomit, že uživatel nakonec vždy rozhoduje o úspěšnosti či neúspěšnosti produktu, protože on jej používá. Mohou sice nastat situace, kdy bude někdo nepoužitelný produkt stejně používat, ale na to není vhodné spoléhat.

Pokud produkt poskytne uživateli adekvátní funkcionalitu, pak jej uživatel bude chtít s vysokou pravděpodobností používat. V dnešní době však může existovat až několik desítek produktů, které poskytují stejnou funkcionalitu. Proto je rovněž důležité vytvořit vztah uživatele k produktu, firmě či značce ovlivněním jeho emocí.

Pokud chceme brát ohled na uživatele, pak musíme poznat kdo jsou, jaké jsou jejich potřeby nebo jaké jsou důvody jejich chování. Pro správný návrh produktu je také důležité rozumět lidskému chování, myšlení a cítění a to nejen pokud jde o používání věcí. [1] [2] [3] [4]

**Emoce, estetika, důvtip**, který byl při návrhu produktu použit, to vše může znamenat rozdíl mezi úspěšným a neúspěšným produktem. [2, str. 5]

### 1.1.1 Emoce

Je známo, že evolučně vyspělejší organismy jsou více emocionální. Člověk je nejemocionálnější ze všech. Lidé se při použití atraktivních věcí cítí lépe a díky tomu jsou schopni přemýšlet kreativněji [3, str. 19].

Snaha o návrh cílený na vztah uživatele k produktu se zdá být dobrým směrem. Pokud vytvoříme produkt, na který bude uživatel reagovat pozitivně, jsme schopni uspokojit jeho vnitřní potřeby a vytvořit v něm takové pocity, aby náš produkt nejen rád používal, ale také byl díky němu produktivnější. Takto ovlivněný uživatel pak spíše doporučí produkt někomu dalšímu, díky čemuž může vzrůst úspěšnost produktu.

Vlivem emocí na proces návrhu produktů se zabývá Donald Norman v knize [3]. Pro lepší pochopení emocí je důležité znát strukturu a fungování lidského mozku. Ten lze rozdělit na afektivní a kognitivní systém. [3, str. 20]

**Afektivní systém** rozhoduje o tom, co je dobré a co je zlé, co je bezpečné a co je nebezpečné, obsahuje emocionální složku a kontroluje svalový systém. Prvotní reakce je vždy z afektivního systému, teprve poté přebírá řízení kognitivní systém.

**Kognitivní systém (poznání)** interpretuje svět, čímž se snaží o širší porozumění a znalosti.

Lidský mozek lze rovněž rozdělit na tři části podle chování, které zpracovávají (**Normanův třívrstvý model mozku**). Jedná se o **instinktivní vrstvu**, **vrstvu řídící chování** a **reflektivní vrstvu**. [3, str. 21]

Návrh produktu by měl cílit na všechny zmíněné vrstvy: „Uživatelův zážitek s produktem by měl být harmonickým spojením návrhu pro instinktivní vrstvu a reflektivní vrstvu se zaměřením na vrstvu řídící chování“ [4, str. 75]

#### 1.1.1.1 Instinktivní vrstva

Instinktivní vrstva provádí rychlá rozhodnutí o tom, co je dobré či špatné, co je bezpečné či nebezpečné, a následně aktivuje motorický systém a notifikuje zbytek mozku. Jedná se o automatickou část mozku. Mozek v této části reaguje na vzhled produktu a rozhoduje, zda produkt vzbudí v uživateli zájem či nikoliv.

**Zaměření návrhu** *Líbí se uživateli produkt?* Návrh cílí na tuto vrstvu se zaměřuje na aspekty, které mohou zachytit smysly. Např. na vzhled produktu. [3, str. 5] [4, str. 73]

### 1.1.1.2 Vrstva řídicí chování

Tato vrstva řídí každodenní chování. Většina lidské činnosti je prováděna v této vrstvě. Je ovlivňována reflektivní vrstvou a může ovlivňovat instinktivní vrstvu.

**Zaměření návrhu** *Jaké pocity má uživatel z používání produktu?* Návrh cílí na tuto vrstvu se zaměřuje na chování produktu. To by mělo korelovat s chování uživatele. [3, str. 5] [4, str. 73]

### 1.1.1.3 Reflektivní vrstva

Reflektivní vrstva řeší způsoby, jakými o sobě a ostatních věcech lidé přemýšlí, kdo jsou a čím chtějí být. Jedná se o vědomé uvažování a reflexe na základě zkušenosti. Je přístupná přes paměť.

**Zaměření návrhu** *Co o uživateli vypovídá, že používá produkt?* Navrhovat pro tuto vrstvu je nejobtížnější. Návrh by se měl snažit o vytvoření dlouhodobého vztahu uživatele s produktem, ale i s firmou (značkou), která produkt vyrábí. [3, str. 5] [4, str. 73]

## 1.1.2 Použitelnost

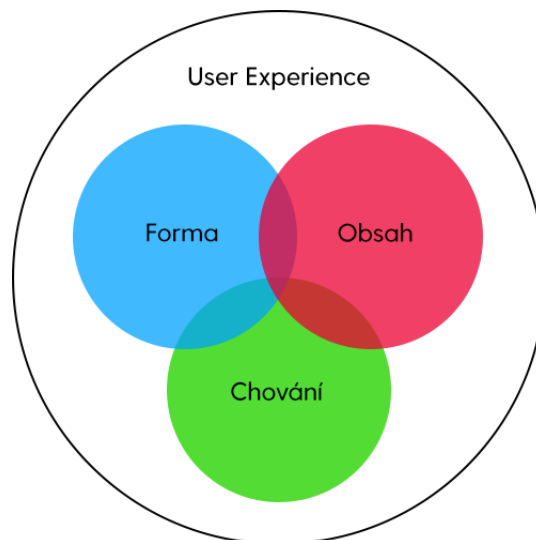
Standartní definice použitelnosti podle **ISO 9241-11**: „**Použitelnost** určuje, do jaké míry lze produkt používat k dosažení cílů efektivně, úspěšně a uspokojivě v daném kontextu použití.“ [2, str. 4]

Jsou-li emoce spojené s **afektivním systémem** mozku, pak je použitelnost spojena s **kognitivním systémem** mozku. V případě fungování mozku jsou oba systémy důležité a společně tvoří velmi mocný celek. Na návrh produktů lze pohlížet stejným způsobem. Pokud se produkt bude uživateli líbit, ale bude nepoužitelný, pak jej nebude používat. Stejně tak může být k neúspěchu odsouzen produkt, který bude použitelný, ale nebude se uživateli líbit. [3] [1]

Použitelnost všedních věcí obhajuje Donald Norman ve své knize [1]. Použitelnost bývá často používána jako kvalitativní míra pro porovnání různých návrhů. Jakým způsobem však určit, co je a co není použitelné? [2, str. 3]

Jednoduše lze říci, že produkty, které vykazují dobrou použitelnost, se lze lehce naučit používat, lze je používat efektivně a lze pomocí nich dosáhnout požadovaných cílů. Použitelnost je jedním ze základních kamenů k docílení pozitivního zážitku uživatele s použitím produktu. [2, str. 3-4]

### 1.1.3 User Experience design



Obrázek 1.1: Forma, obsah a chování produktu definují uživatelský zážitek s produktem.

Aby bylo možné správně pochopit User Experience design, je nutné nejprve definovat pojem User Experience. Dle normy **ISO 9241-210** je **User Experience (UX, uživatelský zážitek)** definováno jako „uživatelské vnímání a odezvy, které plynou z používání produktu“. Jde tedy o pohled na uživatelské vnímání a používání produktu, o zážitek při kontaktu s firmou, která produkt vytváří, a celkově o vztah uživatele k produktu [2, str. 5]. Obrázek 1.1 uvádí spojitost User Experience a oblastí, které jej ovlivňují.

**Forma:** průmyslový design, grafický design, návrh uživatelského rozhraní

**Obsah:** informační architektura, texty, animace, zvuky

**Chování:** interakční design

Každý z výše uvedených oborů patří sice svou větší částí do jedné ze zmíněných oblastí, avšak, jak lze pozorovat z obrázku 1.1, zasahují obory i do ostatních oblastí. Například interakční design může řešit také rozmístění ovládacích prvků (**forma**), ať už na fyzickém výrobku nebo na GUI<sup>1</sup>, nebo může řešit, který druh informace bude uživateli poskytnut (**informační architektura**). [4, str. xxiii]

---

<sup>1</sup>GUI = Grafické uživatelské rozhraní

**User Experience design (UxD)** je proces řízení návrhu z hlediska formy, obsahu a chování s cílem dosažení pozitivního uživatelského zážitku s používáním navrhovaného produktu.

Z definice uživatelského zážitku lze vyvozovat, že **pozitivní uživatelský zážitek** jsou pozitivní reakce uživatele na použití produktu. Chceme, aby uživatel shledával produkt použitelným, aby dokázal produkt jednoduše používat, rozuměl mu, a aby jeho emoce ve vztahu k produktu byly pozitivní.

UxD se snaží o vytvoření pozitivního vztahu uživatele k produktu tím, že umožňuje formovat a definovat uživatelské vnímání produktu zacílením nejen na funkční stránku, ale také na emocionální vztah uživatele k produktu. [4, str. xxiii]

Pokud jsme dosáhli pozitivního zážitku, pak je uživatel spokojen s produktem, je schopný jej používat, rozumí mu a produkt splňuje jeho požadavky a očekávání. To znamená, že produkt vykazuje dobrou **použitelnost**. Je nutné cílit nejen na použitelnost, ale také na **emoce** vyvolané v uživateli, na jeho spokojenost z pohledu **Normanova třívrstvého modelu**, aby bylo možné navrhovaný produkt odlišit od konkurence.

## 1.2 Návrh zaměřený na cíle

**Goal-Directed design (Návrh řízený cíly)** dovoluje porozumět uživatelským cílům, potřebám a motivacím. Zaměřuje se na důvody, proč uživatel produkt používá, jeho postoje a schopnosti. Abychom mohli vytvořit správný produkt, musíme znát uživatele a cíle, kterých chce použitím produktu dosáhnout. [4, str. xx] [4, str. 13]

Díky **Goal-Directed designu** jsem schopni odpovědět na tyto otázky:

- Kdo jsou cíloví uživatelé produktu?
- Jaké jsou cíle uživatelů?
- Jak uživatel přemýšlí o sobě a o svých cílech?
- Jakou by měl mít produkt funkcionalitu?
- Jak by se měl produkt chovat, jakou by měl mít formu a obsah?
- Jak budou uživatelé produkt používat?

[4, str. 29]

### 1.2.1 Cíle

**Cíle** jsou to očekávání nějakého výsledku a říkají, proč se uživatel chová tak, jak se chová, proč produkt používá. Díky určení cílů je možné definovat očekávanou formu, chování a obsah produktu. Informace o cílech uživatelů by se měly, pokud možno, získávat z kvalitativních dat. [4, str. 72]

### 1.2.2 Rozdíl mezi cílem a aktivitou

Jak bylo již zmíněno výše, **cíl** je očekávání nějakého výsledku. Cíl je často jen mlhavým vyjádřením toho, co uživatel chce, neimplikuje konkrétní cestu k dosažení. **Úmysl** je pak konkrétní cesta k dosažení vybraného cíle. **Aktivity** jsou mezikroky, které uživatele vedou k cíli. Aktivitu lze dále rozdělit na **úlohy** a **operace**. [1, str. 72] [4, str. 72]

**Příklad:** Cílem může být získání peněz při studiu. Úmyslem, který povede k dosažení zvoleného cíle, může být získání stipendia. Mezi aktivity patří získání dobrých známek z jednotlivých předmětů. Úlohy musíme splnit, abychom dobrých známek dosáhli a operace jsou kroky, které musíme podniknout, abychom splnili úlohy. Například pro naprogramování úkolu je nutno nejprve nastudovat příslušnou literaturu pro pochopení problému, poté napsat kód a nakonec jej otestovat na příslušných testovacích datech.

„Uživatel se zaměřuje na úlohu, kterou chce použitím produktu splnit. Designer by měl vnímat širší kontext, chápat, *kdo* jsou uživatelé a *jaké* jsou jejich cíle a *proč*.“ [4, str. 14]

### 1.2.3 Druhy cílů

Cíle lze rozdělit podle jejich vztahu k vrstvám Normanova třívrstvého modelu.

#### 1.2.3.1 Zážitekové cíle

*Jak* se chce uživatel cítit? Zážitekové cíle jsou často jednoduché a osobní. Jsou zaměřené na vizuální stránku produktu. Pokud produkt nutí uživatele, aby se cítil hloupě či vině, pak tyto cíle porušuje a výsledkem je, že uživatel nebude chtít produkt používat. [4, str. 76-77]

Pro lepší pochopení uveďme několik příkladů:

- bavit se
- být chytrý
- být soustředěný



### 1.2.3.2 Koncové cíle

Co chce uživatel dělat? Koncové cíle naznačují, z jakého důvodu uživatel produkt používá. Uživatel používá produkt pro plnění aktivit, aby těchto cílů dosáhl. Tyto cíle jsou zaměřené na chování (obsahuje i funkcionalitu) a obsah produktu. Pro jejich správné pochopení je vhodné rozumět kontextu světa, ve kterém je produkt používán, a scénářům použití. [4, str. 77-78]

Pro lepší pochopení opět uvedme několik příkladů:

- být ve spojení s rodinou, přáteli
- sledovat oblíbené seriály
- vědět o problémech ve společnosti
- dokončit dobrý obchod

### 1.2.3.3 Životní cíle

Kdo chce uživatel být? Životní cíle reflektují uživatelské osobní touhy. Pro tyto cíle je nejtěžší vytvářet návrh. Jsou to cíle mimo kontext použití produktu, přesto jsou pro návrh produktu důležité. Říkají, jaká je **motivace** uživatele k dosažení koncových cílů. Životní cíle často nezasahují přímo do procesu návrhu produktu, ale lze je transformovat do jeho komplexních, obecnějších vlastností. [4, str. 78]

Příklady těchto cílů:

- být atraktivní, zajímavý
- uspět ve svém oboru
- žít dobrý život
- být prospěšný svému okolí

## 1.3 Psychologie uživatele

Předcházející text obhajuje uživatelsky zaměřený návrh produktů a říká, jak je možné ovlivňovat uživatelské emoce pomocí **Goal-Directed designu**.

Chceme-li brát v potaz uživatele a jeho potřeby, musíme rozumět také způsobům, jak funguje kognitivní systém lidského mozku, jak se lidé učí věci používat, jak chápou, jak si pamatují a jak se chovají. Pokud budeme brát v potaz tyto poznatky, můžeme opět lépe navrhnout produkt pro lidi, pro použitelnost. Studium těchto faktorů se zabývá **kognitivní psychologie**. [5]

### 1.3.1 Mentální modely

„**Mentální modely** jsou modely lidského vnímání sama sebe, okolí a věcí, se kterými interagují.“ [1, str. 43]

Jsou to představy o fungování věcí, situací atd., které si člověk vytváří. Vznikají na základě zkušeností, instrukcí a výuky. Vznikají interpretací situací, chování nebo funkcionality. Mentální model představuje přesvědčení o funkci něčeho. Je často jednodušší než skutečnost. [1, str. 43] [6, str. 67]

#### 1.3.1.1 Konceptuální modely

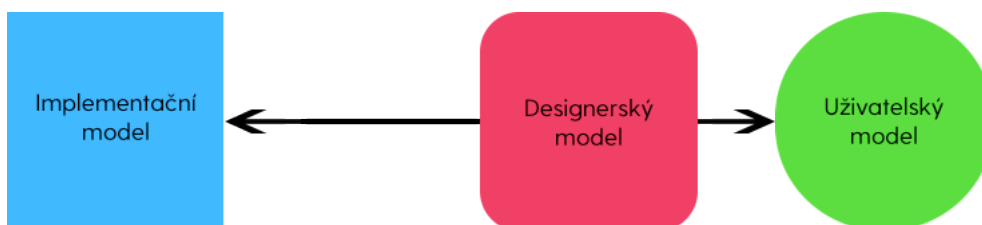
Lidé si pro vysvětlení chování či funkce nějakého předmětu představují, jak by předmět mohl fungovat. Vytvoření této představy znamená vytvoření **mentálního modelu** funkcionality. Vodítka k pochopení funkce předmětu a vytvoření správného **mentálního modelu** mohou být účelovost předmětu a omezení, vše dobře viditelné a pochopitelné.

**Konceptuální model** je uživateli poskytnut skrze návrh konkrétního produktu. Na základě tohoto modelu si pak uživatel vytvoří svůj **mentální model** [1, str. 38-39]

Pokud si uživatel vytvoří špatný **mentální model** produktu, pak je pro něj nepoužitelný a navíc v něm může vyvolávat negativní emoce. Je na designérovi, aby vytvořil takový návrh, který poskytne dobrý **konceptuální model**, pomocí něhož si uživatel vytvoří správný **mentální model** a produktu porozumí. [1, str. 39]

#### 1.3.1.2 Rozdělení modelů

Modely funkce produktu lze rozdělit podle role, kterou hrají v procesu návrhu a následného použití uživatelem. Norman v knize [1] definuje tři základní modely. Vztah mezi těmito modely znázorňuje obrázek 1.2. [1, str. 42] [6, str. 69] [4, str. 17-18]



Obrázek 1.2: Designér by se měl snažit vytvořit takový návrh, aby se designerský model co nejvíce podobal uživatelskému. V takovém případě bude použití produktu pro uživatele intuitivnější a jednodušší.

**Implementační model** (nazývaný Normanem **systemový obraz**) vyplývá z implementace, ze způsobu, kterým byl produkt vytvořen. Jedná se o **konceptuální model** funkce produktu, ten však nemusí vždy korelovat s **mentálním modelem** uživatele. Tento model může být zbytečně složitý a nemusí být uživateli nutně poskytnut.

**Designerský model** je způsob, jakým je produkt reprezentován uživateli. Jedná se o **konceptuální model** vytvořený pro zjednodušení a přiblížení fungování produktu uživateli. Měl by korelovat s uživatelským modelem. V takovém případě pak uživatel správně chápe fungování produktu. Může poskytovat špatnou představu o technickém řešení produktu, pokud tím uživateli zjednoduší pochopení produktu.

**Uživatelský model** je **mentální model**, který si vytváří uživatel na základě výše zmíněných pravidel pomocí designerského modelu a interakcemi s produktem.

**Příklad:** Rozdíly mezi modely lze prezentovat na úloze *přidání plynu v autě*. **Implementační model** moderních aut je relativně složitý. Uživatel stiskne nohou pedál, což zachytí elektronika, která vyhodnotí sílu stisku pomocí počítače, a na základě toho správně dávkuje vstřik paliva. **Designerský model** je celkem jednoduchý: „zmáčkni pedál, čím víc zmáčkneš pedál, tím rychleji pojedíš“. Uživatel tento relativně jednoduchý model přetrasformuje pomocí svých znalostí a zkušeností. Výsledkem může být **uživatelský model**, ve kterém si uživatel představuje, že stlačení plynu přímo ovlivňuje vstřik paliva do motoru. U elektromobilů je tento model úplně špatný, na druhou stranu pomáhá uživateli automobil používat efektivně.

### 1.3.2 Znalosti

Lidé fungují za pomoci dvou základních druhů znalostí: [1, str. 83]

**Deklarativní znalosti** jsou znalosti *čeho*. Jedná se o znalosti faktů a pravidel. Mezi fakta a pravidla mohou patřit například základní pravidla fungování lidské společnosti, pravidla silničního provozu nebo informace, že Země patří do Sluneční soustavy.

**Procedurální znalosti** jsou znalosti *jak*. Učí se pomocí nácviku a příkladů a jsou z velké části podvědomé. Díky nim může být člověk například dobrým hudebníkem nebo lyžařem.

**Příklad:** Rozdíl mezi výše zmíněnými druhy znalostí lze prezentovat na matematice. Základní pravidla týkající se sčítání, odčítání, násobení a dělení, komutativnosti a asociativnosti atd. jsou **deklarativní znalosti**, které si

student musí zapamatovat. Způsoby, pomocí nichž lze správně vypočítat integrál, sice vycházejí z **deklarativních znalostí** práce s integrály, ale čím víc příkladů student spočítá, tím spíš si osvojí **procedurální znalosti**, pomocí nichž je poté schopen vypočítat jakýkoli integrál.

Znalosti lze dále rozdělit podle toho, kde jsou uchovány a jak snadný k nim má člověk přístup.

### 1.3.2.1 Znalosti v hlavě (paměť)

Paměť je úložiště informací, které si člověk nosí všude s sebou. Informace zde uložené jsou však často subjektivní a závisí na způsobu, jakým člověk myslí, jak interpretoval situace, při kterých dané informace získal.

„Naše paměť má tu obecnou vlastnost, že si uchovává jen částečné popisy předmětů a událostí, které sice postačují v situaci, kdy záznam vznikl, ale které mohou selhat při použití v nové situaci.“ [1, str. 83]

Paměť lze rozdělit na dva druhy podle způsobu a doby uložení informací: [1, str. 92]

**Krátkodobá paměť** se týká právě proběhlých událostí. Ukládá informace automaticky, má omezenou kapacitu na **5-7 položek** a je neocenitelná pro každodenní život.

**Dlouhodobá paměť** uchovává minulost (zkušenosti). Neuchovává je však v podobě přesných záznamů, ale v podobě vzorců jako vlastní interpretaci proběhlých událostí (tj. zkreslené a subjektivní). Obvykle nějakou dobu trvá, než se informace zapíše do dlouhodobé paměti, což lze podpořit tréninkem. Kapacita dlouhodobé paměti není známá, je odhadována na cca. miliardu položek.

### 1.3.2.2 Znalosti ve světě

Informace obsažené ve světě si nemusí uživatel pamatovat. Jsou pro něj přístupné v situaci, kdy je potřebuje při interakci s daným produktem. Jedná se však o informace, které nemusí být uživateli nutně poskytnuty. Pokud pak nemá informace uložené v paměti, musí hádat a je odsouzen k chybování.

Je-li informace obsažena ve světě, není nutné, aby si ji uživatel pamatoval. Člověk se rozhoduje do jaké míry si má informace pamatovat, pokud jsou informace obsaženy ve světě, stačí si pamatovat potřebné minimum a zbytek nechat na dobrém návrhu. [1, str. 82]

**Příklad:** Mezi informace ve světě patří například nápisy na dveřích *sem* a *tam*. Informacemi ve světě však nemusí být nutně jen texty. Naopak, přečtení a pochopení textu vyžaduje často daleko vyšší duševní úsilí než podvědomá

reakce na návrh produktu. Pokud je návrh produktu vytvořen dobře, pak je sám o sobě informací ve světě. [1, str. 82]

### 1.3.2.3 Přesné chování z nepřesných znalostí

Jak bylo zmíněno v textu výše, není nutné, aby měl uživatel uloženy všechny znalosti v hlavě. Existují určité limity a pravidla, díky nimž je uživatel, ačkoli vybaven pouze omezenými znalostmi, schopen přesného chování: [1, str. 80-81]

1. **Informace jsou obsaženy ve světě** - toto pravidlo vychází z výše zmíněných informací. Uživatel sice nemá všechny potřebné znalosti, ty mu však poskytuje produkt svým návrhem.
2. **Není nutná dokonalá přesnost** - stačí umět pouze rozeznat správná rozhodnutí od špatných.
3. **Možnosti jsou omezeny fyzickými limity** - předměty svou strukturou a vlastnostmi omezují možnosti použití na možné minimum a tím určují svou funkcionalitu.
4. **Možnosti jsou omezeny kulturními limity** - lidská společnost si vytvořila umělé konvence, které opět omezují možnosti použití. Jedná se o výčet přijatelných možností, které si člověk musí nejprve osvojit jako deklarativní znalosti.

### 1.3.3 Chyby

Chyby jsou většinou logické, pokud jsou analyzovány s odstupem času. Ve chvíli, kdy vznikly, bylo chování vedoucí k nim zřejmé a správné. Všichni chybují a designéři by měli brát tuto skutečnost v potaz a přizpůsobit jí návrh produktů (tj. navrhovat produkty počítající s chybami).

Chyby lze rozdělit na dva základní druhy, oběma je dobré rozumět a znát způsoby, jak se jim vyhnout či jejich následky omezit na minimum. [1, str. 133, 149, 157]

**Zkrat** je běžnou chybou. Člověk chce říct jednu věc, ale řekne jinou, nebo chce udělat jednu věc a udělá jinou. Vznikají v důsledku nepozornosti při provádění osvojeného chování. **Většinou za ně může špatný návrh nebo vyrušení.**

**Omyl** je chyba vzniklá stanovením špatných cílů. Člověk může špatně odhadnout situaci a ve spojení s lidským sklonem k přílišnému zobecňování a zjednodušování vede toto chování ke vzniku omylu. Lidé místo systematické analýzy situace tíhnou k přílišnému spoléhání na paměť. **Za omyl většinou mohou neúplné či nepravdivé informace.**

Jak bylo zmíněno výše, měl by designér brát v potaz lidskou chybovost a přizpůsobit jí návrh. Zejména by měl rozumět chybám a minimalizovat možnost jejich vzniku. Měl by umožnit nápravu chyb a usnadnit odhalování vzniklých chyb. Navíc by se měl dívat na uživatele jako na osobu pokoušející se použít produkt metodou pokus omyl a brát toto chování jako přibližnou aproximaci kroků, které by měly vést ke kýženému cíli. [1, str. 157]

### 1.3.3.1 Bezradnost

Bezradnost vzniká uživatelským chybným použitím produktu, který nebere v potaz jeho omylnost a ponechává veškerou vinu za vzniklé chyby na něm. Lidé mají ve zvyku přisuzovat vzniklé chyby svým nedostatkům. Jak bylo zmíněno výše, bezradnosti se lze vyhnout správným návrhem produktu počítajícím s lidskou chybovostí. [1, 67-69]

**Bezradnost ze zkušenosti** vzniká, pokud se člověku něco opakovaně nedaří. V takovém případě usoudí, že není schopen danou věc (úkon) použít (provést) a přestane se o to pokoušet. Bezradnost ze zkušenosti může vést až ke vzniku depresí. [1, str. 69]

**Naučená bezradnost** vzniká, pokud člověk opakovaně nezvládá například použití počítače či pochopit matematické příklady. V takovou chvíli se začne považovat za nešiku v daném oboru. Tento jev však může mít svou podstatu ve špatně vytvořeném návrhu. „Člověku se něco nedaří a řekne si, že je chyba v něm. Pokud se nezdar opakuje, usoudí, že na daný úkol nestačí. Příště už to ani nezkusí. Výsledek - úkon nikdy úspěšně neprovede, přesně jak si myslel. Uvízne v pasti **sebenaplňujícího se proroctví**“ [1, str. 69]

### 1.3.4 Základní pravidla použitelného návrhu

Ze znalostí získaných díky kognitivní psychologii lze odvodit jednoduchá pravidla, která by měl mít každý designér na paměti při vytváření produktů. Jedná se o pravidla zaměřená na zvýšení použitelnosti produktů. Cílem těchto pravidel je ovlivnit uživatele tak, aby si vytvořil správný mentální model. To však není jediným cílem. Pokud jsou zmíněná pravidla dodržována, pak je práce uživatele s produktem jednodušší, instinktivnější a **uživatelský zážitek je příjemnější**.

Základní pravidla lze shrnout do několika bodů: [1, str. 213]

1. Snižte složitost produktu a prováděných úloh na nutné **minimum**.
2. Nenuťte uživatele si vše pamatovat. Využívejte **znalosti ve světě**.
3. Zajistěte, aby byly důležité prvky dostatečně **viditelné**.
4. Poskytněte uživateli správný **konceptuální model**.

5. Využijte přirozené **limity**.
6. Vždy poskytněte vhodnou **zpětnou vazbu**
7. Berte v úvahu, že uživatelé jsou **chybující** bytosti a přizpůsobte tomu návrh.
8. Využívejte **standardy**, protože je bude mít uživatel s velkou pravděpodobností dobře zažité.

## 1.4 Shrnutí kapitoly

Tato kapitola obhájí uživatelsky zaměřený návrh, který klade důraz na **zážitek uživatele (User Experience)** s použitím produktu. Pro správné aplikování **User Experience designu** je důležité chápat, že zážitek uživatele podmiňuje uživatelský vztah nejen k produktu, ale také k firmě či značce. Výsledkem by měla být snaha o vytvoření maximálně **pozitivního** zážitku.

User Experience design se snaží o dosažení pozitivního zážitku koordinací návrhu **formy**, **obsahu** a **chování** s hlavním zaměřením na uživatele. Tato snaha musí vést k vytvoření **pozitivních emocí** uživatele a **použitelného** produktu. Pro vytvoření pozitivních emocí je důležité zaměřovat proces návrhu na **cíle**, kterých se uživatel snaží použitím produktu dosáhnout, místo na konkrétní úlohy, protože ty vyplývají z cílů. Pro vytvoření použitelného produktu je nutné rozumět lidské kognici a dodržovat **základní pravidla použitelného návrhu**.





## User-Centered engineering

*Podle výzkumu, který v roce 1988 provedl Tom Gilb, vrátil každý \$1 investovaný do User Experience designu přibližně \$10–\$100. [7]*

Výstupem softwarového inženýrství je produkt - software. Výhodou vývoje softwaru oproti jiným odvětvím průmyslu je relativně jednoduchý způsob opravy chyb či vytvoření nové verze a hlavně jednoduchý způsob nasazení těchto změn do produkce. Software často automaticky či s jednoduchou uživatelskou interakcí nainstaluje svou aktualizaci a problémy jsou vyřešeny. Tak jednoduše to s nově postavenou budovou provést nelze.

Přesto však existují metodiky řídicí proces vývoje softwaru, které vycházejí z jiných odvětví průmyslu než je IT (například *Lean development*<sup>2</sup>), a proto je dle autorova mínění možné níže uvedené znalosti aplikovat i mimo IT.

Bude-li v této sekci použito slovo *software*, lze jej libovolně zaměnit za slovo *produkt*. Je však nutné si uvědomit, že obsah kapitoly vychází z teorie vývoje softwaru a je tomu přizpůsoben.

Kapitola *User Experience design* obhájí uživatelsky zaměřený návrh a vysvětluje, proč je důležité brát ohled na uživatele. Poskytuje obecné způsoby, jak brát uživatele v potaz, aby bylo docíleno lepší použitelnosti produktu, uživatelův vztah k produktu byl kladný a jeho zážitek s použitím produktu taktéž.

Uživatel, jehož požadavky na produkt nejsou splněny, jenž je demotivován nepoužitelností produktu, v němž jsou vyvolávány negativní emoce a jehož zážitek s použitím produktu je negativní, je nespokojený uživatel. Nespokojený uživatel může znamenat rozdíl mezi úspěšným a neúspěšným produktem.

Návrh produktu je součástí celého procesu vývoje produktu. **Jak tedy zapojit uživatele do procesu vývoje produktu?**

---

<sup>2</sup>Lean Development je metodika, která vznikla z Lean Manufacturing, výrobního procesu vytvořeného v automobilce Toyota, a zaměřuje se na řízení procesu vývoje tak, aby bylo plýtvání zdroji a časem omezeno na minimum. [8]

**User-Centered engineering (UCE, Inženýrství zaměřené na uživatele)**, jež je zmiňováno v knize [2], je jednou z možností jak systematicky zapojovat uživatele do vývoje, protože poskytuje techniky, které lze v průběhu vývoje využívat a tak zlepšovat uživatelský zážitek. Alan Cooper v knize [4] v rámci **Goal-Directed designu** také předkládá techniky, pomocí kterých lze zapojit uživatele do vývoje softwaru.

### 2.1 Vztah UCE, UxD a Goal-Directed designu

Abychom mohli uživatele zapojit, je důležité zodpovědět několik základních otázek: [2, str. 4]

1. Kdo jsou uživatelé produktu, jaké jsou jejich cíle a jaký je kontext, ve kterém je nebo bude produkt používán.
2. Jaký je požadovaný rozsah funkcionality produktu.
3. Jaké informace budeme uživateli poskytovat.
4. Jaké jsou procesy a procedury, ve kterých je produkt používán.

Pro získání odpovědí nejen na výše uvedené otázky existující techniky, které poskytují **User-Centered engineering** a **Goal-Directed design**. **Goal-Directed design** je **User-Centered engineering**, který navíc klade důraz na cíle uživatele.

**Pokud je při použití zmíněných technik brán ohled na myšlenky a principy, které byly uvedeny v kapitole o UxD, pak je lze považovat za techniky UxD v softwarovém inženýrství.** Pomáhají pak určovat **formu, obsah a chování** a dosáhnout dobré **použitelnosti**, kladných **emocí** uživatele a jeho pozitivního **zážitku** s použitím vyvíjeného softwaru.

### 2.2 Proces vývoje

Pro efektivní zapojení **User Experience designu**<sup>3</sup> do procesu vývoje je nutné, aby metodika<sup>4</sup>, která je pro řízení použita, toto začlenění svou strukturou umožnila. Ne každá metodika je vhodná pro kombinaci s UxD. Následující text rozebírá několik metodik a uvádí, proč jsou či nejsou vhodné pro kombinaci s UxD.

---

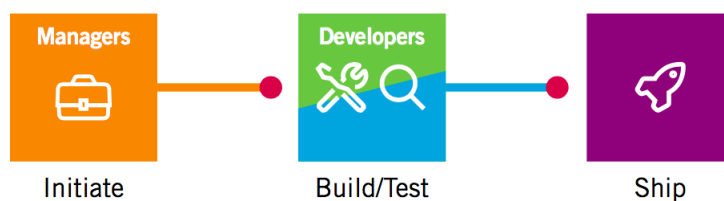
<sup>3</sup>Zapojení UxD je ekvivalentní zapojení uživatelů.

<sup>4</sup>*Metodika* obsahuje doporučení a návody, jakým způsobem řídit proces vývoje softwaru. Řeší jednotlivé fáze procesu vývoje a co se s ním provádí za aktivity. [8, str. 34]

### 2.2.1 Klasické metodiky a UxD

Mezi první metodiky, které byly použity pro vývoj softwaru, patří **Vodopádový model**<sup>5</sup>. Tato metodika sestává z několika oddělených fází. Výstup každé fáze se přenáší jako vstup do následující fáze a nelze s ním nic provádět, neexistuje zde žádná zpětná vazba, pomocí které by bylo možné výstupy opravit. [8]

Mezi fáze může patřit **analýza, implementace a testování**. Je sice možné v analýze získat data o uživateli a postoupit je do implementace pro vytvoření dobrého návrhu, který je následně implementován, ale jelikož neexistuje způsob získání zpětné vazby a možnost reakce na ni, nehodí se tato metodika pro použití s UxD.



Obrázek 2.1: Metodiky typu vodopád. Zdroj: [4, str. 8]

Tuto metodiku a jí podobné lze jednoduše popsat pomocí obrázku 2.1. Metodika očividně postrádá jakýkoli vstup od uživatele a pokud je jednou produkt nasazen do produkce, pak už jej není možné měnit.

Existují další metodiky, které částečně poskytují možnost pro začlenění uživatelů nebo poskytují *iterativní přístup*<sup>6</sup> (například **Unified Process** a jeho důraz na získání požadavků a testování), ale generují velké množství nadbytečné dokumentace, trvají dlouho a pomalu reagují na změny. [2, str. 11] [8]

### 2.2.2 Agile manifesto a UxD

Postupem času si začali vývojáři uvědomovat, že obsáhlé dokumentace, dlouhé pracovní fáze a odloučení od cílové skupiny uživatelů či dokonce od zadavatele projektu nejsou vhodným způsobem řízení vývoje v oblasti IT.

Vzniklo **Agile manifesto (Agilní manifest)**, které v sobě shrnuje několik jednoduchých myšlenek, jejichž použití zlepšuje proces vývoje softwaru. Tyto myšlenky samy o sobě dokazují, že jakákoli metodika, která z manifestu vychází, je vhodná pro začlenění UxD.

<sup>5</sup>Narozdíl od *metodiky* řeší *model* pouze fáze procesu vývoje. Obecně lze však model a metodiku považovat za to samé. [8, str. 34]

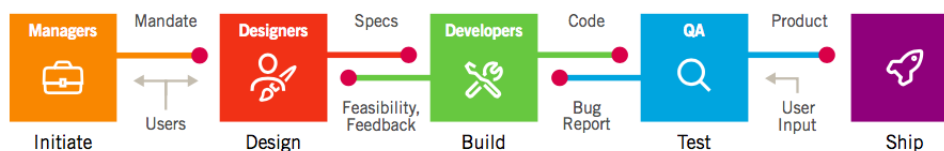
<sup>6</sup>Iterace je krátký cyklus (obvykle 2 - 4 týdny) jehož výsledkem je fungující část produktu [2, str. 13]. Obecně však lze říci, že výsledkem je otestovatelná část produktu, tedy i **uživatelsky testovatelný návrh produktu**.

- **Jednotlivci a interakce** jsou důležitější než procesy a nástroje
- **Fungující software** je důležitější než vyčerpávající dokumentace
- **Spolupráce se zákazníkem** je důležitější než vyjednávání o smlouvě
- **Reakce na změny** jsou důležitější než dodržování plánu

Agile manifesto klade důraz na začlenění uživatelů a zákazníků, na získávání zpětné vazby a na rychlé odpovědi na změnu, to vše v krátkých iteracích jejichž výsledkem je fungující software a díky nimž lze flexibilně odpovídat na požadované změny. „Agilitu zvyšuje neustálá pozornost věnovaná technické výjimečnosti a dobrému designu.“ [9]

Existuje mnoho metodik vycházejících z Agile manifesto. Tyto metodiky dodržují myšlenky uvedené v manifestu a navíc zavádějí vlastní pravidla a metody, které se v řízení procesu vývoje aplikují. Příkladem těchto metodik jsou *Scrum*<sup>7</sup>, *Extrémní programování*<sup>8</sup> nebo *SprintMethod*<sup>9</sup>.

Všechny metodiky vycházející z Agile manifesto dodržují iterativní přístup a z obecného pohledu zapojení UxD jsou důležité spíše fáze, které jsou v každé iteraci prováděny, než konkrétní pravidla vybrané metodiky. Nakonec je však samozřejmě důležité důležité rozumět konkrétní vybrané metodice, aby bylo možné UxD zapojit efektivně (například je nutné se u Scrumu přizpůsobit krátkým, čtrnáctidenním iteracím).



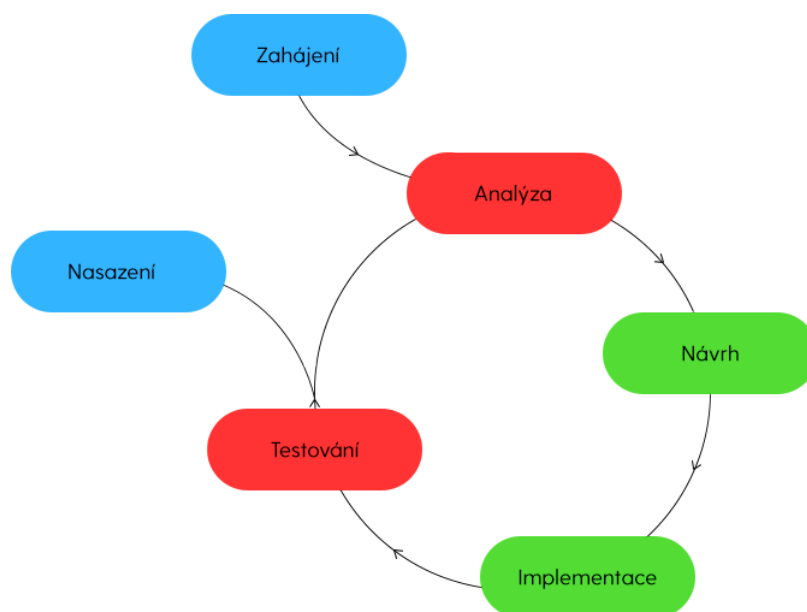
Obrázek 2.2: Fáze procesu vývoje podle Goal-Directed designu. Zdroj: [4, str. 8]

Z informací získaných v [2, str. 15-16] a [4, str. 8] lze odvodit jednoduchý iterativní model s několika fázemi, který lze obecně použít v iterativní metodice. Model vychází z fází procesu vývoje z obrázku 2.2, které poskytuje Goal-Directed design. Na obrázku 2.3 je naznačeno, jak by takový model mohl vypadat.

<sup>7</sup>Dostupné z: <https://www.scrumalliance.org/>

<sup>8</sup>Dostupné z: <http://www.extremeprogramming.org/>

<sup>9</sup>Dostupné z: <http://www.sprintmethod.cz/>



Obrázek 2.3: Fáze jednoduchého iterativního modelu, do kterého lze zapojit UxD.

Obecně lze říci, že se každá iterace skládá ze čtyř fází, kterými jsou **analýza**, **návrh**, **implementace** a **testování**. Úvodní iterace navíc obsahuje *inicializaci* a každá iterace může končit *nasazením* softwaru do produkce. Po dokončení testování a případném nasazení se opět přesouváme do analýzy, kde zkoumáme data získaná v testování a výsledky předáváme k úpravě návrhu a implementace, abychom mohli opět testovat. Ne každý krok musí být nutně prováděn v každé iteraci, například **implementace** může být v prvních iteracích vynechána a v **testování** lze testovat pouze návrh produktu. Naopak v pozdějších iteracích nemusí být nutně prováděn **návrh**. Také *inicializace* nemusí nutně předcházet **analýze**. Často bývá naopak vhodné nejprve otestovat původní produkt, pokud existuje, a poté teprve přejít k **analýze**.

Důležité je si uvědomit, že **jednotlivé fáze neimplikují práci pouze jednoho oddělení vývojového týmu**. Neznamena to, že ve fázi analýzy pracují jen analytici či ve fázi návrhu pouze designéři. Na práci ve všech fázích by se, pokud možno, měl podílet celý tým. Tímto způsobem lze jednoduše při tvorbě návrhu získat zpětnou vazbu od vývojářů v případě, že navrhované řešení není možné technicky provést. Zmíněný způsob navíc umožňuje udržovat všechny členy týmu konstatně zaměřené na vývoj produktu a na uspokojení uživatele a vytvoření jeho pozitivního zážitku. Tyto myšlenky vnáší do procesu vývoje softwaru kniha o **Lean UX**<sup>10</sup> [10].

<sup>10</sup>**Lean UX** je způsob začlenění UxD do agilních metodik a to zejména do již zmíněného **Lean developmentu**. Poskytuje principy a procesy jejichž dodržování umožní lepší směřování

Výše uvedené chování umožňuje být v blízkosti uživatelů a zákazníků, získávat od nich zpětnou vazbu a požadavky a rychle reagovat na změny. Z hlediska začlenění UxD do procesu vývoje mohou jednotlivé fáze obsahovat techniky pro zapojení uživatelů do vývoje. [2, str. 16]

**Analýza** obsahuje techniky pro prvotní získání informací o produktu, o konkurenci, o uživateli a jejich cílech a o požadované formě, obsahu a chování produktu. Umožňuje získat pohled na kontext, ve kterém bude produkt používán. V pozdějších iteracích obsahuje techniky pro analýzu zpětné vazby z testování.

**Návrh** obsahuje techniky pro lepší zaměření návrhu na uživatele. Používá pro to informace získané v analýze. Tím vším se snaží o docílení kvalitního uživatelského zážitku.

**Implementace** je konkrétní realizací vytvořeného návrhu ve vybrané technologii.

Uživatelé jsou do ní zapojeni pouze přes návrh, který je implementován. Přesto by měli mít i vývojáři na paměti blaho uživatele.

**Testování** může být trojí. Zaprvé může být testování prováděno lidmi, kteří jsou pro to speciálně vyškoleni (testeri), zadruhé může být prováděno experty v oboru a zatřetí může být prováděno vybranými uživateli. Existuje opět několik technik, jak uživatele začlenit do testování.

Jak bylo zmíněno výše, pro každý krok existují techniky známé z *User-Centered engineeringu* a *Goal-Directed designu*, které je možné aplikovat pro lepší zapojení uživatele do procesu návrhu. Na tyto techniky se lze ve většině případů dívat pomocí *Lean UX*, tj. jakým způsobem do nich zapojit celý tým vyvíjející software. Nemusí jít však jen o techniky pro přímé zapojení uživatelů, ale také o techniky, které pomáhají týmu k udržení většího zaměření na uživatele. Následující sekce obsahují konkrétní techniky, které lze v jednotlivých krocích aplikovat.

### 2.3 Analýza

Díky analýze je možné pochopit, kdo jsou uživatelé, jaké jsou jejich cíle, jakou očekávají od produktu formu, obsah či chování, nebo v jakém kontextu bude produkt používán. Analýza pomáhá stanovit **business cíle** a **cíle uživatele**, na které je nutné v průběhu návrhu brát ohled.

Pro získání takových informací je nutné ujasnit si se *zajímavými stranami* jaké jsou požavky, cíle a možnosti. Je vhodné kontaktovat *expertu v dané problematice (SME)* a získat expertní pohled na vyvíjený produkt a co

---

procesu vývoje k vysoké agilitě, k velkému zaměření na kvalitní a pozitivní uživatelský zážitek a k malému plýtvání zdroji a časem. [10]

je od něj očekáváno. Také je dobré provést *průzkum trhu* a zjistit, jaké existují podobné produkty a jak řeší možné problémy. Pokud je to možné, je vhodné provést *testování již existujícího produktu* (v případě že provádíme redesign), a to jak expertní, tak uživatelské. Díky těmto informacím je možné následně vytvořit *počáteční hypotézu o uživateli*, díky které je lze provést uživatelský průzkum na správném vzorku respondentů. [4, str. 36-41]

### 2.3.1 Uživatelský průzkum

Jednoduchým pohledem na uživatelský průzkum je jeho rozdělení dle druhu získaných dat, tedy *kvantitativní* či *kvalitativní*.

#### 2.3.1.1 Dotazníky

Kvantitativní průzkum je často prováděn marketingovým oddělením, které má přístup k velkému objemu uživatelů, a je prováděn pomocí **dotazníků**. Výsledná data je potřebné správně interpretovat, často nevedou k získání přehledu o potřebách uživatelů, ale spíše k získání velkého množství požadavků. Uživatelé nejsou často schopni správně určit, co od produktu požadují, jak jej budou používat nebo jaký produkt by si byli ochotni koupit. Je tedy na designérovi, aby byl schopný tyto informace získat, a proto je vhodnější využít následující druh uživatelského průzkumu. [4, str. 32]

#### 2.3.1.2 Etnografické rozhovory

Další možností je provést kvalitativní průzkum, který sice generuje menší objem dat, avšak je lépe zaměřen na cílovou skupinu uživatelů. Cooper ve své knize [4] představuje tzv. **Etnografické rozhovory**.

Etnografické rozhovory kombinují etnografii (tedy pozorování uživatele v jeho prostředí) s rozhovory, při kterých je uživatel dotazován pro lepší pochopení jeho chování, cílů, požadavků atd. Díky etnografickým rozhovorům je možné poznat nejen uživatele, ale také kontext, ve kterém bude produkt používán, a tak jej lépe pochopit. Rozhovory by se měly být prováděny na vzorku uživatelů určených z *hypotézy o uživateli*.

### Základní pravidla kvalitních etnografických rozhovorů

- Rozhovory provádět v prostředí uživatelů.
- Vyhnout se fixní množině otázek.
- Nestavět se do role experta, ale spíše začátečníka, hodně se ptát.
- Použít otevřené i uzavřené otázky.
- Zaměřit se hlavně na cíle, na úlohy až druhotně.

- Nedělat z uživatele designéra.
- Neřešit technologii.
- Vyhnout se sugestivním otázkám.
- Podpořit vyprávění příběhů

[4, str. 44-55]

### 2.3.1.3 Guerilla research

Pokud není možné provést tak komplexní průzkum, jaký uvádí **etnografické rozhovory** (například nejsou prostředí či pochopení od zákazníka), lze provést **Guerilla research**. Tento druh průzkumu zjednodušuje celý postup a uvádí několik kroků: [11, str. 126-128]

1. **Kdo jsou uživatelé? Co o nich víme?** Vytipovat vhodné vzorky uživatelů pro rozhovory.
2. **Jaké otázky chceme uživatelům pokládat?** Dodržovat pravidla otázek z **etnografických rozhovorů**.
3. **Navštívení uživatele:** provést rozhovory s uživateli, nejlépe v jejich prostředí.
4. **Získání informací:** zpracovat získaná data.

### 2.3.2 Shrnutí

Výsledkem uživatelského průzkumu jsou kvantitativní a kvalitativní data, díky jejichž analýze je možné pochopit kontext a doménu použití produktu. Také umožňují definovat archetypy uživatelů (persony) a cíle uživatelů a tím i požadavky na formu, chování a obsah produktu. To vše v kombinaci s získanými informacemi od zainteresovaných stran a od SME pomáhá vytvořit **produkt, který bude plnit jak cíle uživatele, tak business cíle**.

Průzkumu by se neměli účastnit pouze analytici. Podle principů, které uvádí **Lean UX** [10], by se měli průzkumu účastnit všichni členové týmu vyvíjející daný produkt. Zejména pak designéři by měli být jedni z těch, kdo provádí uživatelský průzkum, aby tak získali lepší představu o tom, pro koho navrhují [4]. Navíc by měl být průzkum prováděn opakovaně. V prvních iteracích bývají otázky obecnější, v pozdějších iteracích bývají spíše specificky zaměřené. [4]



Pro zopakování uvedme kroky, které by měly být při analýze provedeny:

1. **Získání informací od zúčastněných stran (zákazníků, vývojového týmu atd.)**
2. **Získání informací od SME**
3. **Provedení průzkum trhu pro srovnání s konkurencí.**
4. **Testování konkurence a pokud je, pak i stávající verze produktu.**
5. **Provedení uživatelského průzkumu nejlépe kvalitativními metodami. Pokud jsou data z kvantitativních metod, lze je též využít.**

## 2.4 Návrh

Informace získané při analýze je nutné nějakým způsobem zpracovat, aby bylo možné díky nim začlenit uživatele do návrhu konkrétního řešení. Ze získaných informací je možné určit požadavky na produkt, konkrétní cílovou skupinu uživatelů a jejich cíle, a tak navrhnout produkt pro plnění vhodných úloh.

Existuje mnoho technik, které lze při návrhu využít pro zlepšení výsledného uživatelského zážitku.

### 2.4.1 Persony

**Persony** jsou archetypy uživatelů, kteří jsou cílovou skupinou produktu. Vytvářejí se hlavně z kvalitativních dat získaných při **etnografickém rozhovoru**.

Personu si lze představit jako abstraktní osobu (jakéhosi průměrného uživatele), která popisuje specifickou skupinu uživatelů produktů. Taková skupina má definované své chování, svou demografii nebo své cíle. Persona je způsob, jakým definovat cíle uživatelů a tím správně navrhnout produkt pro provádění úloh. Jedná se o techniku **Goal-Directed designu**.

Při vytváření person jsou data získaná při **etnografickém rozhovoru** nejprve rozdělena *podle rolí uživatelů*. Tyto skupiny se dále rozdělí *podle proměnných popisujících chování uživatelů*. Každá proměnná může nabývat určitých hodnot a podle těchto hodnot se skupiny vytvoří. Každá taková skupina poté popisuje jednu personu. Personě je vhodné vytvořit popis, který je relevantní vůči vytvářenému produktu: demografický popis, příběh persony a fotografie pro oživení.

Tímto způsobem je možné následně určit cíle persony, které korespondují s cíli uživatelů ve skupině, kterou persona popisuje. Persony jsou také dobrou pomůckou při komunikaci se zúčastněnými stranami, pomáhají při vytváření scénářů použití produktu a také pomáhají vývojářům pro lepší zaměření na uživatele a jejich cíle. [4, str. 62-72] [4, str. 81-92] [2, str. 33-34]

**Příklad:** Automobilový průmysl vytváří různé druhy automobilů: sportovní vozy, rodinné vozy, pracovní vozy. Každý takový vůz umožňuje plnit specifické cíle. Tyto cíle si lze představit jako cíle konkrétní skupiny lidí, pro kterou je automobil určen. Jsou to tedy cíle osoby, popisující tuto skupinu lidí.

Pro lidi, kteří chtějí jezdit rychle a být moderní tak vytvoříme jiný vůz než pro stavaře. Máme jednu osobu pro milovníky rychlosti a jinou pro stavaře, známe jejich cíle a jejich příběh a jsem tak schopni automobily navrhnout lépe, abychom jim cíle pomohli plnit.

Ve výše zmíněném příkladu byl pro každou osobu vytvořen jiný produkt (jiný automobil). To ale u softwaru není nutné. Software jako jeden produkt umožňuje podporovat více osob tím, že poskytne různé druhy uživatelského rozhraní.

### 2.4.1.1 Druhy person

Persony lze rozdělit na několik druhů: [4, str. 88-90]

**Primární persona** je persona, pro kterou je navrhováno jedno uživatelské rozhraní. Produkt může mít více uživatelských rozhraní, každé pro jednu osobu. Primární osoby se od sebe výrazně liší.

**Sekundární persona** je persona, jejíž cíle a požadavky na rozhraní ve velké míře korespondují s požadavky primární osoby, proto pro ní není navrhováno vlastní uživatelské rozhraní, ale lehce upraveno rozhraní pro primární osobu.

**Non-persona** popisuje uživatele, pro které není produkt navrhován.

### 2.4.1.2 Proto-persony

**Proto-persony** představuje Leah Buley v knize [11, str. 132-135]. Jedná se o zjednodušený přístup ke tvorbě osob do situací, kdy nejsou finance či možnosti pro vytvoření tak kvalitního průzkumu, jaký je očekáván pro tvorbu klasických osob. Nejsou tak přesné, jako klasické osoby, ale lze pomocí nich dosáhnout podobných cílů.

Pro vytvoření proto-persony lze využít jakákoliv dostupná data, což může být průzkum trhu, informace od zákazníka nebo například zkušenosti kolegů.

## 2.4.2 Scénáře

**Scénáře** jsou příběhy, které realistickým způsobem popisují, jak budou uživatelé produkt používat. Mohou být vytvořeny jako seznam odrážek nebo jako krátký text. Procesy a úlohy jsou popisovány z pohledu uživatele a jeho použití produktu. Pokud jsou vytvořeny **persony**, pak je vhodné tvořit scénáře z jejich pohledu.

Scénáře jsou lehké pro porozumění a proto jsou vhodné pro komunikaci se zúčastněnými stranami. Popisují produkt z abstraktního pohledu uživatele a ne z konkrétního technologického pohledu implementace.

Pomocí scénářů lze, v kombinaci s **personami**, definovat požadavky na produkt, také pomocí nich lze vytvářet konkrétní návrhy a v neposlední řadě je možné scénáře využívat při testování navrženého produktu s uživateli. Proto jsou scénáře velmi silným pomocníkem při vytváření produktů. [2, str. 35-36]

### 2.4.3 Storyboarding

**Storyboarding** je způsob pro komunikaci mezi zákazníky, uživateli a vývojáři. Pomocí storyboardů je možné ilustrovat, jakým způsobem bude produkt fungovat.

Obecný storyboard je sekvence obrázků s popisem kontextu, situací a dialogů, které se používají ve filmové branži pro popis plánovaných záběrů. Storyboarding může mít podobu náčrtků, ale také může využívat již připravené návrhy uživatelského rozhraní produktu. Používá se tam, kde je text (**scénáře**) nedostačující. Může se vyvíjet s postupujícími iteracemi vývoje produktu. Jak již bylo zmíněno, na začátku může mít podobu náčrtků a postupem času může dostávat realistickou podobu použitím již vytvořených návrhů.

Každý storyboard obsahuje sekvenci popisující chování rozhraní. Tam, kde pouze grafický popis nestačí, obsahuje také popis situace a kontextu a chování uživatele. Storyboarding se používá pro popis použití produktu, popis průchodu uživatelským rozhraním, popis složitých konceptů atd. Jedná se o komplexnější způsob použití **person** a **scénářů** v kombinaci s vizualizací uživatelského rozhraní produktu.

### 2.4.4 Prototypy

#### 2.4.4.1 Rozdělení podle způsobu vytvoření

Nejjednodušším způsobem pro vytváření prototypů jsou **náčrtky** na papír pomocí tužky. Jsou to levné prototypy, pomocí kterých jsme schopni vizualizovat své nápady a konzultovat je s kolegy. Není je nutné vytvářet pouze na papír, ale je možné je vytvářet například na tabule pro lepší prezentaci. Jsou levné a lze je lehce měnit, což je výhodné v ranné fázi návrhu.

V pozdějších iteracích vývoje je vhodné převést náčrtky do formy, kterou lze lépe prezentovat klientům nebo uživatelům. Existují různé programy<sup>11</sup>, pomocí kterých lze vytvářet prototypy a následně je vytisknout na papír nebo je jednoduše prezentovat pomocí počítače.

Náčrtky na papír i na počítači vytvořené prototypy lze použít pro prezentaci průchodů uživatelským rozhraním. Průchod pomocí náčrtků musí si-

<sup>11</sup><https://www.uxpin.com/>, <https://www.sketchapp.com/> nebo například <https://balsamiq.com/>

mulovat člověk. Pro vytváření interaktivních prototypů v počítači lze využít speciální programy<sup>12</sup>, ve kterých je možné například navázat změnu uživatelského rozhraní na zmáčknutí specifického tlačítka.

[2, str. 42-51]

### 2.4.4.2 Rozdělení podle obsažené informace

Jedním z možných prototypů je **wireframe**. Jedná se o drátěný model uživatelského rozhraní. Nepopisuje žádné grafické či typografické vlastnosti rozhraní. Popisuje pouze rozmístění prvků v rámci uživatelského rozhraní. Díky wireframu lze odhadnout, kolik bude která část rozhraní zabírat místa. Wireframy je možné kreslit na papír jako náčrtky nebo vytvářet pomocí speciálních programů.

Dalším druhem prototypu je **mockup**. Mockup zahrnuje částečně i grafickou složku například pro oddělení důležitých částí rozhraní, různé druhy tlačítek atd. Může obsahovat reálný obsah, nemusí však obsahovat výsledný grafický návrh.

[2, str. 42-51] [11, str. 181-185]

### 2.4.4.3 Modely uživatelského rozhraní produktu

Wireframy a mockupy lze považovat za tzv. **lo-fi modely**. Tyto modely lze pomocí různých nástrojů propojit a vytvořit tak interaktivní prototypy, které vytvářejí představu o průchodu uživatelským rozhraním.

Dalším druhem modelu je tzv. **hi-fi model**. Hi-fi modely poskytují interaktivitu a je možné je dobře testovat s uživateli. Takovéto modely lze zobrazit na cílové platformě a poskytnou tak pocit opravdového funkčního produktu, nemusí však být nutně implementovány v cílové technologii.

[2, str. 42-51]

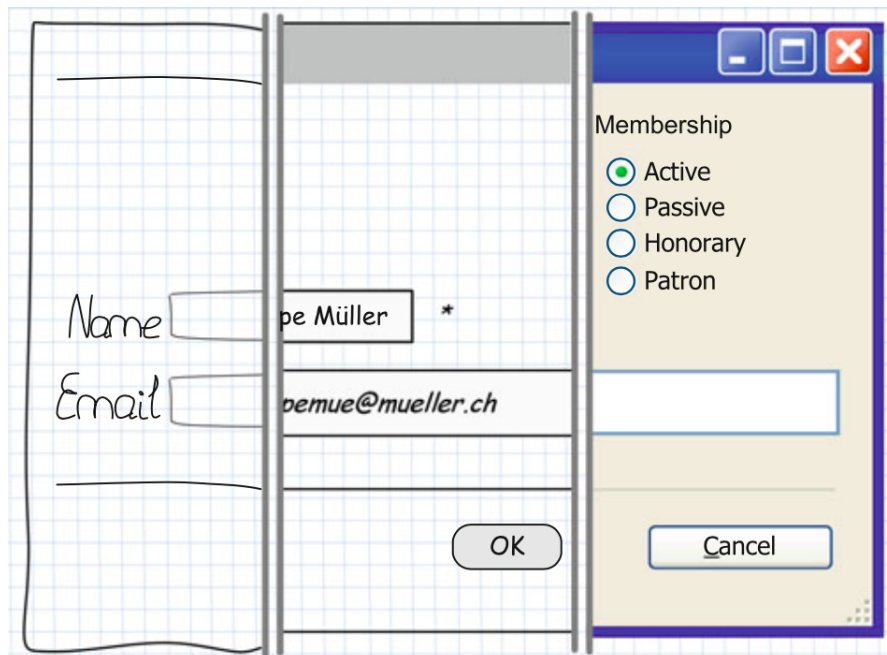
### 2.4.5 Shrnutí

Pomocí **person** a **scénářů** je možné popsat produkt, jeho formu, chování a obsah, z pohledu uživatele. Tímto způsobem lze definovat požadavky na produkt, které splňují uživatelská očekávání. Pomáhají zaměřit návrh na uživatele a optimalizovat tak uživatelské rozhraní. Také pomocí nich lze validovat přechodí verze produktu či konkurenci. [2, str. 37]

Pomocí **storyboardingu** je možné popsat chování produktu ilustrací konkrétních situací použití produktu v kombinaci se **scénáři** a **personami**. Využitím vizualizace uživatelského rozhraní je možné lépe pochopit, jak by se měl produkt chovat a jaké by měly být konkrétní průchody uživatelským rozhraním produktu. [2, str. 39-41]

---

<sup>12</sup><https://www.invisionapp.com/> nebo například <https://proto.io/>



Obrázek 2.4: Rozdíl mezi různými druhy prototypů. Zdroj: [2, str. 43]

Existují různé druhy prototypů, které se liší podle složitosti jejich vytvoření a podle situace, ve které se hodí je vytvářet. Lze je často dobře kombinovat s jinými technikami jako je například **storyboarding**. Obrázek 2.4 ilustruje rozdíl mezi různými druhy prototypů.

Výsledky všech výše uvedených technik lze použít jako specifikaci produktu, což je výhodné při použití iterativních agilních metodik, které se snaží množství dokumentace omezovat na minimum.

**Takto získanou „dokumentaci“ lze předat vývojářům pro vytvoření implementace návrhu v konkrétních technologiích pro danou platformu.**

## 2.5 Testování

V této fázi již existuje návrh produktu nebo možná i fungující produkt. Splňuje opravdu požadavky uživatele? Plní jeho cíle? Pomáhá uživateli plnit úlohy? Je použitelný? Jaké vyvolává v uživateli emoce?

Pro nalezení odpovědí na výše uvedené otázky je nutné produkt otestovat. Existuje mnoho možností, jak testovat. Nejjednodušší rozdělení je na sumativní a formativní testy. [6, str. 233][4, str. 139-141]

### 2.5.1 Sumativní testování

**Sumativní testování** je expertní testování funkčního produktu za účelem odhalení chyb a problémů, které například porušují standarty či heuristiky. Díky tomuto testování je možné odhalit známé chyby v použitelnosti či nedodržení standartů, a zlepšit tak uživatelský zážitek. Poskytuje spíše kvantitativní data. [4, str. 141]

Sumativní testování může pomáhat odhalit nedostatky při dodržení požadavků na danou platformu, tj. jestli webová aplikace splňuje požadavky na webovou platformu, jestli Android aplikace splňuje *Google Material design guidelines*<sup>13</sup>, nebo jestli iOS aplikace splňuje *iOS Human Interface guidelines*<sup>14</sup>. Mezi tyto druhy testování patří Kognitivní průchod či Heuristické vyhodnocení.

**Kognitivní průchod** je způsob expertního testování, kdy expert kontroluje produkt podle svých znalostí a zkušeností a odhaluje chyby a problémy. Může objevit obvyklé chyby nebo nedodržování standartů.

**Heuristické vyhodnocení** je způsob expertního testování, kdy expert kontroluje, jestli produkt neporušuje pravidla, která stanovuje heuristika.

#### 2.5.1.1 Nielsenova heuristika

Jednou z nejznámějších heuristik je Nielsenova heuristika. Obsahuje následující pravidla: [12]

- 1. Viditelnost stavu systému:** systém by měl vždy uživatele informovat o svém stavu, poskytovat vhodnou zpětnou vazbu.
- 2. Shoda mezi systémem a světem:** systém by měl mluvit uživatelským jazykem, dodržovat známé konvence.
- 3. Kontrola a volnost uživatele:** uživatel občas provede něco, co nechce, takže by mu mělo být umožněno provést opravu (podpora *undo* a *redo*).
- 4. Konzistence a standarty:** uživatel by měl chápat, co různá slova, situace a akce znamenají. Dodržujte platformní standarty (viz. zmíněné *guidelines*).
- 5. Prevence chyb:** lepší než chybová hláška je návrh, který vznikům chyb zabraňuje.
- 6. Rozpoznání je lepší než pamatování:** nenuťte uživatele pamatovat si věci. Zobrazte jen potřebné a relevantní informace. Dejte uživateli vizuální nápovědu.

---

<sup>13</sup>Dostupné z: <https://material.io/guidelines/>

<sup>14</sup>Dostupné z: <https://developer.apple.com/ios/human-interface-guidelines/overview/design-principles/>

7. **Flexibilita a efektivita použití:** podporujte jak začátečníky, tak experty. Zjednodušte expertům práci.
8. **Minimalismus:** dialogy by měly poskytovat pouze relevantní a důležité informace.
9. **Pomozte uživateli rozpoznat, pochopit a opravit chyby:** poskytněte chybové hlášky v přirozeném jazyce, navrhněte řešení.
10. **Nápověda a dokumentace:** i když je lepší, aby byl systém použitelný bez dokumentace, přesto ji poskytněte. Informace by měly být lehce vyhledatelné.

Je vidět, že Nielsenova heuristika pomáhá dodržovat **Základní pravidla použitelného návrhu**, je tedy vhodné brát heuristiku v úvahu již při vytváření návrhu.

## 2.5.2 Formativní testování

**Formativní testování** je způsob zapojení uživatelů do testování. Lze díky němu získat zpětnou vazbu k návrhu produktu již při jeho tvorbě. Díky tomuto testování je možné odhalit chyby, které sice neporušují heuristiky či pravidla, ale porušují uživatelův mentální model. Uživatel poskytne relevantnější informace, protože vidíme, jak bude produkt používat. Formativní testování poskytuje kvalitativní data. [4, str. 141]

### 2.5.2.1 Testování použitelnosti

Mezi nejznámější způsoby testování patří **testování použitelnosti**. Jedná se o techniku, při které je uživateli předložen produkt a on jej otestuje. Testování použitelnosti lze provádět jak na funkčním produktu, tak na různých verzích návrhu, ty k tomu však musí být přizpůsobeny. Je tak možné otestovat papírový prototyp či implementaci návrhu. [6, str. 97-112, 235-250]

Pro rigórní testování použitelnosti je nutné nejprve vybrat a domluvit termíny s uživateli, kteří budou produkt testovat. Uživatelé by měli být vybráni z cílové skupiny produktu. Následně musí být připraveny scénáře pro testování a prostředí. Scénáře mohou být konkrétní průchod produktem nebo jen navržený cíl, kdy uživatel prochází produkt podle svého rozhodnutí. Obvykle testování probíhá v pro to speciálně připravené místnosti. Může být řízeno moderátorem či je uživatel ponechán sám pro práci s produktem. Pozorovatelé jsou odděleni v jiné místnosti, například za poloprůhledným sklem, aby nemohli do testování zasahovat. Často bývá z testování pořizován zvukový a obrazový záznam.

**Vždy je nutné ubezpečit uživatele, že není testován on, ale produkt. Uživatel, který je ve stresu, může poskytnout irelevantní informace.**

[6, str. 241-250]

Obecně platí, že nejvíce informací je možné získat od pěti až sedmi uživatelů. Pokud se informace od nich neliší, pak by testování s více lidmi nepřineslo přidanou hodnotu. Testování je vhodné rozdělit do iterací a nejprve provést jedno testování se třemi uživateli, následně opravit nalezené problémy, a poté opět testovat. První iterace s vysokou pravděpodobností odhalí nejzávažnější problémy a uživatelé v dalších iteracích se jimi nebudou muset zabývat, což jim umožní odhalit problémy, které by jinak neodhalili. [6, str. 235-241] [11, str. 202-203] [13, str. 118-119]

### 2.5.2.2 Guerilla testování použitelnosti

Klasické testování použitelnosti je často časově a zdrojově náročné. Pro získání přehledu o uživatelských odezvách na návrh produktu není nutné testovat v tak rigorózních podmínkách.

Je sice vhodné testovat s uživateli, kteří patří do cílové skupiny, ale ne vždy je to možné. Proto je lepší testovat alespoň s někým, než s nikým. Stačí oslovit někoho z kolegů, kteří nemají s produktem nic společného a mají čas, aby návrh prošli a odhalili problematická místa. Také je možné oslovit někoho například v kavárně, čímž dostaneme úplně nezúčastněný pohled na návrh. [6, str. 235-241] [11, str. 202-203]

### 2.5.2.3 Pětisekundový test

Pětisekundový test je vhodnou technikou pro otestování uživatelských reakcí na produkt. Je tak možné zjistit, jestli uživatel zachytil hlavní myšlenku, jestli správně reaguje na produkt, či jaké v něm vyvolal produkt pocity. Pětisekundový test může být přibližnou aproximací použití produktu.

Produkt (například konkrétní obrazovka softwaru) je uživateli předložena na pět sekund. Poté je zakryta a uživatel je dotazován například na otázky, které byly výše zmíněny. [11, str. 204-205]

## 2.5.3 Shrnutí

Výsledky testů pomáhají odhalit chyby produktu. Získaná data je opět možné analyzovat a dostat tak nové informace, pomocí kterých je možné návrh produktu opravit. Je dobře vidět, proč je vhodné využívat iterativní metodiky. Pokud by byly odhaleny chyby a neexistoval způsob, jak je opravit, jak by produkt dopadl v konkurenci ostatních?



## 2.6 Shrnutí kapitoly

Aby bylo možné aplikovat **User Experience design** v praxi, je nutné najít způsob, jakým do inženýrského procesu (vývojového procesu) zapojit uživatele. Jak bylo naznačeno v této kapitole, nejvhodnějším způsobem zapojení UxD je využití **iterativních agilních metodik**. **User-Centered engineering** poskytuje techniky, pomocí kterých lze jednoduše zapojit uživatele do jednotlivých fází iterací. Tímto způsobem je možné pomocí vybraných technik řídit formu, chování a obsah produktu a dosáhnout tak pozitivního uživatelského zážitku. Mezi techniky, které je možné použít, patří **uživatelský průzkum, persony, scénáře** nebo **testování použitelnosti**.

Ve fázi **analýzy** je možné díky uživatelskému průzkumu zjistit, kdo jsou uživatelé a jaké jsou jejich cíle, což poskytuje potřebné informace pro vytvoření správného návrhu.

Ve fázi **návrhu** jsou využívány informace získané z analýzy, aby bylo možné pomocí **User Experience designu** řídit jednotlivé podobory a dosáhnout tak vhodně navržené formy, chování a obsahu a vytvořit pozitivní uživatelský zážitek. **Informace jsou využívány pro vytvoření vhodné informační architektury, vhodného uživatelského rozhraní či vhodného průmyslového designu.**

Ve fázi **testování** je možné spolu s uživatelem validovat návrh ať už v rané fázi pomocí prototypů nebo v pozdější fázi na konkrétním realizovaném produktu. **Tímto způsobem lze odhalovat místa produktu, která dělají uživateli problém a zlepšit tak User Experience produktu.**

Je důležité si uvědomit, že jednotlivé fáze neznamenají práci jen jednoho oddělení vývojového týmu. Všichni členové by se měli podílet na všech fázích, což pomáhá vytvářet kvalitnější návrh a udržovat všechny členy srozuměné s aktuálním děním.

**Použití User Experience designu v softwarovém inženýrství tedy znamená zapojení uživatelů do procesu vývoje softwaru takovým způsobem, aby bylo možné řídit proces tvorby formy, obsahu a chování softwaru za docílením pozitivního a kvalitního uživatelského zážitku.**



## Ekosystém aplikací pro události

*Informace zde uvedené byly získány konzultacemi ve společnosti, která ekosystém vyvíjí.*

**App4Event**<sup>15</sup> je ekosystém pro vytváření a správu aplikací pro události. Aplikace pro události (**klientské aplikace**) se připojují do infrastruktury ekosystému, do kterého vytvářejí a odkud konzumují data.

Díky způsobu, jakým je ekosystém navržen, je možné jednoduše vytvořit novou klientskou aplikaci pro událost, naplnit ji daty a vystavit ke stažení pro účastníky. Klientské aplikace podporují různé druhy funkcionality a je na pořadateli události, aby vybral, co bude v aplikaci použito.

Ekosystém umožňuje vytváření klientských aplikací pro různé platformy, díky čemuž lze pro jednu událost vytvořit klientské aplikace pro mobilní telefony, web či stolní počítač.

### 3.1 Architektura ekosystému

Centrálním prvkem je REST API<sup>16</sup> vytvořené v technologii Node.js<sup>17</sup>. REST API se dělí na klientskou a administrativní část.

**Administrativní část** umožňuje data vkládat, upravovat, mazat či číst, tedy umožňuje kompletní manipulaci s daty. Toto REST API je používáno pro plnění dat, která později konzumují klientské aplikace, a pro jeho použití je nutné se autentizovat a autorizovat.

<sup>15</sup>Dostupné z: <https://www.app4event.com/>

<sup>16</sup>**Representational State Transfer (REST)** je architektura aplikačního rozhraní (API), která využívá protokol HTTP (Hypertext Transfer Protocol) pro server-client komunikaci po internetu. [14]

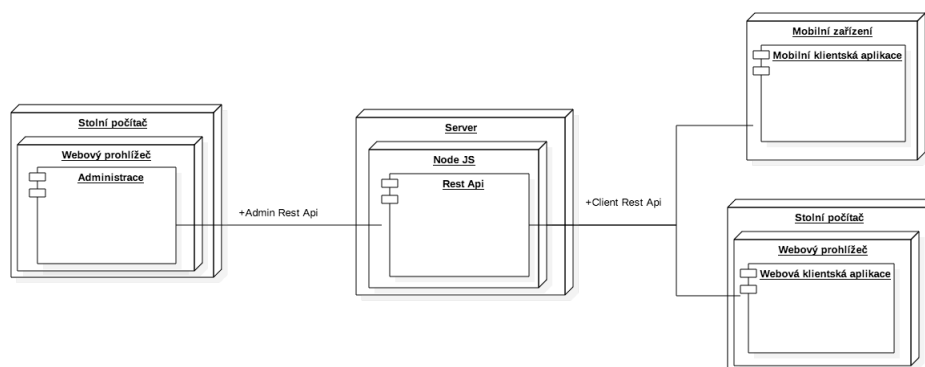
<sup>17</sup>Interpret programovacího jazyka JavaScript pro operační systémy. Více informací viz. <https://nodejs.org/>

### 3. EKOSYSTÉM APLIKACÍ PRO UDÁLOSTI

**Klientská část** umožňuje částečnou manipulaci a čtení dat. Do této části REST API se přihlašují klientské aplikace pomocí identifikátoru zařízení, který jim byl vygenerován.

Pro správu událostí a vkládání dat do ekosystému je vytvořena speciální webová aplikace (**administrace**), která využívá výše zmíněné administrativní REST API. Do administrace je pořadateli události poskytnut přístup a může pomocí ní manipulovat s daty.

Pro klientské aplikace existuje kostra obsahující veškerou funkcionalitu podporovanou pro danou platformu. Z této kostry se vytvoří nová klientská aplikace pro konkrétní událost, aktivují či deaktivují se vybrané funkce a použije se unikátní grafický návrh vytvořený na míru pro konkrétní událost. Klientské aplikace se vystaví pro stažení či použití a pomocí klientského REST API konzumují data dané události. Z pohledu uživatelského rozhraní sdílejí všechny klientské aplikace pro danou platformu jeden návrh, který je poskytnut v kostře klientské aplikace. Všechny podporované druhy událostí mají stejné uživatelské rozhraní a je snaha tuto situaci neměnit. Takto lze relativně jednoduše, rychle a bez ztráty technologické kvality vytvářet klientské aplikace s unikátním grafickým návrhem a kvalitními funkcemi vybranými na míru pro konkrétní událost. Klientskou aplikaci tedy nevytváří pořadatelé automaticky pomocí nějakého systému, ale vždy ji vytváří programátor z připravené kostry.



Obrázek 3.1: Zjednodušená architektura ekosystému

## 3.2 Terminologie událostí

Ekosystém v současné době podporuje jako události **hudební festivaly**, **filmové festivaly** a **konference**. Z hlediska každé události jsou nejdůležitější následující tři pojmy:

- **Performer** - ten nebo to, co se prezentuje na události
- **Track** - místo, kde k prezentaci dochází
- **Session** - popisuje prezentaci konkrétního performerera na konkrétním tracku ve vybraný čas

V práci budou použity tyto obecné pojmy. Pokud bude nutné, budou použity následující konkrétní pojmy:

**Filmové festivaly:** film, kino, promítání

**Hudební festivaly:** hudebník, pódium, koncert

**Konference:** přednášející, sál, přednáška

### 3.3 Podporovaná funkcionalita

Následuje výčet funkcí, které jsou v současné době v ekosystému podporovány:

- Program události jako seznam session
- Vizualizovaný program jako časová osa
- Prohlížení seznamu performerů a tracků
- Přidání konkrétních session do oblíbených
- Informování o blížícím se začátku sessions
- Čtení novinek z události
- Informace o události
- Notifikace v reálném čase (**push notifikace**)
- Interaktivní mapa s možností naplánování trasy
- Obrázkové mapy
- Vytváření dotazníků pro celou událost a pro konkrétní performery a session
- Globální diskuze a diskuze u konkrétních performerů

### 3.4 Podporované druhy aplikací

Jak bylo uvedeno výše, dovoluje ekosystém díky použitému REST API tvorbu klientských aplikací pro různé druhy platform. Ekosystém v současné době podporuje **mobilní klientské aplikace** pro platformu *Android* a *iOS* a **webové klientské aplikace**.

#### 3.4.1 Mobilní klientské aplikace

Jedná se o primární klientské aplikace, pro které byl ekosystém vytvořen. Mobilní klientské aplikace podporují veškerou funkcionalitu, které je možné se současným stavem architektury ekosystému dosáhnout. Tyto klientské aplikace jsou vytvořeny jako interaktivní program s přidanou hodnotou, kterou mobilní telefony a připojení do sítě poskytují. Aplikace se snaží poskytnout uživatelský zážitek, kterého nelze pouze s papírovými programy dosáhnout. Fungují v offline módu, kdy jsou schopny poskytovat veškerou funkcionalitu kromě té, která požaduje připojení do sítě (např. některé notifikace a diskuze nebudou offline fungovat).

#### 3.4.2 Webové klientské aplikace

Webové klientské aplikace jsou v současné době poskytovány pořadatelům zdarma a jedná se pouze o čtečky dat dané události. Jedinou přidanou hodnotou je možnost přidání session do oblíbených a čtení novinek. Měly by suplovat mobilní klientské aplikace pro jiné platformy než je Android a iOS.

**Z důvodu snahy o vytvoření kvalitního návrhu a také rozsáhlosti projektu při zaměření na všechny druhy událostí, se bude tato práce nadále zabývat pouze hudebními a filmovými festivaly.**

---

# Analýza

Předchozí kapitoly se zaměřovaly na obecný popis `User Experience designu`, jeho použití v softwarovém inženýrství pomocí technik `User-Centered engineeringu` a také na představení zadaného ekosystému pro události.

Tato a následující kapitoly odpovídají krokům cyklu obecné iterativní metodiky, která byla zmíněna v kapitole `User-Centered engineering`, a provádějí celý vývojový cyklus aplikace.

**Analýza** je jednou z nejdůležitějších fází vývoje, protože umožňuje získat informace o možnostech a požadavcích, kterými je nutné se řídit v následujících fázích. V rámci analýzy je možné provést *vyhodnocení testování předchozích verzí produktu*, získat informace od *zajímavých stran*, provést *průzkum trhu* pro srovnání s konkurencí nebo *uživatelský průzkum*. Díky takto získaným informacím je možné definovat **uživatelské a business cíle** a následně **požadavky a omezení** kladené na výsledný produkt.

Je důležité, aby se na analýze podíleli také designéři uživatelského rozhraní, protože jim přímá komunikace s cílovými uživateli pomůže lépe pochopit jejich potřeby, aby tak mohli následně vytvořit lepší návrh uživatelského rozhraní.

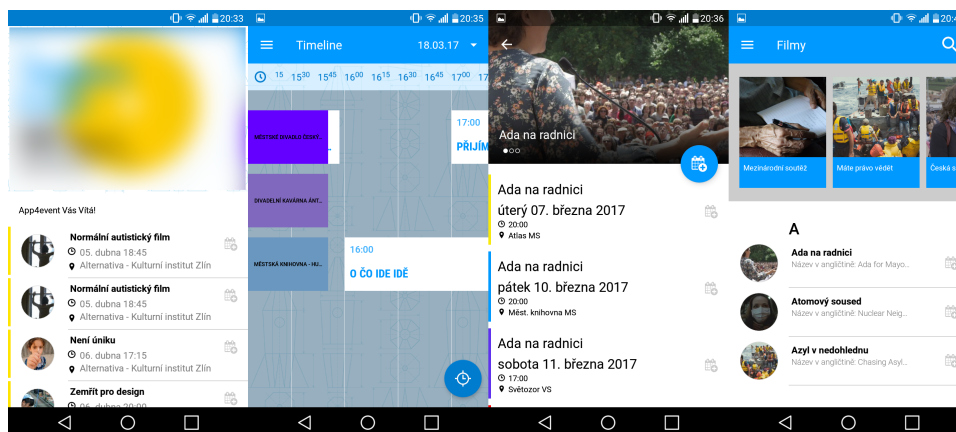
## 4.1 Testování současné verze klientských aplikací

V současné chvíli jsou ekosystémem podporovány klientské aplikace pro mobilní a webovou platformu. Z pohledu druhů událostí nejsou klientské aplikace rozdílné. Pro danou platformu sdílejí jeden návrh uživatelského rozhraní a lze je měnit vypínáním a zapínáním vybraných funkcí. Záznamy obrazovek uvedené v textu jsou jen pro ilustraci. **Všechny obrazovky testovaných aplikací z ekosystému lze nalézt na přiloženém médiu.**

## 4. ANALÝZA

### 4.1.1 Mobilní klientská aplikace

Níže uvedené problémy byly objeveny v rámci kognitivního průchodu aplikací a heuristického vyhodnocení. Aplikace byla také testována spolu s několika uživateli pomocí Guerilla testování použitelnosti a to s verzí pro pražský festival dokumentárních filmů o lidských právech (viz. obrázek 4.1) přímo v průběhu festivalu. Uživatelé poskytli svůj názor ústní cestou v průběhu festivalu a nalezené problémy byly přidány do seznamu uvedeného níže.



Obrázek 4.1: Testovaná verze mobilní klientské aplikace

Současná verze aplikace obsahuje chyby v návrhu uživatelského rohraní, kdy jsou například relativně zajímavé funkce uživateli skryty a těžce se hledají. Následuje výčet nalezených problémů (pokud problém porušuje Nielsenovu heuristiku, pak je v závorce uvedeno, který bod).

- Seznam sessions v *programu* je nepřehledný, nestrukturovaný a nelze jej filtrovat. Zobrazují se všechny session od nejaktuálnějších a probíhajících.
- Pokud nejsou žádné nadcházející session, pak je *program* prázdný a není uvedena žádná informace o tom, co se děje (Nielsen č.1).
- Aplikace uživatele po zapnutí okamžitě přesměruje do *programu* a pokud nemá data, pak se občas zobrazí pouze prázdná obrazovka s grafikou události. Neinformuje o stahování dat (Nielsen č.1).
- Pokud si uživatel přidá session do svého programu, tak jej o tom aplikace informuje pouze změnou ikony (Nielsen č.1).
- Nelze otevírat push notifikace v aplikaci. Po kliknutí na push notifikaci je uživatel přesměrován pouze do *programu* v aplikaci.



- Nelze z přímo z *časové osy* lehce skrýt vybrané tracky. Skrytí tracků lze provést až na obrazovce s jejich seznamem, na který se člověk musí proklikat skrze menu. V seznamu chybí možnost skrytí a zobrazení všech tracků a možnost filtrování.
- Při skrytí všech tracků chybí na *časové ose* informace, že se nic nezobrazuje, protože jsou všechny tracky skryty (Nielsen č.1).
- *Detail tracku* obsahuje seznam všech session, které na něm probíhají, ale seznam opět není strukturovaný a je nepřehledný.
- V seznamu performerů, pokud neexistuje pro daného performeru žádná sessions, pak se stejně zobrazuje tlačítko pro přidání jeho session do oblíbených.
- Občas se zobrazí hláška *žádná data*, ale není uvedena žádná informace, jestli se data stahují (Nielsen č.1).
- Pokud se stahuje obrázek je nahrazen placeholderem, ale není nijak prezentováno, že se opravdu stahuje a uživatel má čekat na dostahování (Nielsen č.1).
- Grafika obsahuje jako placeholder loga s názvem ekosystému, který pro koncového uživatele nemá význam a je zavádějící, vhodnější by bylo nahradit grafikou události (Nielsen č.4).
- Po skončení session uložené do oblíbených není uživateli doporučeno, aby performeru ohodnotil. Hodnocení pak postrádá smysl.
- *Program* obsahuje seznam session uložených do oblíbených. Pokud se uživatel chce dostat na plánování trasy, aby se na tuto session dostal, nelze to provést jednoduše přímo z *programu*, ale musí si najít track, kde session probíhá a přes *seznam tracků* v menu se proklikat do *detailu tracku*, kde teprve může přejít na plánování trasy.
- Pokud uživatel používá hlavně *časovou osu*, pak je nucen po zapnutí aplikace 2x kliknout, aby se dostal na obrazovku s časovou osou. Bylo by vhodnější nechat uživatele zvolit si výchozí obrazovku.
- Na obrazovce s *dotazníky* je uživatel nucen každý dotazník odesílat zvlášť. Zobrazují se dotazníky u nich bylo uzavřeno hlasování, navíc není informace o uzavření dotazníku dostatečně zvýrazněna (Nielsen č.1).
- Obrazovka *sociální sítě* obsahuje horizontální menu pouze s jednou položkou *Twitter*, což je zbytečné.
- **Android** - *časová osa* obsahuje příliš velké položky a mezery mezi tracky a při větší počtu tracků je velmi nepřehledná.

## 4. ANALÝZA

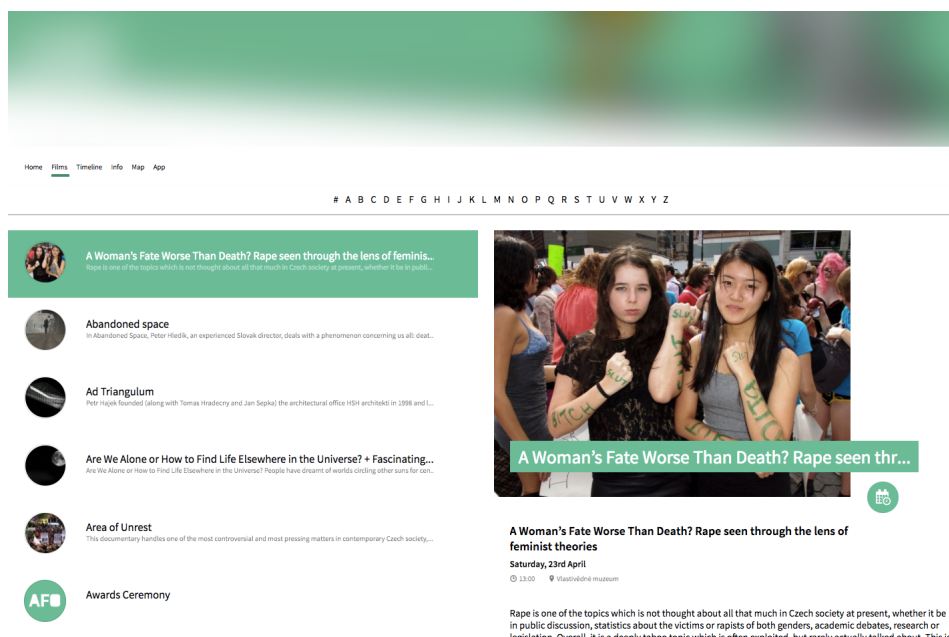
- **iOS** - po přidání session do oblíbených se ihned pošle systémová notifikace „*Session začíná za 5 minut.*“
- **Android, iOS** - občas není uživatel po kliknutí na tlačítko *zpět* vrácen na předchozí obrazovku, ale je vyhozen z aplikace.

### 4.1.2 Webová klientská aplikace

Jak bylo uvedeno výše, aplikace je v současné chvíli poskytována zdarma pořadatelům událostí a je využívána primárně jako náhrada za mobilní klientskou aplikaci pro platformy, které nejsou ekosystémem podporovány.

Při zobrazení na stolním počítači je aplikace a zejména její časová osa velmi přehledná. Pro mobilní telefony však není aplikace příliš přizpůsobena. Je zobrazena v rozlišení pro desktopy, což znamená že jsou texty na telefonu příliš malé a špatně se čtou.

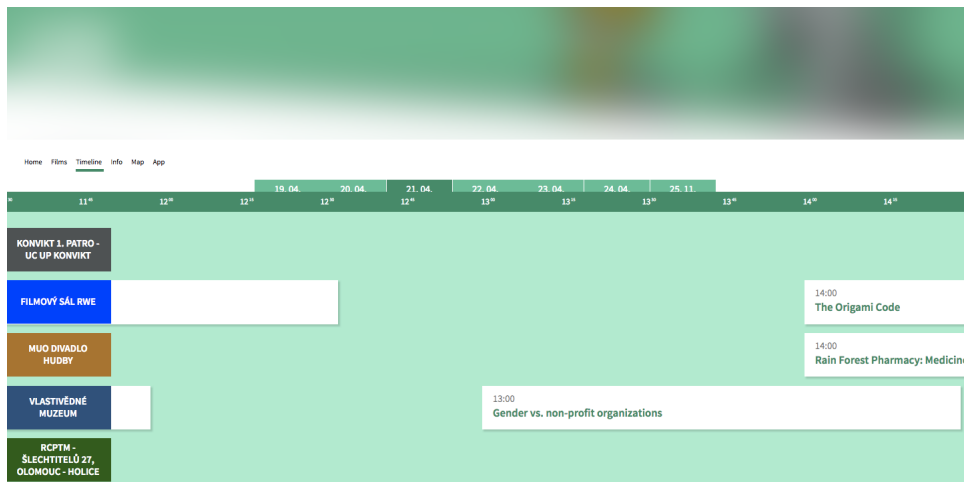
Aplikace byla testována kognitivním průchodem a heuristickým vyhodnocením na verzi pro olomoucký filmový festival a verzi pro pražskou konferenci mobilních vývojářů (viz. obrázky 4.2, 4.3 a 4.4).



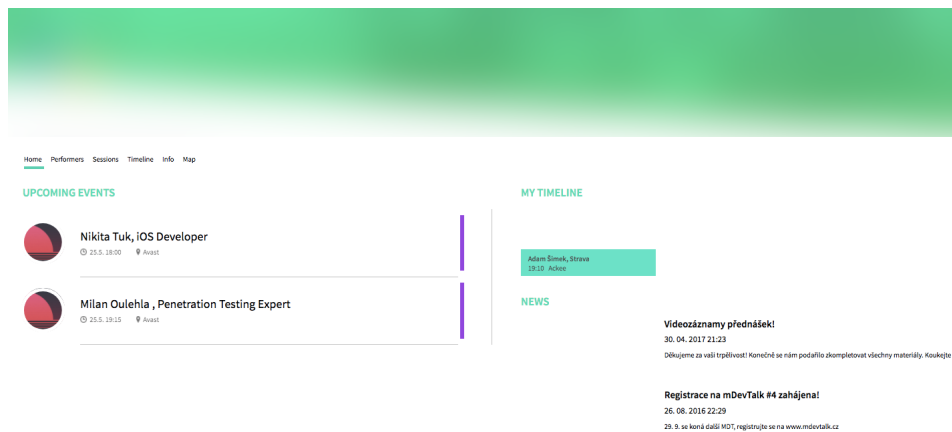
Obrázek 4.2: První testovaná webová aplikace: seznam performerů

- V aplikaci často dochází k míchání obecné terminologie s terminologií pro konkrétní druh události (Nielsen č.2 a č.4).

## 4.1. Testování současné verze klientských aplikací



Obrázek 4.3: První testovaná webová aplikace: časová osa



Obrázek 4.4: Druhá testovaná webová aplikace: úvodní stránka

- Jedná se o JavaScriptovou<sup>18</sup> aplikaci a při větším množství dat (zejména filmové festivaly) je v při použití v mobilním telefonu značně pomalá.
- Pokud neexistují nadcházející session, pak je *úvodní stránka* prázdná a neobsahuje informace, proč tomu tak je (Nielsen č.1).
- Na *úvodní stránce* se zobrazují oblíbené session i v případě, že už proběhly (Nielsen č.1).

<sup>18</sup>JavaScript je programovací jazyk, pomocí kterého lze psát aplikace fungující v prohlížeči jinak spíše dokumentově zaměřeného webu. Aplikace funguje v rámci dokumentu zobrazeného prohlížečem a modifikuje obsah dokumentu.

- *Úvodní stránka* obsahuje novinky, ty však není možné přečíst celé, jen kus textu a nic víc. Pokud nemá novinka obrázek, je pro něj vyčleněno zbytečné místo.
- Na stránce s performery dochází k duplikacím, pokud má performer více session, není tedy jasné, jestli se jedná o seznam performerů nebo session. Bylo by vhodné zobrazit performeru jednou a v detailu zobrazit seznam session, které je možné přidat do oblíbených.
- Při přidání session do oblíbených se zobrazí hláška, že se tak událo. Při přidání performeru se nic nestane. Navíc je zde oddělováno přidávání performerů a session do oblíbených, i když má větší smysl přidávat pouze session (Nielsen č.1).
- Po kliku na session v *časové ose* je uživatel přesměrován na seznam session, ten však nemusí být uveden v menu (pořadatel si to například nepřeje) a proto uživatel vypadne z kontextu aplikace.
- Na mobilním telefonu nefunguje proklik z *časové osy* na *mapu* pro info o konkrétním tracku. Při kliku na track na mapě se neotevře informace o adrese atd. (což na desktopu funguje).

**Aplikace je použitelná spíše pro procházení obsahu festivalu, čím koliduje s papírovým programem, který je narozdíl od ní dostupný offline.**

## 4.2 Průzkum a testování konkurenčních aplikací

Z hlediska návrhu uživatelského rozhraní klientských aplikací není důležité srovnávat konkurenci ekosystému jako takového, ale spíše klientských aplikací. Následuje výčet několika mobilních aplikací pro události a zajímavých rozdílů oproti řešením, která poskytují klientské aplikace ekosystému. Autorovi se nepodařilo objevit žádnou konkurenci pro webové klientské aplikace poskytované ekosystémem.

Aplikace byly testovány pomocí **kognitivního průchodu** s braným zřetel na pravidla, která stanovuje **Nielsonova heuristka**.

### 4.2.1 Colours of Ostrava

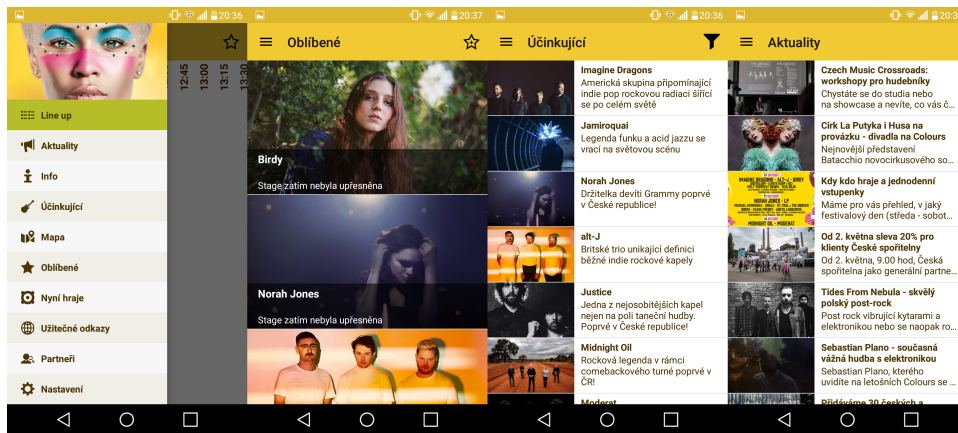
Mobilní aplikace pro hudební festival (viz. obrázek 4.5). Aplikace působí celkově jednoduchým a účinným dojmem, poskytuje kvalitní a hezký grafický návrh.

- Aplikace nemá výchozí domovskou obrazovku, ale *loading screen*<sup>19</sup>. Uživatel si může vybrat, které obrazovka bude výchozí.

---

<sup>19</sup>Obrázovka, které se zobrazuje při načítání dat do aplikace

## 4.2. Průzkum a testování konkurenčních aplikací



Obrázek 4.5: Mobilní aplikace festivalu Colours of Ostrava.

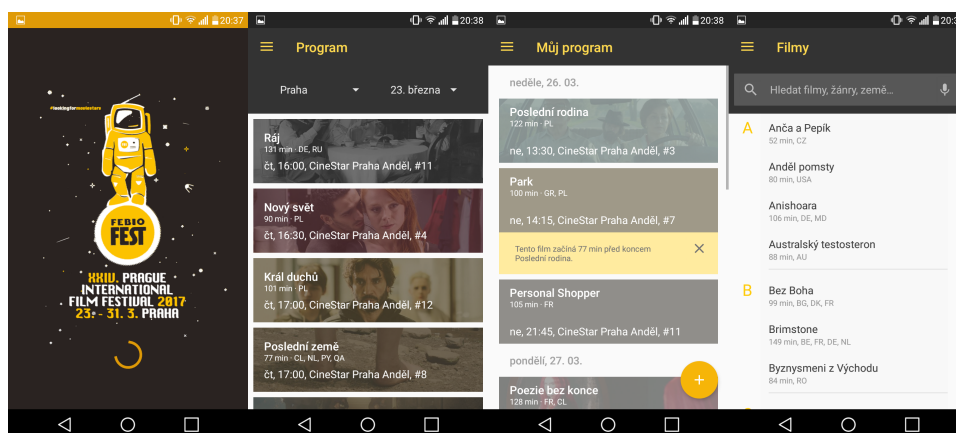
- Lze přidat hudebníky do *oblíbených* (protože každý hudebník vystupuje pouze jednou, nepřidávají se do oblíbených vystoupení).
- Aplikace poskytuje celkem komplexní možnost řazení a filtrování hudebníků.
- Samostatná obrazovka *oblíbené* obsahuje hudebníky zařazené do oblíbených s informací o jejich koncertu.
- Mapa v aplikaci je pouze obrázková, což je ale v rámci hudebního festivalu nejspíše dostačující řešení.
- Obrazovka *nyní hraje* obsahuje probíhající koncerty.
- Aplikace má samostatnou obrazovku pro partnery.

### 4.2.2 Febiofest

Mobilní aplikace pro pražský filmový festival (viz. obrázek 4.6). Aplikace působí velmi jednoduchým dojmem, což se podepsalo zejména na programu, který není přehledný kvůli nemožnosti filtrovat promítání podle kin. Aplikace neposkytuje důležité informace o festivalu, pouze novinky.

- Neobsahuje časovou osu, pouze program jako seznam promítání.
- Obrazovka *program* není přehledná, nelze zobrazovat promítání podle kin, pouze podle dní a měst.
- Lze vytvářet vlastní program, což je něco jako oblíbené.
- Na obrazovce *můj program* lze nalézt řazení podle dní a informace o překrývajících se promítáních (pouze textově, není časová osa).

## 4. ANALÝZA



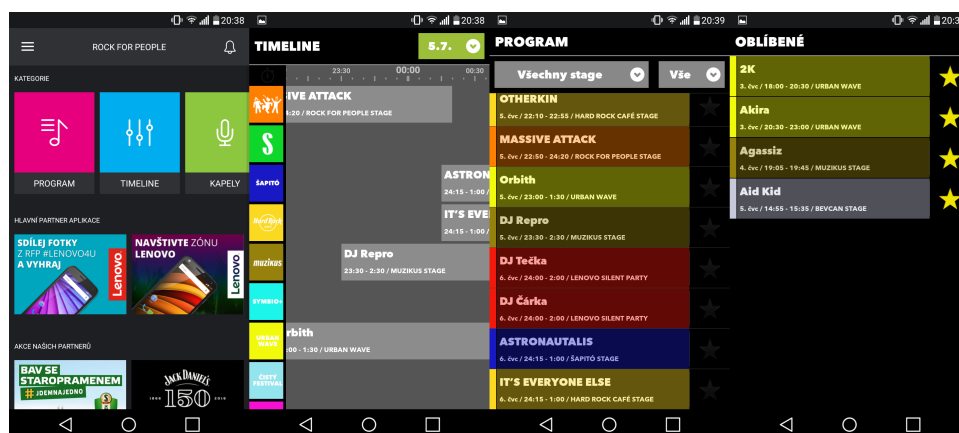
Obrázek 4.6: Mobilní aplikace festivalu Febiofest.

- Snaží se o rezervaci lístků, což však přímo z aplikace nelze provádět.
- Obsahuje aktuální *novinky* se zajímavými trailery filmů.
- Umožňuje přidat do kalendáře v mobilním telefonu promítání zařazená do *mého programu*.
- Pokud nejsou žádné novinky, pak se zobrazí pouze nadpis, ale žádný obsah, což je matoucí (Nielsen č.1).

### 4.2.3 Rock for People

Mobilní aplikace pro hudební festival v Hradci Králové (viz. obrázek 4.7). Velmi jednoduchá a přehledná aplikace, které klade důraz na program a časovou osu a prezentaci partnerů, kteří zabírají většinu úvodní obrazovky.

- *Úvodní obrazovka* obsahuje odkazy na nejdůležitější sekce a informace o partnerech festivalu.
- Nemá složité novinky, pouze obrazovku *notifikace* s krátkými zprávami pro účastníky.
- Obsahuje *program* jako seznam koncertů s možností filtrování po dnech a podiích a také obrazovku s *časovou osou* pro lepší přehlednost.
- Lze přidávat kapely do *oblíbených* a jejich koncerty jsou poté zvýrazněny na časové ose.
- Při přidání kapely do oblíbených notifikuje o přidání pouze změnou ikony.
- Obsahuje pouze obrázkovou mapu areálu festivalu.



Obrázek 4.7: Mobilní aplikace festivalu Rock for People.

#### 4.2.4 Shrnutí

Až na malé výjimky nebyly při testování nalezeny žádné problémy a všechny vybrané aplikace jsou navrženy dobře. Žádná z nich neměla větší problémy z hlediska Nielsenovy heuristiky ani kongnitivního průchodu.

### 4.3 Business cíle

Business cíle říkají, proč chce *pořadatel* pro událost vytvořit aplikaci nebo proč chce vytvářet aplikace *společnost vyvíjejí ekosystém (vývojáři)*. Návrh uživatelského rozhraní je tedy ovlivněn nejen cíli uživatelů, ale i business cíli.

#### 4.3.1 Business cíle vývojářů

Business cíle vývojářů budou mít zejména vliv na technickou stránku tvorby aplikací, ale mohou ovlivnit i návrh uživatelského rozhraní aplikací. Mezi tyto cíle může patřit:

- Vytvoření profitabilního produktu.
- Prezentování se potenciálním zákazníkům.
- Kontinuální zlepšování funkcionality a designu pro udržení náskoku před konkurencí.
- Poskytnutí kvalitního produktu za rozumnou cenu s minimálním množstvím práce.
- Pomáhání se zlepšením českých i zahraničních událostí.

### 4.3.1.1 Specifické cíle s webovou klientskou aplikací

V současnosti není webová klientská aplikace primárním produktem ekosystému a z tohoto důvodu jsou na ni kladeny specifické požadavky, které jsou generovány následujícími business cíli:

- Poskytnout zdarma jednoduchou verzi klientské aplikace a prezentovat na ní možnosti, které jsou dostupné v mobilní verzi.
- Prodat mobilní klientskou aplikaci.

### 4.3.2 Business cíle pořadatelů událostí

Business cíle pořadatelů událostí budou mít opět vliv na návrh uživatelského rozhraní aplikací. Mezi jejich business cíle může patřit:

- Snaha o inovativnost.
- Poskytování informací v reálném čase v průběhu události.
- Poskytnutí přidané hodnoty, kterou mají aplikace oproti papírovým programům.
- Získání relevantní zpětné vazby od návštěvníků.
- Pořádání kvalitní události, která uspokojí cílovou skupinu návštěvníků.

## 4.4 Uživatelský průzkum

Pro uživatelský průzkum byla vybrána technika **Guerilla research**, která zjednodušuje přístup k provádění průzkumu pomocí osobních rozhovorů s uživateli.

### 4.4.1 Určení vhodných respondentů

Potenciálními uživateli aplikací z ekosystému jsou lidé navštěvující festivaly. Taková skupina uživatelů se vyznačuje značnou demografickou diverzitou. Pokud se však zaměříme na návštěvníky festivalů, kteří často používají mobilní telefony a obecně moderní technologie, pak se lze omezit na mladé lidi ve věku od 16 do 35 let a průzkum provedný v rámci takové skupiny by měl přinést relevantní výsledky. V takovém případě lze jednoduše vybírat ze skupiny takto zaměřených studentů středních a vysokých škol.

**Pro uživatelský průzkum byli jako cílová skupina stanoveni studenti ve věku 16 - 26 let, kteří navštěvují festivaly a často používají mobilní telefony.**



#### 4.4.2 Otázky pro uživatelský průzkum

Před vlastním vedením rozhovorů je vhodné vytvořit seznam otázek, které budou uživateli pokládány. Otázky je dobré používat jako pomocníka pro situace, kdy se rozhovor nijak dále nerozvíjí. V opačném případě je vhodnější reagovat na odpovědi uživatele a rozvíjet rozhovor tímto směrem.

Otázky by měly směřovat k získání představy o cílech uživatele a motivaci a až později se zaměřit na konkrétní produkt. Následuje seznam otázek použitých při rozhovorech.

1. Máme představu o důvodech, proč festivaly navštěvujete?
2. Plánujete návštěvu festivalu dopředu nebo na poslední chvíli?
3. Vybíráte si konkrétní představení nebo chodíte náhodně?
4. Jaké informace potřebujete před navštívením festivalu?
5. Zjišťujete si informace o festivalu dopředu?
6. Chtěli byste mít možnost zapojit se do dění na festivalu?
7. Používáte na festivalech papírové programy? Co v nich hledáte nejčastěji?
8. Použili byste mobilní aplikaci festivalu?
9. Použili byste spíše mobilní nebo spíše webovou aplikaci a proč?
10. Co byste od takové aplikace pro festival očekávali?
11. Zajímala by Vás možnost hodnocení festivalu? Hodnocení performerů? Poskytnutí zpětné vazby?
12. Zajímala by Vás možnost ovlivnění festivalu?
13. Zajímala by Vás možnost socializace v aplikaci?

#### 4.4.3 Výsledky průzkumu

Průzkum byl proveden pomocí rozhovorů na vzorku pěti respondentů z vytipované cílové skupiny. Všechny pět respondentů poskytlo velmi podobné informace a z tohoto důvodu nebylo více rozhovorů provedeno. **Poznámky pořízené při průzkumu jsou uvedeny v příloze B.**

Respondenti se víceméně shodli, že z pohledu obsahu aplikace by v případě filmového i hudebního festivalu očekávali podobné informace. Jelikož je pro ně nejdůležitější navštívení konkrétních session, měla by aplikace obsahovat **program**. V případě filmového festivalu by respondenti plánovali návštěvu více

dopředu a v aplikaci by očekávali možnost vytvoření **vlastního programu**. V případě hudebního festivalu jde respondentům více o socializaci a možnost navštívení koncertů s přáteli a program přizpůsobují v průběhu festivalu aktuální situaci. V tomto případě je pro ně důležitější **časová osa**. V obou případech však nezamítají použití jak časové osy, tak vytvoření vlastního programu.

Respondenti se shodli, že by byli rádi **informováni o začátku session**, kterou si přidali do svého programu, například patnáct minut dopředu. Také by ocenili **informace o změnách v programu**, aby se jim mohli přizpůsobit. V případě hudebních festivalů by ocenili **informaci o právě probíhajících koncertech**.

Jak při návštěvě hudebního, tak při návštěvě filmového festivalu, by byli respondenti rádi informováni o různých **novinkách** z festivalu. Možnost zapojit se do **diskuze u novinek** nebo jejich případné hodnocení by taky spíše využili. Možná by využili i přímou možnost kontaktovat pořadatele v rámci chatu v aplikaci, ale nejspíše by to v aplikaci přímo nehledali.

V případě možnosti hodnocení filmů by respondenti nejspíše tuto možnost využili, ale museli by na ni být upozorněni, jinak by ji v aplikaci nehledali. Pokud by bylo navíc hodnocení využito v rámci festivalu například pro výběr nejlepšího filmu, pak by hodnocení rozhodně provedli. Pro plánování svého programu by však hodnocení nevyužili, protože plánují dopředu, kdy aplikaci ještě nemají staženou. V rámci hudebních festivalů hodnocení hudebníků pro respondenty moc smysl nemělo.

V případě hodnocení festivalu jako takového pro poskytnutí zpětné vazby by respondenti hodnocení provedli spíše v případě, že by k tomu byli nějakým způsobem motivováni například možností něco vyhrát. Tato otázka však nemá jasnou odpověď a jde spíše o povahu člověka, někdo rád pomůže nezjistně.

Respondenti se shodli, že by využili případnou možnost **naplánování trasy do kina**, kde se promítá jejich vybraný film, ale plánování trasy v případě hudebních festivalů jim moc smysl nedávalo. Zde by ocenili spíše **grafickou mapu popisující areál** festivalu. Pokud by festival nebyl v areálu, ale například na různých místech po městě, pak by plánování i na hudebním festivalu přeci jen použili.

Obecně by respondenti v aplikaci očekávali informace, které lze nalézt i v papírovém programu, ale rádi by přidanou hodnotu, kterou jim dodávají jejich mobilní zařízení. V opačném případě by aplikaci nejspíše nepoužili.

Výše uvedené informace se týkají spíše mobilní klientské aplikace. V případě webové klientské aplikace respondenti neshledávali důvod k jejímu použití, pokud by jejich mobilní platforma byla podporována mobilní klientskou aplikací. Webovou klientskou aplikaci by použili **spíše v případě plánování festivalu na počítači z pohodlí domova**, protože například je časová osa v takovém případě přehlednější a lépe se plánuje navštívení konkrétních session než v mobilním telefonu.

Rozhodně by využili **možnost vytvoření vlastního programu ve webové klientské aplikaci a následné jednoduché nahrání do mobilní klientské aplikace**. Informace o festivalu hledají na jeho webových stránkách a v takovém případě by museli být na těchto stránkách upozorněni na možnost použití webové klientské aplikace pro vytvoření vlastního programu.

**Aplikace by měla sloužit jako pomocník pro návštěvu festivalu, pro získání informací, které uživatelé použijí, aby si festival více a bez problémů užili.**

### 4.5 Persony a jejich cíle

Z informací získaných při uživatelském průzkumu a to zejména z osobních rozhovorů s uživateli lze odvodit dvě hlavní **persony**. Obě persony jsou si značně podobné, ovšem každá má své specifické potřeby. Jednu z person lze považovat za **primární** a druhou za **sekundární**, což dovoluje vytvořit jedno uživatelské rozhraní, které bude sloužit oběma personám. Díky zmíněné podobnosti jsou persony na stejné úrovni důležitosti a lze je zaměnit.

Architektura ekosystému a cíle vývojářů uvalují velké restriky na možnosti návrhu uživatelského rozhraní. Mohlo by se stát, že by obě persony byly primární, což by vedlo ke snaze vytvořit dvě rozdílná uživatelská rozhraní. To však koliduje se snahou o jednotné uživatelské rozhraní pro všechny druhy událostí. Díky tomu, že výsledné persony mohou sdílet jedno uživatelské rozhraní, se tedy situace výrazně zjednodušuje.

Pro tvorbu person byl použit zjednodušený přístup pomocí tzv. **proto-person**, který se snaží o využití všech získaných informací. Informace získané pomocí **Guerilla research** byly použity spolu s informacemi od marketingu a zúčastněných stran, tímto způsobem vznikly následující persony.

#### 4.5.1 Josef Nový

**Demografické informace** Josefovi je 28 let a již od střední školy pracuje na plný úvazek. Mezi jeho četné záliby patří mimo jiné hudba, proto často navštěvuje různé koncerty a také hudební festivaly. Je společensky aktivní, na festivaly jezdí s přáteli užít si volný čas, poslechnout si oblíbené kapely a možná objevit nové zajímavé kapely.

**Další zajímavé informace** Josef je technicky zdatný, proto nemá problém s používáním mobilního telefonu. Často využívá různé druhy aplikací, které do svého telefonu instaluje.

Josef nerad vyplňuje nějaké dotazníky, ale na druhou stranu rád dostane něco zadarmo, pokud by takový dotazník měl vyplnit.

### Cíle osoby s navštívením festivalu

- Užít si volný čas s přáteli a dobře se pobavit.
- Užít si festival bez starostí.
- Poslechnout si oblíbené kapely a objevit nové zajímavé kapely.
- Být informovaný o zajímavostech na festivalu.

#### 4.5.2 Adéla Nováková

**Demografické informace** Adéla je 23 let a studuje vysokou školu. Při škole navíc na částečný úvazek pracuje, což jí dovoluje navštěvovat různé události. Zajímá se o umění, proto navštěvuje i filmové festivaly.

**Další zajímavé informace** Adéla je společensky a kulturně aktivní. Ráda poskytne pomoc, pokud ji někdo potřebuje, nevádí jí tedy ani vyplňovat různé dotazníky.

Adéla ráda plánuje svůj volný čas, protože má velké množství povinností a potřebuje o nich mít přehled. Také navštívení filmového festivalu si plánuje předem. Adéla se nerada registruje do aplikací, které nepoužívá pravidelně. Maximálně pokud by byla registrace jednoduchá jako například přes *Facebook*.

### Cíle osoby s navštívením festivalu

- Zažít kulturní zážitek s přáteli, ale i sama.
- Shlédnout filmy, které si vybrala (naplánovala).
- Rozšířit své znalosti ohledně filmů.
- Být informovaná o novinkách v oblasti filmu.
- Zapojit se do festivalu.

## 4.6 Požadavky a omezení

**Business cíle vývojářů** uvalují na návrh uživatelského rozhraní spíše restriktive a to zejména z hlediska snahy o jednotnost uživatelského rozhraní pro všechny druhy událostí. V takovém případě je jednodušší udržovat funkcionality v rámci jedné kostry aplikace, která se sdílí přes všechny druhy událostí. Z tohoto důvodu by měla být snaha navrhnout takové rozhraní, které bude vhodné pro všechny druhy událostí a navíc bude klást důraz na funkce specifické pro jednotlivé druhy událostí.

Také snaha společnosti o svou prezentaci potenciálním zákazníkům by měla být brána v úvahu. Ostatní cíle společnosti mohou být splněny podpořením cílů pořadatelů a uživatelů.

Na webovou klientskou aplikaci jsou ze strany vývojářů pokládány specifické požadavky a restriktce z důvodu snahy prodat mobilní klientskou aplikaci, která je primárním produktem ekosystému.

Z pohledu **business** cílů pořadatelů události je kladen důraz na kvalitní prezentaci partnerů a značky festivalu a na možnost komunikace s účastníky, která přináší přidanou hodnotu jak pořadatelům (v podobě zpětné vazby od účastníků), tak účastníkům (v podobě například okamžité informovanosti o změnách).

Snaha o zlepšení přístupu k informacím o události následně vede k návrhu klientské aplikace, která bude podporovat dosažení cílů uživatelů (účastníků).

Cíle uživatelů generují nejvíce požadavků na návrh uživatelského rozhraní a měly by být z hlediska **User Experience designu** první, na co bude designér brát ohled. Bohužel však nelze brát ohled pouze na požadavky uživatelů, ale je nutné brát v potaz i požadavky vycházející z business cílů. Tyto dva druhy požadavků jdou často proti sobě a je nutné, aby designér vhodně navrhl uživatelské rozhraní ke kompromisu.

### 4.6.1 Seznam požadavků

Následující seznam obsahuje požadavky, jejichž splnění by mělo podporovat dosažení business cílů a cílů uživatelů.

- Jednotné a znovupoužitelné uživatelské rozhraní pro všechny druhy událostí.
- Snaha o donucení pořadatelů ke koupi mobilní klientské aplikace poskytnutím ve webové klientské aplikaci pouze standartních funkcí a prezentováním přidané hodnoty na mobilní klientské aplikaci.
- Mobilní klientská aplikace funkční i v případě odpojení zařízení od sítě (kromě funkcí, pro které je připojení do sítě kritické, *pozn. jedná se o požadavek, který není možné v této práci splnit, jelikož není cílem práce implementovat mobilní klientskou aplikaci*).
- Podpora prezentace společnosti vyvíjející ekosystém a pořadatelů a jejich partnerů.
- Poskytnutí možností, které přináší papírový program.
- Možnost vytvoření vlastního programu a prezentování informací z něj plynoucích.

- Možnost komunikace mezi pořadateli a účastníky pro poskytnutí zpětné vazby a důležitých informací.
- Možnost získat dodatečné informace, které při plánování události nebyly známy.
- Možnost zapojení se do průběhu události.
- Možnost naplánování návštěvy události.
- Možnost přizpůsobení aplikace uživatelem podle jeho potřeb.

**Specifické požadavky na webovou klientskou aplikaci** Ze strany strany vývojářů existují značné restriktce na webovou klientskou aplikaci z důvodu specifických business cílů.

- Omezit funkcionalitu na základní minimum, které poslouží uživateli, pokud jeho mobilní platforma není Android nebo iOS nebo pokud pořadatel nezakoupil mobilní verzi aplikace.
- Neoptimalizovat pro mobilní zařízení, ale zařídit funkčnost všech funkcí i na mobilním zařízení.

### 4.7 Shrnutí kapitoly

Výstupem fáze vývoje popsané v této kapitole je nejen všeobecné povědomí o potřebách uživatelů, kteří budou klientskou aplikaci používat, ale zejména konkrétní sada **požadavků** a **omezení**, které jsou kladeny na uživatelské rozhraní a celou klientskou aplikaci, a také **persony** a jejich cíle, které lze dále využít v procesu vývoje pro lepší zaměření na koncové uživatele. Všechny tyto informace pomáhají k zaměření návrhu klientských aplikací na uživatele a jejich potřeby a poskytnutí kvalitního uživatelského zážitku.

## Návrh uživatelského rozhraní

Fáze **návrhu** využívá informace získané v analýze pro vytvoření uživatelského rozhraní, které bude plnit jak **uživatelské cíle**, tak **business cíle** vývojářů a pořadatelů.

Již ve v kapitole *Analýza* byly vytvořeny **persony**, které pomohly lépe definovat uživatelské cíle a které lze také využít k lepšímu zaměření návrhu, protože se prezentují jako opravdoví uživatelé. Lze je také využít při např. vytváření **scénářů**. Nejdůležitější částí návrhu je vytvoření samotných návrhů uživatelského rozhraní a prototypů. V rámci práce byly vytvořeny papírové **wireframy** a interaktivní **mockupy**.

### 5.1 Navrhované funkce

Navrhované funkce lze z hlediska **User Experience designu** rozdělit do dvou kategorií:

**Primární funkce** jsou požadovány uživateli pro všechny druhy událostí.

**Sekundární funkce** nejsou nejsou nutně požadovány uživateli, ale mohou je vyžadovat pořadatelé, nebo jsou to takové funkce, které je možné se stávající architekturou ekosystému poskytnout.

**Funkce jsou odvozeny z požadavků a cílů, které by měly pomáhat uskutečnit.**

#### 5.1.1 Primární funkce

##### 5.1.1.1 Program

Chce-li klientská aplikace zastupovat papírové programy, pak musí zahrnout jejich nejdůležitější funkci a tou je program sám o sobě. Program může mít různé podoby, které jsou konkrétní vizualizací po sobě jdoucích session. Mezi tyto vizualizace může patřit časová osa nebo nějak seřazený seznam session.

Aby měl program v klientské aplikaci přidanou hodnotu, pak musí být nějakým způsobem interaktivní, tj. například zobrazovat program od aktuálního času, zvýrazňovat session, které má uživatel ve vlastním programu nebo poskytovat možnost skrýt session a tracky, které uživatele nezajímají. Také by se mohl hodit proklik na informace o session nebo na konkrétní track na mapě přímo z časové osy.

### 5.1.1.2 Vlastní program

Další přidanou hodnotou může být možnost vytvořit si vlastní program ze session, které uživatele zajímají. Díky této funkci nemusí uživatel hledat informace v seznamu všech session a má na očích jen to, co jej opravdu zajímá. Díky vlastnímu programu můžeme uživateli poskytnout další přidanou hodnotu:

- Můžeme informovat uživatele, že bude začínat nebo probíhá session, kterou má ve svém programu.
- Můžeme uživateli podle jeho programu nabízet další obsah.
- Můžeme uživateli vizualizovat jeho program v celém programu nebo podle něj celý celý program filtrovat.

### 5.1.1.3 Informace v reálném čase

Díky připojení do sítě je možné uživateli nabídnout informace, které jsou k dispozici až v průběhu události. Je možné uživatele informovat o:

- Odpadnutí performerera, kterého má ve svém programu.
- Přesunutí nebo úpravě session, kterou má ve svém programu.
- V případě nějakých potíží o postupech, které má dodržet.
- O možnosti hodnotit právě shlédnutého performerera nebo o možnosti hodnocení celé události a poskytnutí zpětné vazby.
- Přímo komunikovat v diskuzi/chatu.
- Obecně poslat push notifikaci nebo vytvořit novinku, aby byl uživatel informován.

### 5.1.1.4 Statické informace

Jedná se o informace, které by měl uživatel znát před událostí i po celou dobu konání události. Takové informace poskytují papírové programy a proto by je měla poskytovat i aplikace.



- Obecné informace o události a jejím průběhu.
- Informace o dalších možnostech, které událost poskytuje.
- Informace o restrikcích a požadavcích na návštěvníky.
- Informace o partnerech události.
- Informace o místě konání události, areálu, rozmístění tracků atd.

#### 5.1.1.5 Přizpůsobení aplikace

Pro poskytnutí dobrého uživatelského zážitku je vhodné uživateli umožnit přizpůsobení aplikace jeho konkrétním požadavkům.

### 5.1.2 Sekundární funkce

#### 5.1.2.1 Filtrování a vyhledávání obsahu

Filtrování a vyhledávání obsahu umožňuje uživateli jednoduše nalézt informace bez nutnosti procházet všechna data v klientské aplikaci. Filtrování a vyhledávání je vhodné poskytovat v každé sekci klientské aplikace, kde má alespoň nějaký smysl.

#### 5.1.2.2 Prezentace partnerů

Jedná se o jeden ze způsobů plnění business cílů pořadatelů. Události mají často mnoho partnerů na jejich podpoře stát a padá existence celé události. Je proto dobré umožnit pořadatelům prezentovat partnery události.

- Úvodní obrazovka mobilní klientské aplikace s největšími partnery.
- Domovská stránka webové klientské aplikace se sekci partnerů.
- Možnost přidání partnerů do sekce s informacemi o festivalu.

#### 5.1.2.3 Zpětná vazba

Zpětná vazba je důležitá pro zlepšování události i klientské aplikace, proto je dobré poskytnout účastníkům možnost vyjádřit se k události. Zpětnou vazbu je možné získat různými způsoby a z pohledu druhu informací, které lze získat, je možné rozdělení na:

**Kvantitativní data** - dotazníky a jejich vyhodnocení.

**Kvalitativní data** - diskuze/chat s uživateli.

Klientská aplikace může tyto funkce obsahovat, ale je také na pořadatelích, aby návštěvníkům nějakým způsobem vysvětlil, proč je potřeba zpětnou vazbu poskytnout, protože lidé většinou nemají chuť vyplňovat dotazníky.

### 5.1.2.4 Socializace

V případě novinek, které do klientské aplikace přidávají pořadatele, je možné poskytnout jednoduché sociální funkce jako je možnost komentování novinek a jejich hodnocení (například palcem nebo srdíčkem).

Také je možnost vizualizovat data ze sociálních sítí jako je *Facebook*, *Twitter* nebo *Instagram*. Zejména obrázkový obsah je pro uživatele velmi atraktivní.

## 5.2 Návrhová rozhodnutí

Výše uvedené funkce jsou obecnou představou, co by měla či mohla klientská aplikace poskytovat, a vycházejí ze stanovených požadavků. Tyto funkce se následně promítají do návrhu uživatelského rozhraní, které by mělo svým rozložením, interakcemi a obsahem zmíněné funkce zahrnout. Ne všechny funkce se přímo promítnou do uživatelského rozhraní klientské aplikace (například funkce využívající systémové notifikace), ale i tak by měly být v návrhu uvedeny. Návrh uživatelského rozhraní by měl brát v potaz všechny pravidla uvedená v kapitole o *User Experience designu*, aby tak podpořil cíle uživatelů. Nestačí tedy pouze funkce vymyslet a „nějak“ vytvořit návrh, ale naopak je nutné v návrhu brát v potaz veškeré získané informace pro vytvoření kvalitního uživatelského zážitku (zejména mít na paměti **základní pravidla použitelného návrhu**).

Uživatelské rozhraní aplikací lze rozdělit po obrazovkách (anglicky **screen**) pro mobilní klientské aplikace a po stránkách (anglicky **page**) pro webové klientské aplikace. Každá obrazovka (stránka) obsahuje specifické rozložení prvků, obsah, interakce v rámci obrazovky (stránky) a na jiné obrazovky (stránky). Následující sekce jsou rozděleny podle obrazovek (stránek), které byly pro daný typ klientské aplikace navrženy nejprve pomocí papírových **wireframů**, kdy bylo možné rychle provádět změny a ustálit záměry s návrhem, a později pomocí interaktivních **mockupů**.

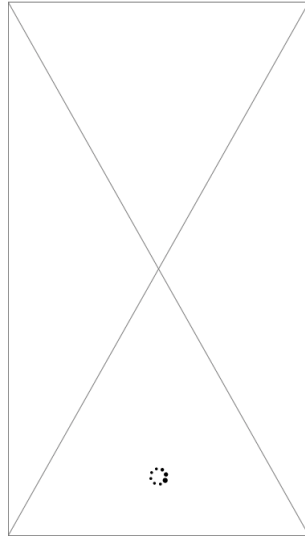
### 5.2.1 Mobilní klientská aplikace

Aplikace je primárním produktem ekosystému a proto by měla být navržena nejlepším způsobem pro dosažení business cílů a cílů uživatelů.

Do aplikace byly navrženy následující obrazovky: **loading screen**, **dnešní program**, **časová osa**, **můj program**, **performeři**, **detail skupiny performerů**, **detail performerera**, **novinky**, **detail novinky**, **tracky**, **detail tracku**, **informace**, **detail informace**, **mapy**, **dotazníky** a **diskuze**.

### 5.2.1.1 Loading screen

Obrazovka *loading screen* (viz. obrázek 5.1) byla navržena pro prezentaci grafiky události a jejích partnerů. Uživatelé ji uvidí jako první po spuštění aplikace po dobu stahování dat důležitých pro běh aplikace. Z tohoto důvodu je vhodné na ni umístit grafiku a sponzory a tím podpořit business cíle pořadatelů.



Obrázek 5.1: Mobilní klientská aplikace: loading screen

### 5.2.1.2 Dnešní program

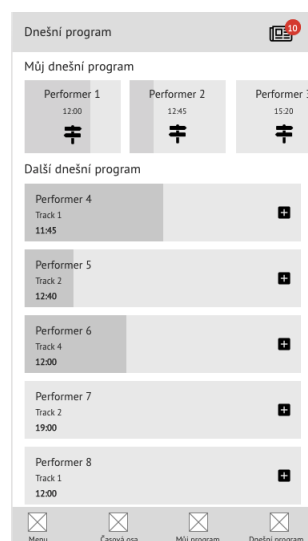
Obrazovka *dnešní program* (viz. obrázek 5.2) je výchozí obrazovka, na kterou je uživatel přesměrován z *loading screen*.

Obrazovka obsahuje dvě sekce. První sekce je seznam dnešních session z *mého programu* seřazených podle času začátku session od nejbližších. Každá položka obsahuje vizualizaci v případě, že už session začala. Také obsahuje odkaz na plánování trasy na track, kde session probíhá, pokud má track přiřazený pin<sup>20</sup>. Po kliknutí na položku je uživatel přesměrován na *detail performeru*.

Druhá sekce obsahuje seznam všech dnešních session seřazených opět podle času začátku od nejbližších a každá položka opět obsahuje vizualizaci, pokud session již probíhá, a proklik na *detail performeru*. Navíc lze každou položku ze seznamu přidat do *mého programu*.

V horní liště obrazovky je přidán proklik na *novinky* s počtem nepřečtených novinek.

<sup>20</sup>Bod na mapě vizualizující adresu.



Obrázek 5.2: Mobilní klientská aplikace: dnešní program

### 5.2.1.3 Časová osa

*Časová osa* (viz. obrázek 5.3) je důležitá jak pro plánování, tak pro orientaci v průběhu události. Je časteji využívána návštěvníky hudebních festivalů, kteří se díky ní mohou orientovat v jednotlivých koncertech a jakým způsobem se překrývají a díky tomu upravovat své plány. Také ji využijí návštěvníci filmových festivalů pro naplánování svého programu a obecně je vhodná pro lepší orientaci v programu události.

Z výše uvedených důvodů byl zvolen přístup, který používá například *Google Calendar* nebo *Kalendář v MacOS* k zobrazení po dnech, a na který je mnoho uživatelů zvyklých.

*Časová osa* je zobrazována vertikálně po sloupcích. Úplně první sloupec je úžší a zobrazuje osu s časem rozdělenou po patnácti minutách. Každý další sloupec má v hlavičce uveden název tracku a zobrazuje časovou posloupnost všech session, které na něm probíhají. *Časovou osou* lze pohybovat ve vertikálním směru pro průchod programem v čase a horizontálním směru pro průchod programem přes tracky. Tímto způsobem jsou vedle sebe zobrazeny průběhy všech session na všech trackech.

Při kliknutí na session v *časové ose* je uživatel přesměrován na *detail performerera*, který se na dané session prezentuje. Při kliknutí na track je uživatel přesměrován na *detail tracku*. Navíc je přímo v ose u každého tracku možnost jeho skrytí. Tímto způsobem může uživatel filtrovat obsah *časové osy*.

Další možnost filtrování poskytují dva filtry v hlavičce obrazovky. Pomocí jednoho lze přesunout *časovou osu* na začátek vybraného dne a pomocí druhého lze zobrazit pouze tracky z daného města, což se hodí pro události, které probíhají ve více městech. Také bylo přidáno tlačítko na zobrazení pouze

session, které si uživatel přidal do *svého programu*. Tyto session jsou navíc v *časové ose* zvýrazněny. *Osa* také obsahuje tlačítko, které přesune uživatele na aktuální čas.



Obrázek 5.3: Mobilní klientská aplikace: časová osa

#### 5.2.1.4 Můj program

*Můj program* (viz. obrázek 5.4) obsahuje seznam session, které si do něj uživatel přidal. Seznam je rozdělen do sekcí podle tracků, které mají session přiřazeny. Seznam session patřící k vybranému tracku je možné skrýt či zobrazit pro lepší přehled v programu. Každá položka v programu obsahuje tlačítko pro odebrání session z *mého programu* a vizualizaci, pokud session probíhá. Přímo z programu je také možné přejít na plánování trasy na vybraný track.

#### 5.2.1.5 Seznam performerů

*Seznam performerů* (viz. obrázek 5.5) je rozdělen na dvě sekce. První sekce obsahuje horizontální seznam skupin performerů s jejich obrázkem a názvem. Po kliknutí na položku v seznamu je uživatel přesměrován na *detail skupiny performerů*.

Druhá sekce obsahuje abecedně seřazený seznam performerů. Každá položka obsahuje obrázek performerera, jeho název, část popisu a tlačítko pro přidání performerera do *mého programu*. Po kliknutí na položku je uživatel přepnut na *detail performerera*. Po kliknutí na tlačítko pro přidání do *mého programu* je zobrazen pop-up<sup>21</sup> s možností rychlého výběru konkrétní session, kterou chce

<sup>21</sup>Zobrazí se okno překrývající část původní obrazovky, které je možné opět zavřít, a uživatel není přepínán mezi obrazovkami.

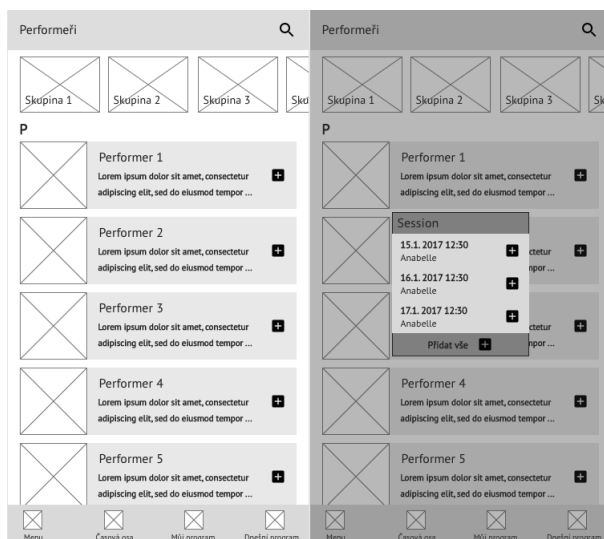
## 5. NÁVRH UŽIVATELSKÉHO ROZHRANÍ



Obrázek 5.4: Mobilní klientská aplikace: můj program

uživatel přidat do *mého programu*.

Seznam performerů lze filtrovat a to jak podle jména performerů, tak podle tagů<sup>22</sup>.

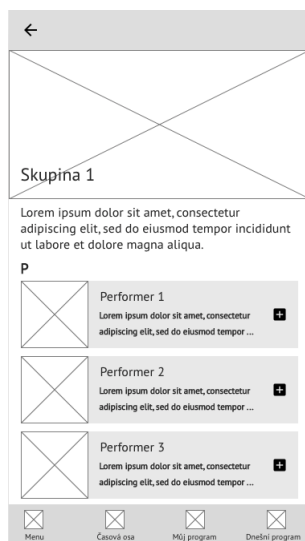


Obrázek 5.5: Mobilní klientská aplikace: seznam performerů

<sup>22</sup>Tag = značka. Ekosystém umožňuje k některým entitám přiřadit množinu značek.

### 5.2.1.6 Detail skupiny performerů

Ekosystém umožňuje zařazovat performery do tématických skupin, což se hodí zejména u filmových festivalů. Každá tématická skupina má svůj název a popis a může mít uvedeny obrázky. Přesně tyto informace se zobrazují na obrazovce *detailu skupiny performerů* (viz. obrázek 5.6) se seznamem performerů přiřazených do skupiny, který je navržen podobně jako úplný seznam performerů.

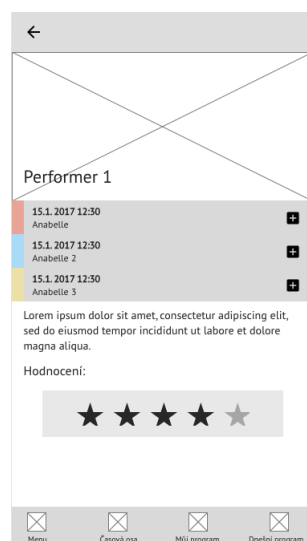


Obrázek 5.6: Mobilní klientská aplikace: detail skupiny performerů

### 5.2.1.7 Detail performerera

Obrazovka s detailem performerera (viz. obrázek 5.7) obsahuje veškeré informace, které jsou k němu v ekosystému uvedeny: obrázky, název, popis, odkazy a další informace. Také obsahuje seznam všech session, na kterých se performer prezentuje, s možností jejich přidání či odebrání z *mého programu*. Součástí obrazovky je i možnost hodnocení performerera pomocí klasického pětihvězdičkového hodnocení.

Byl upřednostněn seznam performerů a jejich detail s uvedenými session oproti seznamu všech session a to kvůli předpokladu, že na filmových festivalech při plánování zajímají návštěvníky vybrané filmy a pro ně si najdou konkrétní promítání. Na hudebních festivalech budou návštěvníci spíše plánovat pouze hlavní kapely a ostatní plány budou upravovat pomocí *časové osy* až v průběhu festivalu.



Obrázek 5.7: Mobilní klientská aplikace: detail performerera

### 5.2.1.8 Novinky

Aby bylo možné informovat návštěvníky i v průběhu události a to nejlépe v reálném čase, byl v ekosystému navržen systém novinek, které uveřejňují pořadatelé v administraci. Aplikace si novinky při uveřejnění stáhne a zobrazí je uživateli. Na obrazovce *dnešní program* a u položky *novinky* v menu se zobrazuje počet nepřečtených novinek. Obrazovka *novinky* (viz. obrázek 5.8) obsahuje jejich seznam s uvedeným názvem, popisem a obrázkem. V novinkách je možné vyhledávat.

Pro využití sociálních funkcí byla do návrhu přidána možnost označení novinky jako „líbí se mi“ a komentování novinek. Na obrazovce *novinky* se u každé novinky zobrazuje počet označení „líbí se mi“ a počet komentářů. Kliknutím na „líbí se mi“ označí uživatel novinku a kliknutím na počet komentářů je přesměrován na *detail novinky* do komentářů.

Seznam novinek může obsahovat i některé push notifikace, které je vhodné uživateli uložit, aby si je poté mohl opět přečíst. Push notifikace nemají sociální funkce ani detail a mají pouze informační hodnotu. Mohou mít navázanou akci na klik jako třeba přepnutí na konkrétní obrazovku.

### 5.2.1.9 Detail novinky

*Detail novinky* (viz. obrázek 5.9) obsahuje kompletní obsah novinky se všemi obrázky, které byly k novince přidány, a se všemi komentáři, které k novince uživatelé aplikace přidali, a možnost přidat nový komentář (jméno, které bude u komentáře uvedeno, si zvolí uživatel v nastavení aplikace). Opět je zde tlačítko pro označení novinky jako „líbí se mi“.





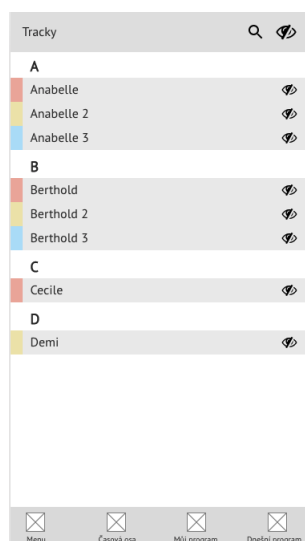
Obrázek 5.8: Mobilní klientská aplikace: novinky



Obrázek 5.9: Mobilní klientská aplikace: detail novinky

### 5.2.1.10 Seznam tracků

Obrazovka *seznam tracků* (viz. obrázek 5.10) obsahuje seznam všech tracků řazený abecedně. Uživatel může na obrazovce zvolit skrytí tracku na *časové ose* nebo track opět na ose zobrazit, také může skrýt či zobrazit všechny tracky najednou. Po kliknutí na konkrétní track v seznamu je uživatel přesměrován na *detail tracku*. Obrazovka opět obsahuje možnost vyhledávání pro případ, kdy má událost mnoho tracků.



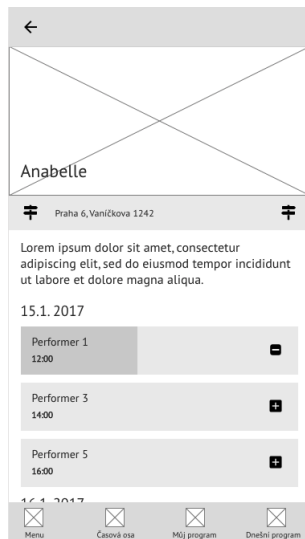
Obrázek 5.10: Mobilní klientská aplikace: seznam tracků

### 5.2.1.11 Detail tracku

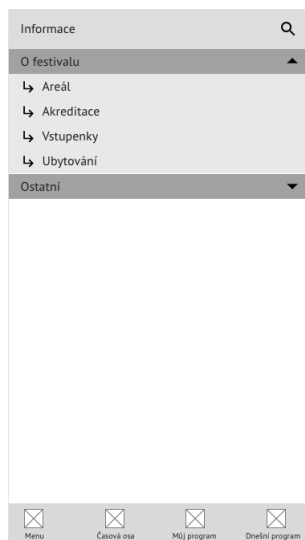
*Detail tracku* (viz. obrázek 5.11) obsahuje název, popis, obrázky a další informace, které byly k tracku přiřazeny. Je-li k tracku přiřazen pin, pak je zde uvedena také adresa a možnost naplánování trasy. Detail obsahuje také seznam všech session, které jsou na tracku plánovány, seřazený podle dní, ve kterých probíhají. Každou session lze opět přidat či odebrat z *mého programu*.

### 5.2.1.12 Informace

Informace jsou velmi důležitou součástí klasického papírového programu. Návštěvník v nich nalezne všechny důležité informace, které potřebuje před návštěvou i během ní vědět. Obrazovka *informace* (viz. obrázek 5.12) obsahuje do sekcí rozdělený seznam odkazů na *detaily informací*. Sekce jsou použity ke strukturování seznamu informací a každou sekci lze zavřít pro lepší přehled v seznamu. Součástí obrazovky je vyhledávání v obsahu všech informací.



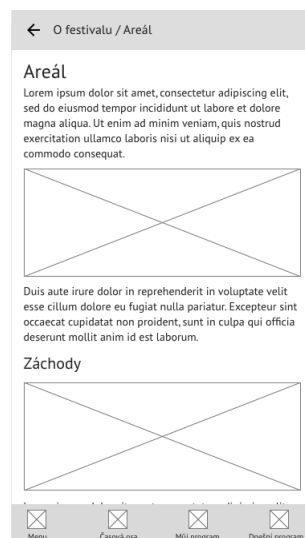
Obrázek 5.11: Mobilní klientská aplikace: detail tracku



Obrázek 5.12: Mobilní klientská aplikace: informace

### 5.2.1.13 Detail informace

Konkrétní jedna obrazovka s informacemi (viz. obrázek 5.13) například „o události“, „o partnerech“, „o ubytování“ atd. Kompletní strukturu sekcí a informací a jejich obsahu vytvářejí pořadatelé v administraci ekosystému. Každý *detail informace* pak obsahuje interpretovaný obsah v HTML<sup>23</sup>, který v administraci vytvořili pořadatelé.



Obrázek 5.13: Mobilní klientská aplikace: detail informace

### 5.2.1.14 Mapy

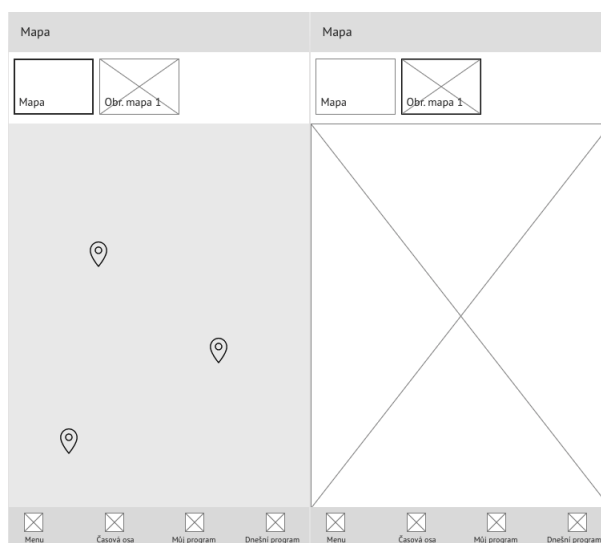
Mapy jsou rozděleny na dva druhy. **Obrázkové mapy** využijí spíše hudební festivaly pro zakreslení obsahu areálu festivalu. **Interaktivní mapy** mohou využít všechny události, hodí se v případě, kdy je událost pořádána na více místech.

Obrazovka *mapy* (viz. obrázek 5.14) je rozdělena do dvou sekcí. První sekce obsahuje seznam všech map, které jsou k události poskytnuty. Zobrazuje se pouze, pokud existují obrázkové mapy. První položka je vždy odkaz na interaktivní mapu a ostatní jsou odkazy na obrázkové mapy. Po kliknutí na položku se přepne obsah druhé sekce.

Druhá sekce obsahuje buď obrázkovou mapu nebo interaktivní mapu. Interaktivní mapa obsahuje vizualizované piny. Pokud uživatel klikne na pin, pak se zobrazí malý pop-up s informacemi o adrese a tracku, který je s pinem spojený. Po kliknutí na pop-up je uživatel přesměrován na obrazovku *detail tracku*.

---

<sup>23</sup>**HyperText Markup Language (HTML)** je značkovací jazyk, který interpretuje a zobrazuje webový prohlížeč.



Obrázek 5.14: Mobilní klientská aplikace: mapy

### 5.2.1.15 Dotazníky

Dotazníky mohou posloužit k získání zpětné vazby od návštěvníků. Obrazovka je navržena tak, aby bylo vyplnění a odeslání dotazníků co nejjednodušší. Aby uživatel nemusel vyplňovat žádná textová pole, jsou v ekosystému navrženy pouze dotazníky typu **hvězdičkové hodnocení**, **ano/ne**, **jedna možná odpověď** a **více možných odpovědí** (tento způsob byl zvolen pro snížení kognitivní náročnosti na uživatele, protože neradi vyplňují dotazníky).

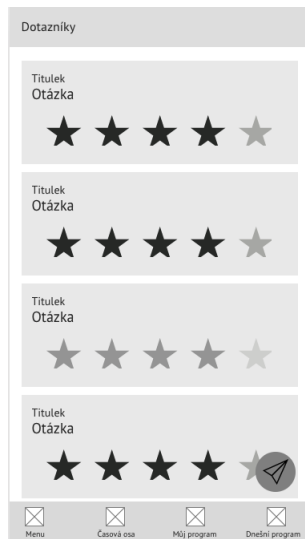
Obrazovka (viz. obrázek 5.15) obsahuje všechny dotazníky, které přidali pořadatelé v administraci. Na každý dotazník lze přímo na obrazovce odpovědět. Pokud uživatel odpověděl alespoň na jeden dotazník, zobrazí se tlačítko pro odeslání vyplněných dotazníků. Uživatel nemusí vyplnit všechny dotazníky. U dotazníků, které uživatel odeslal, se mohou zobrazit vizualizované výsledky v podobě jednoduchých grafů.

### 5.2.1.16 Diskuze

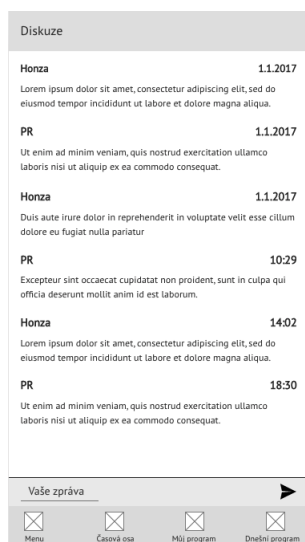
Obrazovka s *diskuzí* (viz. obrazovka 5.16) byla navržena jako velmi jednoduchý chat, ve kterém mohou uživatelé komunikovat spolu nebo s pořadatelem. Obsahuje seznam všech příspěvků v diskusi seřazený od nejnovějších se jmény uživatelů, datem vytvoření příspěvku a jeho obsahem. Ve spodní části obrazovky je pak pole pro vytvoření příspěvku a možnost jeho odeslání. Jméno, které bude u příspěvku uvedeno, si zvolí uživatel v nastavení aplikace.

## 5. NÁVRH UŽIVATELSKÉHO ROZHRANÍ

---



Obrázek 5.15: Mobilní klientská aplikace: dotazníky



Obrázek 5.16: Mobilní klientská aplikace: diskuze

### 5.2.1.17 Dodatečné informace k návrhu

Na každé obrazovce se zobrazuje rychlé horizontální menu se čtyřmi položkami, které je umístěno ve spodní části obrazovky. Tři položky jsou nejčastější obrazovky, které uživatelé využívají. Pořadatelé mohou vybrat, které to budou, výchozí jsou *časová osa*, *můj program* a *dnešní program*. Čtvrtá položka otevírá menu s odkazy na ostatní obrazovky.

Pokud si uživatel přidá či odebere položku ze svého program, je zobrazena systémová notifikace o této změně, aby tak uživatel jasně chápal, co provedl za akci, a mohl ji v případě nutnosti opravit.

Před začátkem session, kterou má uživatel uloženou v *mém programu*, se zobrazí systémová notifikace s upozorněním o začátku session. Uživatel si může zvolit v nastavení aplikace kolik minut před začátkem se má notifikace odesílat. Když session skončí, tak může uživateli přijít push notifikace o možnosti hodnocení performerů. Tato notifikace se ukládá do *novinek* a obsahuje proklik na detail performerů a je na pořadateli, jestli budou nebo nebudou odesílány. Další push notifikace se odesílají, pokud pořadatelé přidají nějakou novinku.

Aby mohli uživatelé aplikaci lépe přizpůsobit svým potřebám, je možné v nastavení aplikace zvolit, která obrazovka bude výchozí po ukončení *loading screen*.

### 5.2.1.18 Shrnutí

Návrh vychází z původního stavu mobilních klientských aplikací a ekosystému. Některé obrazovky byly pozměněny jen nepatrně (např. *seznam performerů*), některé byly úplně odstraněny (*program* byl rozdělen na více obrazovek), jiné byly předělány (např. *časová osa*) a některé byly navrženy navíc, aby pomohly plnit cíle uživatelů (např. *můj program*).

Návrh využívá veškeré možnosti ekosystému a se snaží držet v hranicích, které ekosystém svými možnostmi vytváří. Některé funkce (zejména sociální u *novinek*) byly přidány nově, protože dle autorova mínění v aplikacích chybí. Je možné dělat tak zásadní změny, protože není cílem práce mobilní klientskou aplikaci implementovat, u webové klientské aplikace by to nešlo.

Návrh se snaží být obecný a nezaměřovat se ani na jednu podporovanou mobilní platformu (*iOS* a *Android*). Počítá se s tím, že bude nutné návrh pro jednotlivé platformy dle *guidelines* lehce upravit. Návrh v tomto stavu by mělo být možné implementovat pro obě platformy a neměl by porušovat žádné *guidelines*. Spodní menu rychlého výběru bylo v nedávné době přidáno i do *guidelines* pro Android.

Takto navržené uživatelské rozhraní mobilní klientské aplikace plní následující stanovené požadavky:

- **Jednotné a znovupoužitelné uživatelské rozhraní pro všechny druhy událostí:** bylo splněno jednotným návrhem pro všechny druhy

událostí s možností vypnout či zapnout některé funkce pro konkrétní událost.

- **Podpora prezentace společnosti vyvíjející ekosystém a pořadatelů a jejich partnerů:** řeší grafický návrh pro konkrétní událost, obsah *loading screen* a *informací*.
- **Poskytnutí možností, které přináší papírový program:** řeší *časová osa*, *seznam performerů*, *seznam tracků*, *informace* a *mapy*.
- **Možnost vytvoření vlastního programu a prezentování informací z něj plynoucích:** řeší funkce pro vytváření vlastního programu a obrazovka *můj program* a *dnešní program*.
- **Možnost komunikace mezi pořadatelem a účastníky pro poskytnutí zpětné vazby a důležitých informací:** podpořeno ve funkcích obrazovek *diskuze* a *dotazníky* a v notifikacích.
- **Možnost získat dodatečné informace, které při plánování události nebyly známy:** podpořeno pomocí *novinek* a *notifikací*.
- **Možnost zapojení se do průběhu události:** Řeší hodnocení performerů, komentáře novinek nebo *diskuze* a *dotazníky*.
- **Možnost naplánování návštěvy události:** podpořeno pomocí tvorby vlastního programu, možností plánování trasy nebo poskytování informací.
- **Možnost přizpůsobení aplikace uživatelem podle jeho potřeb:** uživatel si může vybrat svou výchozí obrazovku nebo si může vybrat, jak dlouho dopředu mu přijde notifikace o začátku session, kterou má ve svém programu.

### 5.2.2 Webová klientská aplikace

Do webové klientské aplikace bylo navrženo pět stránek. Uživatel si může naplánovat vlastní program, procházet aktuální program a performery, zobrazit si mapu a také základní informace o události. Díky těmto funkcím je možné podpořit uživatelské cíle v případě plánování návštěvy i v průběhu události a zároveň neporušit business cíle vývojářů.

Do aplikace byly navrženy následující stránky: **homepage**, **časová osa**, **seznam performerů**, **mapa** a **informace**.

#### 5.2.2.1 Homepage

*Homepage (Domovská stránka)* (viz. obrázek 5.17) je výchozí a nejdůležitější částí každé webové aplikace. Na tuto stránku přijdou všichni uživatelé a proto



by měla obsahovat nejdůležitější informace. Aby bylo možné aplikací suplovat mobilní klientské aplikace pro nepodporované mobilní platformy, byly do této stránky navrženy následující sekce:

**Můj program** obsahuje seznam session seřazený podle času jejich začátku od nejbližších. Položky v seznamu vizualizují probíhající session a obsahují proklik do mapy na track, kde session bude probíhat (pokud má track přiřazen pin), a také proklik na informace o performerovi. Položky, které už proběhly, se dále v programu nezobrazují.

Pokud uživatel nemá v programu žádné session, pak se zobrazí informace o možnosti program vytvořit. V případě, že událost skončila, je zde uvedena informace o této skutečnosti.

**Program pro nejbližších 24 hodin** obsahuje seznam session, které budou probíhat v následujících dvacetičtyřech hodinách, čímž se lze vyhnout výpisu všech session a navíc informovat, pokud událost již skončila. Seznam je rozřazen podle tracků a session z tracku, který uživatele nezajímá, lze lehce skrýt. Položky v programu lze přidat do *mého programu* a opět obsahují vizualizaci, pokud již probíhají.

Pokud už neexistuje žádný program, pak se tato sekce vůbec nezobrazuje.

**Novinky** obsahují výpis novinek, které k události přidali pořadatelé. Zobrazuje se vždy první obrázek novinky (pokud existuje) a následně její obsah. Pokud není zobrazena sekce *aktuální program*, pak novinky zaberou i její prostor.

Pro podporu business cílů pořadatelů byla navržena následující sekce:

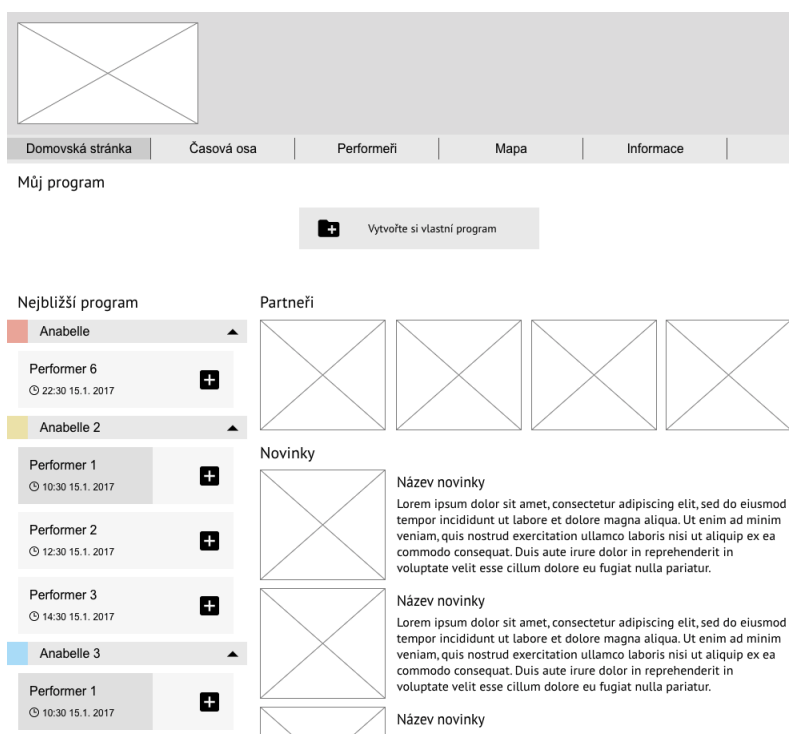
**Partneři** jsou sekce, která se zobrazuje pouze, pokud o to mají pořadatelé zájem. V této sekci je možné uvést loga partnerů události a prezentovat je tak návštěvníkům.

### 5.2.2.2 Časová osa

*Časová osa* (viz. obrázek 5.18) je důležitá pro plánování návštěvy i orientaci v programu události. Z tohoto důvodu byl opět zvolen přístup, který používá například *Google Calendar* nebo *Kalendář v MacOS* k zobrazení po dnech, a na který je mnoho uživatelů zvyklých.

*Časová osa* je zobrazována vertikálně po sloupcích. Uplně první sloupec je úžší a zobrazuje osu s časem rozdělenou po patnácti minutách. Každý další sloupec má v hlavičce uveden název tracku a zobrazuje časovou posloupnost všech session, které na něm probíhají. *Časovou osou* lze pohybovat ve vertikálním směru pro průchod programem v čase a horizontálním směru pro průchod programem přes tracky. Tímto způsobem jsou vedle sebe zobrazeny průběhy

## 5. NÁVRH UŽIVATELSKÉHO ROZHRANÍ



Obrázek 5.17: Webová klientská aplikace: domovská stránka

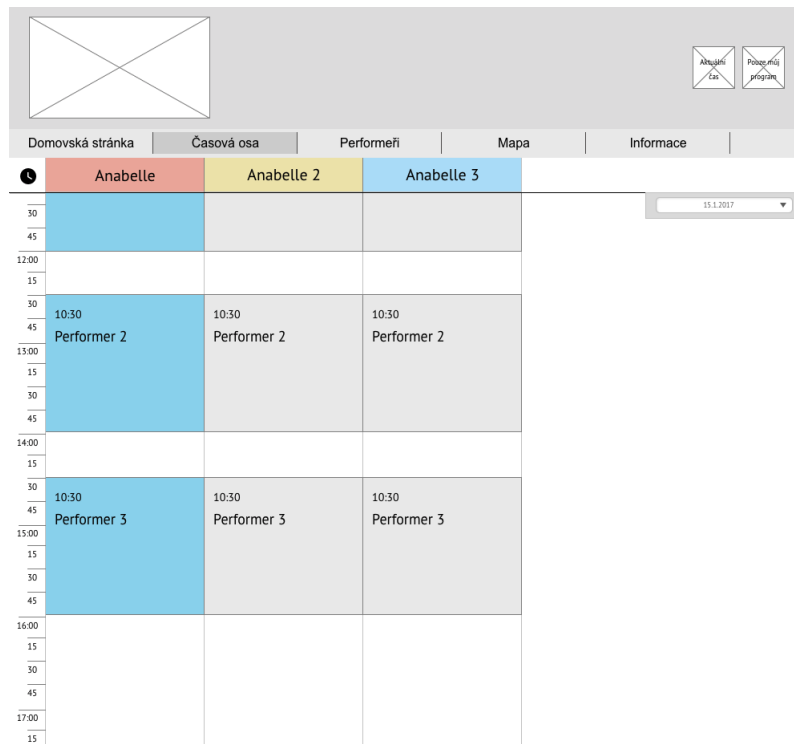
všech session na všech tracích a byl zvolen kvůli přirozenějšímu vertikálnímu pohybu pomocí kolečka myši, které je při použití webu na stolních počítačích často využíváno.

Při kliknutí na název tracku je uživatel přesměrován na mapu do detailu pinu přiřazeného ke tracku. Při kliknutí na session je uživatel přesměrován na *detail performerera*. V *časové ose* jsou barevně zvýrazněny session, které má uživatel v *mém programu*.

V *časové ose* se lze rychle přesouvat pomocí výběru dne z jejich seznamu, který je umístěn vždy nad stránkou ve fixní poloze pravého dolního rohu. V této pozici je také umístěno tlačítko na zobrazení pouze session z *mého programu* a tlačítko pro přesunutí osy na aktuální čas.

### 5.2.2.3 Seznam performerů

Stránka (viz. obrázek 5.19) obsahuje informace o všech performerech události. Je rozdělena na dvě sekce ve dvou sloupcích. První sekce obsahuje seznam performerů řazený abecedně, kde každá položka obsahuje obrázek performerera, jeho název a část jeho popisu. Seznam lze filtrovat podle názvu performerera a podle tagů, které má přiřazené. Po kliknutí na položku v seznamu se ve druhém sloupci zobrazí detail konkrétního performerera.



Obrázek 5.18: Webová klientská aplikace: časová osa

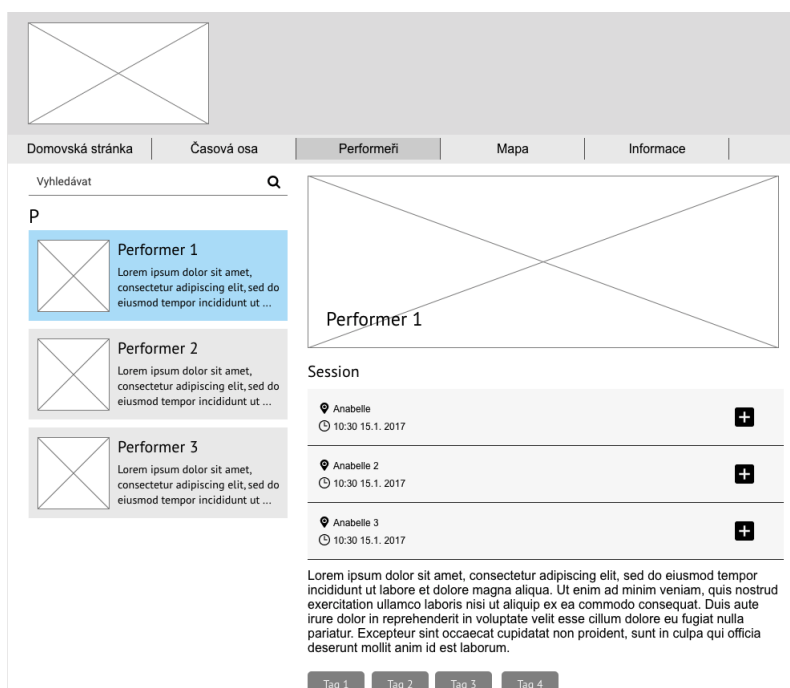
*Detail performerera* obsahuje veškeré informace, které jsou o něm v ekosystému uvedeny, a také seznam všech session, kterých se performer účastní, a možnost jejich přidání či odebrání z *mého programu*.

#### 5.2.2.4 Mapa

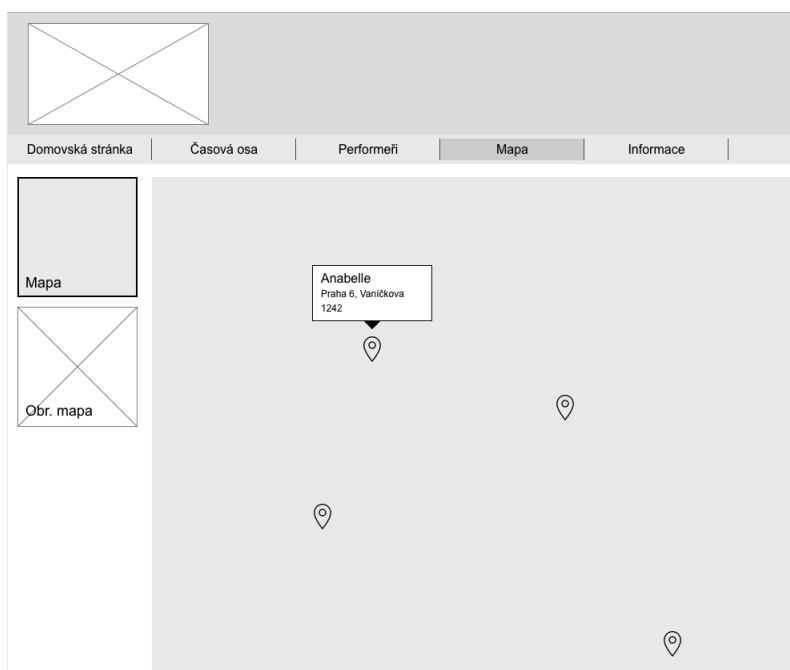
Stránka (viz. obrázek 5.20) opět obsahuje dvě sekce jako dva sloupce. První sekce se zobrazuje pouze v případě, že byly v administraci pro danou událost vloženy obrázkové mapy. Jedná se o seznam náhledů map, kde první náhled je vždy odkaz na mapu s piny. Ostatní náhledy jsou odkazy na obrázkové mapy.

Po kliknutí na náhled v první sekci se mění obsah druhé sekce podle druhu mapy. Pokud se jedná o obrázkovou mapu, tak se zobrazí tento obrázek. Pokud se jedná o mapu s piny, pak se zobrazí interaktivní mapa s vizualizovanými piny. Po kliku na pin v mapě se zobrazí informace o názvu tracku, který je k pinu přiřazen a adresa pinu. Ke stejné vizualizaci dojde při prokliku na mapu z *mého programu* nebo *časové osy*.

## 5. NÁVRH UŽIVATELSKÉHO ROZHRANÍ



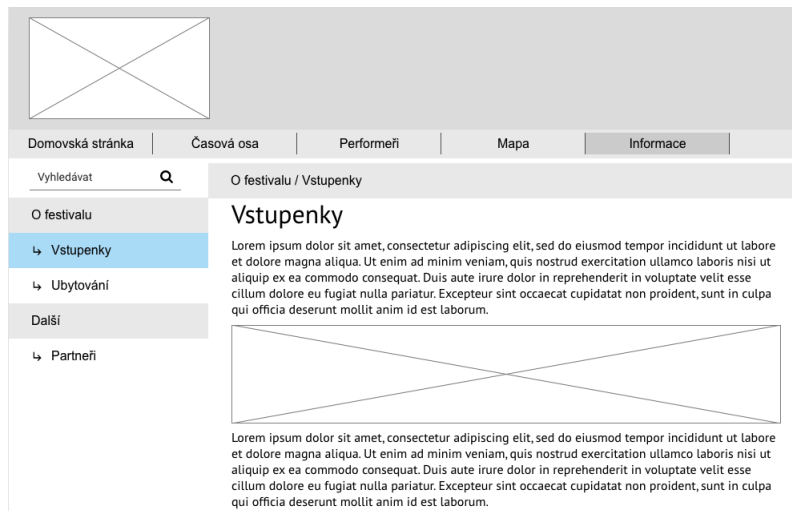
Obrázek 5.19: Webová klientská aplikace: seznam performerů



Obrázek 5.20: Webová klientská aplikace: mapy

### 5.2.2.5 Informace

Jelikož ekosystém dovoluje pořadatelům přidávat informace o festivalu v administraci pomocí editoru, který generuje HTML kód, zobrazuje stránka tento vygenerovaný kód. Stránka (viz. obrázek 5.21) je opět rozdělena na dva sloupce. V prvním sloupci je seznam všech sekcí s informacemi a po kliknutí na konkrétní sekci se zobrazí v druhém sloupci informace v HTML, které byly vytvořeny v administraci.



Obrázek 5.21: Webová klientská aplikace: informace

### 5.2.2.6 Shrnutí

Aplikace musí splňovat business cíle vývojářů, proto byly použity jen základní funkce. Také není bráno v potaz přizpůsobení rozhraní pro mobilní zařízení z důvodu donucení pořadatelů ke koupi mobilních klientských aplikací. Přesto by však měla takto webová klientská aplikace pomáhat plnit cíle uživatelů.

Uživatelé se vyjádřili, že by aplikaci použili spíše před událostí pro její naplánování. To je možné, ekosystém však v současné době nepodporuje uživatelské účty a proto není možné synchronizovat vytvořený vlastní program do mobilní klientské aplikace.

Jelikož je cílem práce navržené uživatelské rozhraní také implementovat ve webové klientské aplikaci, musel se návrh držet v hranicích, které stanovuje současný stav ekosystému. Tato situace byla zjednodušena business cíli vývojářů, které nedovolují aplikaci udělat příliš složitou.

Takto navržené uživatelské rozhraní webové klientské aplikace plní následující stanovené požadavky:

- **Jednotné a znovupoužitelné uživatelské rozhraní pro všechny druhy událostí:** bylo splněno jednotným návrhem pro všechny druhy událostí.
- **Snaha o donucení pořadatelů ke koupi mobilní klientské aplikace poskytnutím ve webové klientské aplikaci pouze standardních funkcí a prezentováním přidané hodnoty na mobilní klientské aplikaci:** v aplikaci byly poskytnuty pouze základní funkce pro jednoduché plánování návštěvy a zjištění informací o události.
- **Podpora prezentace společnosti vyvíjející ekosystém a pořadatelů a jejich partnerů:** řeší grafický návrh pro konkrétní událost, *úvodní stránka* se sekci partnerů a *informace*.
- **Poskytnutí možností, které přináší papírový program:** řeší program poskytnutý v *časové ose*, *seznam performerů*, *informace* a *mapy*.
- **Možnost vytvoření vlastního programu a prezentování informací z něj plynoucích:** potlačeno business cíly vývojářů, ale lze si vytvořit vlastní program.
- **Možnost získat dodatečné informace, které při plánování události nebyly známy:** řeší sekce novinek na *úvodní stránce*
- **Možnost naplánování návštěvy události:** řeší možnost vytvoření vlastního programu.
- **Omezit funkcionalitu na základní minimum, které poslouží uživateli, pokud jeho mobilní platforma není Android nebo iOS nebo pokud pořadatel nezakoupil mobilní verzi aplikace:** uživatel si může vytvořit vlastní program a procházet informace o události pomocí stránek *úvodní stránka*, *časová osa*, *seznam performerů*, *informace* a *mapy*.
- **Neoptimalizovat pro mobilní zařízení, ale zařídit funkčnost všech funkcí i na mobilním zařízení:** technický požadavek, bude řešen v implementaci.

### 5.3 Shrnutí kapitoly

Mezi výstupy fáze vývoje popsané v této kapitole patří **navrhované funkce**, které vycházejí z uživatelských cílů a business cílů, které by měly podporovat plněním stanovených požadavků. Dalším výstupem jsou dva druhy vzniklých lo-fi prototypů, které jsou v této kapitole popsány pomocí **návrhových rozhodnutí**. Prototypy realizují vybrané navrhované funkce z hlediska uživatelského rozhraní. Prvním z nich jsou papírové **wireframy**, které lze jednoduše

a rychle měnit a které byly použity zejména v počáteční fázi návrhu uživatelského rozhraní pro ustálení myšlenek a záměrů. Po ustálení záměrů s návrhem byly wireframy převedeny do interaktivních mockupů pomocí webové aplikace *UXPin.com*<sup>24</sup>. Mockupy lze použít jako dokumentaci uživatelského rozhraní pro fázi implementace a lze je také relativně dobře testovat s uživateli. **Wireframy i mockupy lze nalézt na přiloženém médiu.**

---

<sup>24</sup>Dostupné z: <https://www.uxpin.com>





---

# Implementace webové klientské aplikace

**Implementace** je fáze vývoje, ve které je vytvořený návrh uživatelského rozhraní převeden do konkrétních technologií pro danou platformu, a může být v prvních iteracích vývoje vynechána, dokud není návrh uživatelského rozhraní zpřesněn natolik, aby se vyplatila jeho realizace.

V této fázi je důležité, aby programátoři implementující uživatelské rozhraní nebyli odděleni od designérů a mohli s nimi konzultovat myšlenky, které za návrhem stojí, protože mohou vyvstát otázky ohledně situací, na které designéři nemysleli a je třeba je upřesnit. Komunikace designérů a programátorů podporuje zapojení myšlenek *Lean UX*.

## 6.1 Výběr technologií

Před samotnou implementací je potřeba vybrat technologie, které je vhodné použít na dané platformě. Nejedná se však pouze o technologie, které budou aplikaci tvořit, ale i o technologie, pomocí kterých bude aplikace vytvořena (bude rozvedeno později).

V případě webové aplikace je nutné vybrat vhodné technologie pro webovou platformu, tedy technologie, se kterými bude schopen pracovat webový prohlížeč a webový server. Jelikož aplikace využívá klientské REST API, které poskytuje ekosystém, není nutné vytvořit složitou serverovou logiku a je možné aplikaci naprogramovat tak, aby fungovala pouze ve webovém prohlížeči. Tento postup dovolí vytvořit relativně chytrou aplikaci, kterou lze rychle nasadit do produkčního prostředí a jejíž provoz není drahý.

Na druhou stranu taková aplikace závisí na podpoře použitých technologií v každém webovém prohlížeči, což může být s množstvím prohlížečů na trhu problém.

### 6.1.1 Hyper Text Markup Language

**Hyper Text Markup Language (HTML)** je značkovací jazyk, který je používán pro vytváření webových stránek (HTML dokumentů). HTML popisuje strukturu a význam obsahu dokumentu pomocí elementů, které jsou definovány speciálními značkami (tagy). Tagy jsou interpretovány a jejichž obsah je zobrazován pomocí webového prohlížeče. Cílem webového prohlížeče tedy není zobrazovat HTML tagy, ale na jejich základě zobrazovat obsah dokumentu. Existuje více verzí HTML a pro tuto práci byla vybrána nejnovější verze **HTML5**. [15]

#### 6.1.1.1 HTML Document Object Model

**Document Object Model (DOM)** je programově nezávislé rozhraní, pomocí kterého mohou programy přistupovat k obsahu a vzhledu dokumentu a dynamicky je měnit [16].

**HTML DOM** vytváří webový prohlížeč po načtení HTML dokumentu a popisuje daný dokument pomocí stromu objektů. Díky HTML DOM lze programově přistupovat k HTML dokumentu a měnit jej. [17]

### 6.1.2 Cascading Style Sheets

HTML popisuje obsah webové stránky. **Cascading Style Sheets (CSS)** popisuje, jakým způsobem má webový prohlížeč webovou stránku zobrazit. Pomocí definovaných CSS pravidel lze popsat, jakým způsobem mají jednotlivé HTML elementy vypadat, a jednu definici vzhledu (CSS styly) lze použít pro více dokumentů. [18]

CSS také umožňuje, díky vlastnosti **media queries**, oddělit vzhled dokumentů pro různé účely, mezi které může patřit zobrazení na různých velikostech obrazovek či zobrazení pro tisk. Takto lze zobrazit jeden HTML dokument různými způsoby na různých zařízeních a poskytnout tak uživateli kvalitní uživatelský zážitek. [19]

Ke vzhledu definovanému pomocí CSS lze opět programově přistoupit pomocí HTML DOM a měnit jej. [17]

#### 6.1.2.1 CSS frameworky

CSS frameworky<sup>25</sup> jsou sady předpřipravených CSS stylů, které může programátor využít pro úpravu vzhledu webových stránek bez nutnosti psát vlastní styly. Rozdíl mezi různými CSS frameworky je zejména v množství možností, které poskytují. Pro tuto práci byl vybrán komplexní framework **Bootstrap 3**<sup>26</sup>, který přináší veškeré nutné styly pro psaní použitelných webových strán-

---

<sup>25</sup>**Framework** je předpřipravená programová struktura.

<sup>26</sup>Dostupné z <http://getbootstrap.com/>

nek a aplikací. Bootstrap byl použit pro základní vzhled aplikace a zbytek vzhledu byl následně upraven pomocí vlastních CSS stylů.

Použití CSS frameworku bylo vybráno, protože je cílem práce navržení uživatelského rozhraní a ne grafický návrh, tudíž je důležité zejména rozložení prvků, obsah a chování aplikace, i když i grafický návrh je samozřejmě neoddělitelnou součástí návrhu, protože umožňuje formovat emoce, které budou v uživateli vyvolány (viz. kapitola o UxD).

### 6.1.2.2 Syntactically Awesome StyleSheets

**Syntactically Awesome StyleSheets (SASS)** je rozšíření, které přináší výhody klasických programovacích jazyků do CSS orientovaného pouze na pravidla, a proto byl vybrán na tvorbu stylů pro tuto práci. Mezi výhody, které SASS přináší, patří:

- Rozdělení na nezávislé kusy kódu (`@mixin`).
- Rozdělení na moduly (`@import`).
- Proměnné a jiné programovací struktury jako podmínky nebo cykly (`$var`, `@if`, `@for` atd.).
- Možnost dědění pravidel (`@extend`) a další.

Pomocí SASS lze i složité styly udržovat relativně jednoduché a dobře organizované. Jelikož webové prohlížeče neumí SASS interpretovat, je nutné jej nejprve převést na CSS. Projekt SASS poskytuje knihovny a programy pro překlad mezi syntaxemi SASS a CSS. [20]

### 6.1.3 JavaScript

**JavaScript** je programovací jazyk, který je interpretován prohlížečem a umožňuje modifikaci a úpravu jinak statických HTML dokumentů. Pomocí JavaScriptu je možné vytvářet webové aplikace zpracovávané pouze v prohlížeči a proto byl vybrán pro implementaci webové klientské aplikace. [21]

JavaScript vychází ze standartu ECMAScript (ES), který definuje jeho syntax a sémantiku [22]. JavaScript navíc ke standartu ECMAScript přidává řadu aplikačních rozhraní, které je možné při programování JavaScriptových aplikací využívat. Mezi tyto aplikační rozhraní patří například rozhraní pro modifikaci a práci s HTML dokumenty pomocí HTML DOM. Dalším aplikačním rozhraním je například rozhraní pro asynchronní dotazy **Asynchronous JavaScript and XML (AJAX)**, pomocí kterého je možné vytvářet a zasílat na pozadí požadavky na webové servery přímo z programového kódu JavaScriptu a získané odpovědi opět používat k modifikaci HTML dokumentů, to vše bez nutnosti HTML dokument znovu načítat [23]. Existuje mnoho dalších aplikačních rozhraní například pro práci přímo s prohlížečem, s historií prohlížeče atd [21].

### 6.1.3.1 Moderní verze JavaScriptu

V současných prohlížečích je plně podporována verze JavaScriptu, která vychází ze standartu ES5. Tato verze byla vydána v roce 2009 a od té doby byly vytvořeny novější standardy jako například ES6 nebo ES7. [22]

Problém použití nových verzí standartu je, že je staré prohlížeče nepodporují a tím pádem by nebylo možné nové verze využívat, pokud je požadována funkčnost aplikací ve starých verzích webových prohlížečů. Současně s novými verzemi ES přichází i nová aplikační rozhraní jako například **Fetch API**, které zjednodušuje vytváření asynchronní požadavků a také by nebylo možné jej používat.

### Babel

Cestou pro psaní JavaScriptových programů v nových standardech je použití kompilace programového kódu z nových do starých standartů. V tomto případě se jedná o kompilaci z verzí ES6 a ES7 do podporované verze vycházející ze standartu ES5. Jedním z takových kompilátorů je **Babel**, díky kterému lze používat moderní JavaScript již dnes [24].

Samotná kompilace však nepomůže v případě nových aplikačních rozhraní, protože Babel kompiluje pouze syntaxi. Pro tento případ obsahuje Babel také tzv. **polyfilly**, které umožňují použití nových aplikačních rozhraní i v prohlížečích, které je nepodporují. Polyfilly jsou programové kusy JavaScriptového kódu, které nahrazují či simulují neexistující aplikační rozhraní. [24]

Ne všechna aplikační rozhraní lze pomocí polyfillů simulovat a proto je nutné zvážit, jaký poměr mají jednotlivé prohlížeče v globálním použití a jestli je nutné je podporovat.

### 6.1.3.2 JavaScriptové frameworky

Za dobu existence JavaScriptu vznikly různé knihovny a frameworky, které zjednodušují programátorovi vytváření aplikací. Mezi tyto knihovny patří například **jQuery**<sup>27</sup>, které zjednodušuje získávání elementů z DOM pomocí CSS selektorů a které bylo vybráno pro použití v této práci.

Existují také komplexní frameworky, pomocí kterých lze psát složité JavaScriptové aplikace a které obsahují veškeré programové struktury potřebné pro tvorbu takových aplikací. Mezi tyto frameworky patří například **Angular**<sup>28</sup>. Problém těchto frameworků je, že i pro jednoduché aplikace je nutné napsat mnoho programového kódu, což je pro účely této práce zbytečně složité.

---

<sup>27</sup>Dostupné z: <https://jquery.com/>

<sup>28</sup>Dostupné z: <https://angular.io/>

## React

**React** je knihovna pro jednoduchou tvorbu uživatelských rozhraní JavaScriptových aplikací. Uživatelské rozhraní v Reactu je tvořeno z komponent. Každá komponenta je JavaScriptová třída, která zobrazuje a obsluhuje část uživatelského rozhraní a proto má jednu povinnou metodu `render`, která je volána Reactem v případě, že je komponenta kreslena do DOMu HTML dokumentu. Komponenty jsou definovány dvěma druhy proměnných: [25]

**Properties** jsou proměnné určené pouze pro čtení a předávají se do instance komponenty při jejím vytváření a překreslování.

**State** jsou vnitřní proměnné komponenty, které definují její stav. Pokud se stavové proměnné změní, pak dojde k překreslení komponenty (opětovné volání metody `render`).

Komponenty procházejí životním cyklem, který má definované metody volané v různých fázích tohoto cyklu. Mezi tyto metody patří například: [25]

**componentWillMount** je metoda, která je volána v případě, že bude instance komponenty vložena do DOMu a vykreslena.

**componentDidMount** je metoda, která je volána po vložení instance do DOMu.

**componentDidUpdate** je metoda, která je volána po změně stavu a překreslení komponenty.

**componentWillUnmount** je metoda, která je volána před vyjmutím komponenty z DOMu.

React využívá pro vytváření komponent, které je prováděno například v metodě `render` nebo při vykreslování celého uživatelského rozhraní do HTML dokumentu, zápis pomocí syntaxe **JSX**. JSX umožňuje zapisovat vytváření komponent pomocí syntaxe podobné HTML přímo v JavaScriptovém kódu. Tento způsob umožňuje jednoduché a rychlé vytváření komplexních uživatelských rozhraní a z tohoto důvodu byl React vybrán pro vytvoření uživatelského rozhraní webového klienta aplikace. [25]

Pro zlepšení uživatelského zážitku při použití aplikací napsaných pomocí knihovny React se často používá knihovna **React-router**, která dovoluje psát JavaScriptové aplikace v Reactu, které se chovají, jako by načítaly nové HTML dokumenty místo modifikování DOMu (ukázka použití React-router s JSX viz. zdrojový kód 6.1). Pro tento účel je využíváno nové aplikační rozhraní JavaScriptu pro práci s historií prohlížeče **History API**. Díky React-router je možné sdílet jinému uživateli odkaz na sekci v aplikaci a on ji po zobrazení také uvidí, ačkoli se nejedná o samostatný HTML dokument (tj. musí dojít k vykonání programu a úpravě obsahu aplikace dle struktury odkazu).

```
1 import React from 'react';
2 import {Route} from 'react-router';
3
4 import HomepageContainer from '../containers/HomepageContainer';
5 import TimelineContainer from '../containers/TimelineContainer';
6 import PerformersContainer from '../containers/PerformersContainer';
7 import InfoContainer from '../containers/InfoContainer';
8 import MapsContainer from '../containers/MapsContainer';
9
10 const Routes = () => (
11   <div>
12     <Route exact path="/" component={HomepageContainer} />
13     <Route path="/timeline" component={TimelineContainer} />
14     <Route exact path="/performers" component={
15       PerformersContainer} />
16     <Route path="/performers/:id" component={PerformersContainer
17       } />
18     <Route exact path="/info" component={InfoContainer} />
19     <Route path="/info/:id" component={InfoContainer} />
20     <Route exact path="/maps" component={MapsContainer} />
21     <Route path="/maps/:id" component={MapsContainer} />
22   </div>
23 );
24 export default Routes;
```

Zdrojový kód 6.1: Použití JSX společně s knihovnou React-router ve webové klientské aplikaci.

### Redux

Pro udržování datového obsahu aplikace je také vhodné zvolit některou knihovnu. **Redux** je jednou z knihoven, kterou lze využít pro udržování datového obsahu JavaScriptových aplikací.

Redux definuje tři základní koncepty, které je nutné při jeho použití respektovat: [26]

1. Použití jednoho datového úložiště (**storu**), které udržuje celý stav aplikace.
2. Stav aplikace je **pouze pro čtení**.
3. Změny stavu aplikace (úprava **storu**) jsou prováděny pomocí tzv. **pure functions**<sup>29</sup> = **reducers**<sup>30</sup>.

Datový tok v Reduxu je synchronní a vždy stejný. Redux **store** definuje metodu **dispatch** pro vyvolání změny stavu aplikace a metodu **subscribe**

---

<sup>29</sup>**Pure function** je funkce, která pro danou kombinaci vstupních proměnných vrací vždy stejný výstup.

<sup>30</sup>**Reducer** je **pure function**, která přijímá **akci** a **stav** aplikace a vrací vždy **nový stav** aplikace.

pro odposlech změn stavu aplikace (metoda registruje funkce, které budou po změně stavu volány).

Změny stavu aplikace jsou prováděny voláním metody `dispatch` s *akcí*. *Akce* je JavaScriptový objekt s jednou povinnou položkou *type*, která rozlišuje druhy akcí. Akce jsou v `reducerech` aplikovány na stav aplikace a generují nový stav.

V případě, že došlo ke změně stavu aplikace, jsou zavolány všechny funkce, které byly zaregistrovány pomocí metody `subscribe`, a je jim předán aktuální stav, aby mohly na změnu nějakým způsobem reagovat. [26]

Redux je možné využívat s různými knihovnami na tvorbu uživatelských rozhraní, takže také s Reactem. *Akce* je možné pomocí metody `dispatch` vyvolávat přímo z React komponent a je možné registrovat metody z komponent pro naslouchání na změnu stavu pomocí metody `subscribe`. Pokud použijeme React a Redux, jsme schopni obstarat datový obsah aplikace a její uživatelské rozhraní.

Jelikož je taková aplikace pouze synchronní a asynchronní akce typu stažení dat z REST API by bylo nutné provádět v React komponentách, tedy v části, která má pouze kreslit uživatelské rozhraní, přidává Redux další konstrukt, který nazývá **middleware**. **Middleware** jsou kusy kódu, které upravují původní metodu `dispatch` a jsou aplikovány na *akci* před tím, než je použita pro vytvoření nového stavu. Díky middlewaru nemusí být *akce* pouze JavaScriptový objekt, ale může to být například pole objektů či funkce, a specifický middleware se postará o to, aby do finální metody `dispatch` dorazily pouze JavaScriptové objekty.

## Shrnutí

Kombinací knihoven React a Redux spolu s vhodnými Redux middlewary získáváme programovou strukturu, která je dostačující k vytváření interaktivních webových aplikací běžících ve webovém prohlížeči, které jsou navíc značně rychlé díky optimalizaci obou knihoven. React umožňuje tvořit uživatelská rozhraní a reagovat na akce, které s rozhraním provádí uživatel. Redux umožňuje udržovat stav (data) aplikace a Redux middlewary umožňují provádět asynchronní volání a operace a další business logiku aplikace. jQuery je možné použít v kombinaci v Reactem pro práci s HTML DOMem. Z výše uvedených důvodů byly tyto knihovny vybrány pro implementaci webového klienta aplikace.

### 6.1.4 Moderní tvorba webových aplikací

Jak bylo zmíněno výše, je nutné kromě technologií, které tvoří aplikaci, vybrat i technologie, pomocí kterých bude aplikace vytvořena. Tyto technologie pomáhají uchovávat, udržovat či různými způsoby zpracovávat zdrojový kód před

tím, než je aplikace vystavena k použití pro uživatele (do produkce, v tomto případě na webový server). Jedná se o technologie pro vývoj aplikací a jejich cílem je zejména zjednodušit práci programátorům.

### 6.1.4.1 Verzování

Verzování je velice důležité pro správu zdrojového kódu. Umožňuje uchovávat starší verze, jednoduše opravovat chyby, vracet se ke starším verzím atd. Jedním ze systémů pro verzování je **Git**. Git je zdarma k použití a je velmi jednoduché jej používat.

Základem Gitu je repozitář, který obsahuje veškeré verze a historii, která byla pomocí Gitu uchována. Repozitář lze jednoduše vytvořit v jakékoli složce a začít verzovat. V tomto případě se jedná o lokální repozitář. Dále lze lokální repozitář připojit ke vzdálenému repozitáři a mít tak vzdálenou zálohu všech uložených verzí. Pro vytváření vzdálených repozitářů a práci s nimi existují různé webové aplikace jako například *GitHub*<sup>31</sup> nebo *GitLab*<sup>32</sup>, který lze navíc nainstalovat na vlastní server. Aplikace *GitLab* je zdarma poskytnuta Fakultou informačních technologií a proto byl pro verzování programového kódu vybrán právě Git. [27]

### 6.1.4.2 Task runners

Jak bylo uvedeno výše, používají se při tvorbě webových aplikací technologie, které je nejprve nutné převést, aby bylo možné aplikaci použít ve dnešních webových prohlížečích. Mezi tyto technologie patří například SASS pro lepší práci se styly a nebo nové standardy JavaScriptu, které je nutné převést na podporované standardy.

Překlad SASS na CSS, kompilace nových standartů do podporovaných, minifikace zdrojových kódů, optimalizace obrázků či nasazení na webový server lze považovat za **úlohy (tasks)**, které je potřeba s programovým kódem provádět. Pro automatizaci takových úloh byly vytvořeny speciální programy, pomocí kterých je možné úlohy definovat a následně spouštět. Mezi tyto programy, nazývané **task runnery**, patří například **Grunt** nebo **Gulp**.

Základní vlastností **task runnerů** je vykonávání úloh na konkrétním druhu technologie nezávisle na ostatních druzích. To znamená, že JavaScript se kompiluje nezávisle na SASS nebo obrázcích, SASS se překládá nezávisle na JavaScriptu a obrázcích a tak dále. Úlohy lze následně spojovat do front a díky tomu lze spuštěním jedné úlohy provést všechny důležité modifikace zdrojového kódu a nainstalovat jej na webový server. [28]

---

<sup>31</sup>Dostupné z: <https://github.com/>

<sup>32</sup>Dostupné z: <https://about.gitlab.com/>



### 6.1.4.3 Webpack

Poté, co Node.js přenesl JavaScript z webových prohlížečů také do operačních systémů, přidal do své verze JavaScriptu některé konstrukty, se kterými neumí webové prohlížeče pracovat. Mezi tyto konstrukty patří rozdělení zdrojového kódu JavaScriptu do modulů závislostí, které mohou být v případě potřeby nahrány a použity. Díky modulům lze jednoduše spravovat závislosti aplikace a udržovat zdrojový kód přehledný.

JavaScript je v prohlížeči interpretován postupně takovým způsobem, jakým je načítán z HTML dokumentu, který jej obsahuje. Vznikají tím problémy s globálními proměnnými a kontextem existence proměnných, což zhoršuje přehlednost zdrojového kódu a jeho udržitelnost. **Webpack** přichází s řešením tohoto problému.

Webpack umožňuje psát modulární JavaScriptový kód také pro prohlížeče tím, že jej před použitím v prohlížeči převede z modulární verze do balíčku (bundle), který interně řeší načítání závislostí a funguje v prohlížeči. Díky Webpacku je tedy opět možné psát modernější JavaScriptové aplikace.

**Primárním úkolem Webpacku je zpracovat modulární JavaScriptový kód, aby fungoval ve webových prohlížečích.** Z tohoto pohledu se může zdát, že je spuštění Webpacku další úlohou pro `task runner`. Webpack jde však dále tím, že umožňuje jako moduly v JavaScriptu načítat i jiné technologie, než je JavaScript. Se správnou konfigurací je Webpack schopen načítat jako moduly například SASS, CSS nebo obrázky. Důležité však je, že výsledkem Webpacku je vždy JavaScriptový kód, proto musí být tyto moduly nějakým způsobem transformovány do JavaScriptu nebo extrahovány ze zdrojového kódu pryč. [29] [30]

#### Základní položky konfigurace Webpacku

Webpack je program napsaný v technologii Node.js, tedy je to JavaScript, který zpracovává JavaScript. Konfigurace Webpacku je Node.js modul, který je předán Webpacku jako programu. Nejdůležitější položky konfigurace jsou **entry** a **output**, které stačí k tomu, aby Webpack fungoval jako module bundler pro JavaScript. Další položky umožňují Webpacku načítat moduly i jiného typu, než je JavaScript, a provádět nad nimi transformace a úpravy. [29] [31]

**Entry** obsahuje soubory, které Webpack použije jako vstupní bod pro generování stromu závislostí modulů a jejich převedení do výstupního balíčku. Může se jednat o jeden soubor, pole souborů či JavaScriptový objekt pojmenovaných souborů.

**Output** říká, kam a jakým způsobem má Webpack vygenerovat balíček či balíčky. To, jestli bude výsledkem transformace jeden nebo více balíčků,

záleží na druhu **entry**. Soubor a pole souborů generují jeden balíček, objekt pojmenovaných souborů generuje více balíčků.

**Loadery (loaders)** jsou transformace, které Webpack aplikuje na zpracovávané moduly. **Loadery** umožňují Webpacku pracovat s jinými druhy modulů než je JavaScript. Existují **loadery** pro práci se styly, pro práci se soubory, pro práci s JavaScriptem atd. Webpack také umožňuje psát modulovatelný JavaScriptový kód v moderních standardech ECMAScript, protože může používat jako **loader** již zmíněný Babel. Další **loadery** umožňují překlad JSX syntaxe do klasického JavaScriptu nebo překlad SASS do CSS.

**Pluginy (plugins)** jsou transformace, které Webpack aplikuje na balíček. Díky **pluginům** je možné provádět dodatečné transformace a úpravy, které ovlivní celý balíček. Jak bylo již zmíněno, výsledkem Webpacku musí být JavaScript, proto je nutné vložené CSS převést na JavaScript. **Pluginy** však umožňují vyextrahovat CSS do vlastních souborů a oddělit je od JavaScriptu. Dále je možné **pluginy** použít na minifikaci balíčku, rozdělení balíčku na více částí nebo definování různých globálních proměnných.

Z výše uvedeného je patrné, že je Webpack při správné konfiguraci silným nástrojem. Díky Webpacku je možné psát moderní modulární JavaScriptový kód využívající externí knihovny jako je React nebo Redux. Také umožňuje práci s dalšími typy technologií jako SASS a to vše v rámci jednoho projektu, který je následně Webpackem zpracován a výsledkem je sada souborů, které lze předložit webovému prohlížeči pro interpretaci. Z tohoto důvodu byl vybrán Webpack pro vytvoření webové klientské aplikace místo některého ze zmíněných **task runnerů**. Zdrojový kód 6.2 je obsah **entry** souboru pro Webpack použitý ve webové klientské aplikaci. [29] [31]

### 6.2 Parametrizované generování webové klientské aplikace

Protože je jedním z business cílů vývojářů „poskytnout zdarma jednoduchou verzi klientské aplikace“, je důležité, aby tvorba nových webových klientských aplikací se specifickým obsahem a grafikou pro konkrétní událost byla co nejjednodušší. Z tohoto důvodu byl zdrojový kód rozdělen na dvě části:

**Kostrá** je zdrojový kód klientské aplikace, který je společný pro všechny generované aplikace.

**Konfigurace** je zdrojový kód, který je specifický pro konkrétní událost. Jedná se o nastavení klientské aplikace z hlediska obsahu a vzhledu. Také může modifikovat části uživatelského rozhraní.

```
1 // Pouziti SASS v JS programu
2 import './sass/main.scss';
3
4 // Polyfilly
5 import 'isomorphic-fetch';
6 import 'babel-polyfill';
7
8 // JavaScriptove knihovny
9 import 'expose-loader?#!expose-loader?jQuery!jquery';
10 import 'waypoints/lib/jquery.waypoints';
11 import ReactDOM from 'react-dom';
12 import React from 'react';
13 import { Provider } from 'react-redux';
14
15 // Konfigurace Redux store
16 import configureStore from './js/store/configureStore';
17
18 // Nacteni hlavni React komponenty
19 import App from './js/containers/AppContainer';
20 const root = document.getElementById('app');
21
22 // Vykresleni hlavni komponenty do HTML DOM
23 if (root) {
24   ReactDOM.render(<Provider store={configureStore()}>
25     <App />
26   </Provider>, root);
27 }
```

Zdrojový kód 6.2: Entry soubor pro Webpack z webové klientské aplikace.

Kostra aplikace je z hlediska generování vždy stejná, jedná se o zdrojový kód v technologiích JavaScript ES6 a ES7, JSX, SASS a HTML. Tento kód je převeden do verze interpretovatelné a spustitelné webovým prohlížečem pomocí technologie Webpack a jejích loaderů a pluginů.

Webpack byl zvolen pro generování aplikací také z toho důvodu, že je díky loaderům možné sjednotit konfiguraci do jednoho typu souboru. Webpack umí načítat do JavaScriptového kódu jako moduly mimo jiné i soubory typu **JavaScript Object Notation (JSON)**, které převede na JavaScriptové objekty. Díky rozšíření loaderu pracujícího se soubory typu SASS, je možné JSON načítat také do SASS kódu, kde je převeden na SASS proměnné. Konfiguraci v JSON lze také jednoduše načíst přímo v konfiguraci Webpacku a použít ji například pro pojmenování složky, do které je aplikace po zpracování Webpackem vložena. Tyto vlastnosti umožňují mít konfiguraci aplikace sjednocenou do jednoho typu souboru.

Konfigurace aplikace tedy obsahuje konfigurační soubory ve formátu JSON (viz. zdrojové kódy 6.3 a 6.4), pomocí kterých lze měnit obsah a vzhled aplikace, a také obrázky, jako je logo události, loga partnerů či různá pozadí a ikony. Spuštěním Webpacku (konfigurace Webpacku viz. příloha C) je spojena kostra a konfigurace a tak lze jednoduše a rychle vygenerovat novou webovou klientskou aplikaci pro konkrétní událost, kterou lze umístit na webový

```
1 {
2   "id": "58ecbc99490bf300109db0a0",
3   "type": "film",
4   "folder": "a4f",
5   "name": "Testing_web_app",
6   "secure": true,
7   "partners": false
8 }
```

Zdrojový kód 6.3: Konfigurační soubor obsahu webové klientské aplikace.

```
1 {
2   "styles": {
3     "color-black": "blue",
4     "color-white": "red",
5     "color-grey": "yellow",
6     "color-grey-darker": "cyan",
7     "color-grey-dark": "magenta"
8   }
9 }
```

Zdrojový kód 6.4: Konfigurační soubor vzhledu webové klientské aplikace.

server a která běží ve webovém prohlížeči a využívá klientské REST API ekosystému.

### 6.3 Shrnutí kapitoly

Implementace webové klientské aplikace plní následující stanovené požadavky:

- **Jednotné a znovupoužitelné uživatelské rozhraní pro všechny druhy událostí:** bylo splněno parametrizovaným způsobem generování nových webových klientských aplikací.
- **Neoptimalizovat pro mobilní zařízení, ale zařídit funkčnost všech funkcí i na mobilním zařízení:** optimalizace tedy nebyla v implementaci řešena, výsledná aplikace však funguje v jakémkoli webovém prohlížeči, tedy i mobilním.

Výsledkem fáze vývoje popsané v této kapitole je plně funkční webová klientská aplikace, která implementuje navržené uživatelské rozhraní z předchozí fáze a které stačí dodat pouze grafický návrh. Tato aplikace je vytvořena takovým způsobem, aby bylo možné jednoduše změnit její obsah a grafiku a aby tak mohla být generována opakovaně pro nové události. **Zdrojové kódy a také jednu konkrétní vygenerovanou webovou klientskou aplikaci lze nalézt na příloženém médiu.**

---

# Testování

Ve fázi testování je možné **sumativně** či **formativně** validovat návrh uživatelského rozhraní, který byl vytvořen v předchozích fázích, a to nejen na implementovaných aplikacích, ale také na prototypch uživatelského rozhraní.

Díky testování použitelnosti je možné odhalit problémy, které expert nebo designér nejsou schopni nalézt a které lze nalézt pouze pozorováním uživatele pracujícího s aplikací. Tímto způsobem se lze již v raných fázích vývoje vyhnout chybám, které by se mohly prodražit, a také lze návrh uživatelského rozhraní vést správným směrem.

Opět je vhodné, aby se testování účastnili také designéři, ale i programátoři, což jim umožní na vlastní oči vidět a pochopit chování uživatelů, kteří pracují s jejich produkty.

## 7.1 Testování wireframů

V první fázi návrhu uživatelského rozhraní klientských aplikací byly vytvořeny papírové návrhy v podobě **wireframů**, které byly postupně konzultovány s vývojáři ekosystému a snaží se dodržovat **základní pravidla použitelného návrhu**, byly kontrolovány pomocí **kognitivního průchodu** a byly testovány pomocí **pětisekundových testů**.

**Tímto způsobem byly testovány wireframy pro mobilní i webovou klientskou aplikaci.**

## 7.2 Testování mockupů mobilní klientské aplikace

Po ustálení záměrů s návrhem byly **wireframy** převedeny do interaktivní podoby. Vznikly tak jednoduché interaktivní **mockupy**, které mohly být předloženy uživatelům pro získání jejich názoru.

### 7.2.1 Způsob testování

Bylo provedeno **formativní testování** vytvořených mockupů technikou **Guerrilla testování použitelnosti** ve třech iteracích dvakrát se třemi a jednou se dvěma uživateli. Problémy objevené v jedné iteraci byly vždy před začátkem další iterace vyřešeny, čímž se předcházelo opakovanému nalézání stejných problémů.

Testování bylo vedeno způsobem, kdy byl testující ponechán, aby sám procházel prototyp a komentoval jej a emoce a myšlenky, které v něm vyvolává. Moderátor (autor této práce) pomáhal v situacích, ve kterých nebyl mockup dostatečně interaktivní, a dovysvětloval detaily, které nebylo možné do prototypu zachytit. Také byl pořízen záznam obrazovky a zvuku z průběhu testování. **Výpisky z testování jsou k nalezení v příloze D.1 a kompletní záznamy na přiloženém médiu.**

**Tímto způsobem byly testovány pouze mockupy pro mobilní klientskou aplikaci, protože uživatelské rozhraní webové klientské aplikace bude testováno po implementaci a testování použitelnosti se lépe provádí na funkční aplikaci než na prototypu.**

### 7.2.2 První iterace testování

Problémy ve všech iteracích byly definovány při vyhodnocení záznamů testování a jsou hodnoceny na škále **1 (nejméně důležitý)** až **5 (nejvíce důležitý)**. Přes toto hodnocení byla snaha vyřešit vše. První iterace testování pomohla odhalit následující problémy:

- Ikona plánování cesty je zavádějící, uživateli nic neřekne (**2**; pokusit se vybrat lepší ikonu, jinak spíše práce pro grafického designera).
- Na obrazovce *dnešní program* chybí proklik na detail performerů, což uživatele ruší, musí se doptávat (**1**; problém prototypu, ale nutno vyřešit).
- Je potřeba zvýraznit ve všech seznamech ty session, které patří do *mého programu* (**5**; důležitá informace, je potřeba graficky upravit).
- Obrazovku *můj program* by bylo vhodnější rozdělit po dnech a trackech, ne pouze po trackech, pro uživatele je nepřehledná (**3**; vhodné opravit pro lepší přehlednost).
- V sekci *můj dnešní program* na obrazovce *dnešní program* uvést i track, na kterém session probíhá (**5**; důležitá informace, nutno přidat).
- V *detailu novinky* přidat rozdílnou vizualizaci v případě, že uživatel označil/neoznačil novinku jako „líbí se mi“ (**1**; spíše problém prototypu, ale ruší uživatele).

- V *detailu novinky* přidat možnost označení komentáře jako „líbí se mi“ a řazení komentářů podle tohoto hodnocení (**3**; hodí se pro lepší přehlednost, volnost uživatele).
- V *detailu novinky* změnit terminologii formuláře pro vytvoření nového komentáře (**4**; není přesně pochopitelné, co formulář dělá).
- Upravit zvýraznění *session*, které má uživatel v *mém programu*, protože na *časové ose* je zvýraznění zavádějící (**5**; pomohlo by sjednotit zvýraznění v celé aplikaci).
- Po každém vyplněném dotazníku upozornit uživatele na tlačítko pro odeslání dotazníků (**4**; například skrytím a znovu zobrazením).
- Na obrazovce *diskuze* změnit terminologii formuláře pro přidání příspěvku (**4**; stejné jako u komentářů novinek).
- Přidat k příspěvkům možnost odpovědi a vlákna odpovědí (**5**; zvýší přehlednost).
- Přidat hodnocení „líbí se mi“ k příspěvkům a odpovědím a možnost řazení podle tohoto hodnocení, což by mělo zvětšit přehlednost obsahu (**5**; zvýší přehlednost, usnadní hledání zajímavých informací).

### 7.2.3 Druhá iterace testování

Problémy nalezené při první iteraci testování byly vyřešeny a změny byly promítnuty do mockupu. Uživatelé testující v druhé iteraci tak nebyli vyrušováni již známými problémy a mohli se zaměřit na menší problémy.

V druhé iteraci byly objeveny problémy, které lze vyřešit následujícími úpravami:

- Ikona plánování trasy je stále zavádějící (**2**; raději přidat i text nebo ponechat na grafickém designérovi).
- Přidat plánování trasy z pinu na interaktivní mapě (**3**; mohlo by se hodit).
- Odstranit řazení tracků dle abecedy v jejich seznamu. Řadit dle pořadí nastaveného v administraci (**3**; tracků většinou není tak hodně, aby bylo nutné je řadit dle abecedy).
- Řazení tracků v *časové ose* dle množství položek z *mého programu* (**3**; zajímavý nápad, mohlo by zvýšit přehlednost).
- Přesunutí možnosti skrytí všech tracků v *seznamu tracků* z hlavičky, kde je zavádějící (**5**; například pod hlavičku nad seznam).

- Po skrytí tracku informovat uživatele o tom, co se právě stalo a jak to může změnit (**5**; důležitá informace, těžko se přidá do prototypu, lepší použít systémovou notifikaci).
- Přidat proklik na *detail tracku* z *časové osy* (**4**; může se hodit na hudebních festivalech).
- Upravit proporce textu v *diskuzi* (**3**; obsah zprávy větší, datum vytvoření menší, více prostoru mezi příspěvky).
- Navrhnout vizualizace odpovědi na příspěvek v *diskuzi* (**5**; zapomenuto, nutno přidat).
- Navrhnout pop-up nastavení jména do *diskuze* a komentářů v *detailu novinky* (**5**; opět zapomenuto).
- Přidat informace o konci session všude, kde je potřeba (**5**; důležitá informace pro přehled v programu).

### 7.2.4 Třetí iterace testování

Výše uvedené úpravy byly promítnuty do vytvořeného mockupu. Ve třetí iteraci už testující uživatelé nenalezali žádné závažné problémy, ale spíše často hodnotili návrh dle svých osobních preferencí. Z tohoto důvodu už nebyly prováděny další změny navrženého rozhraní.

### 7.2.5 Shrnutí výsledků testování

První iterace odhalila největší problémy prototypu a následující iterace pomohly prototyp a tím i návrh uživatelského rozhraní postupně doladit. Na většině problémů odhalených v první iteraci se shodli dva ze tří testujících uživatelů. Ve druhé iteraci nalézali uživatelé méně zásadní problémy, které byly přesto zapracovány do návrhu, protože by mohly ovlivnit kvalitu práce s aplikací. Pomocí výsledků této iterace bylo možné doladit mockup do finální podoby. Ve třetí iteraci byly zkontrolovány úpravy z předchozích iterací a uživatelé už neobjevili žádné další zásadnější problémy, díky čemuž bylo možné testování a úpravy návrhu ukončit. **Verze prototypů ze všech tří iterací lze nalézt na příloženém médiu.**

Obecně lze říci, že se návrh nijak zvláště neměnil, nebyly vytvořeny či rušeny žádné obrazovky a ani nebyly přidány žádné zásadní funkce. Spíše byly doladěny nesrozumitelné a zavádějící části a chybějící důležité informace. Testování pomohlo návrh doladit a už první iterace byla velice přínosná. Při každé iteraci byly objeveny jiné chyby, což podporuje teorii, že je lepší testovat iterativně s méně lidmi, než jednorázově s více lidmi.



## 7.3 Testování použitelnosti webové klientské aplikace

Pro testování použitelnosti implementace webové klientské aplikace byla v administraci ekosystému vytvořena testovací událost, která byla naplněna daty. Pro tuto událost byla vygenerována nová testovací aplikace, která byla použita pro testování použitelnosti.

### 7.3.1 Způsob testování

Narozdíl od klasického testování použitelnosti nebyl kladen důraz na oddělení designéra od testujícího a nezávislé pozorování průběhu testování v jiné místnosti. *Guerilla testování použitelnosti* tyto detaily nevyžaduje a je vhodné v případech, kdy nejsou prostředky a čas na klasické testování použitelnosti. Testování bylo provedeno ve třech iteracích dvakrát se třemi a jednou se dvěma uživateli. Problémy objevené v jedné iteraci byly před začátkem další iterace vyřešeny, aby se předešlo opakovanému nalézání stejných problémů.

Testování bylo řízeno moderátorem (autorem této práce), který zadával testujícímu uživateli úkoly k vyřešení. Bylo připraveno několik scénářů, které měl testující za úkol splnit. Testující uživatel dostal také prostor se k průběhu testování a celé aplikaci vyjádřit. Pokaždé byl pořízen záznam obrazovky a zvuku z průběhu testování. **Výpisky z testování jsou k nalezení v příloze D.2 a kompletní záznamy na přiloženém médiu.**

### 7.3.2 Scénáře

Scénáře byly vytvořeny z pohledu *person* vytvořených v kapitole *Analýza* a měly by vést k plnění jejich cílů.

1. Josef chce zjistit, jestli je v areálu festivalu Wi-Fi připojení, aby mohl přidat fotografie na sociální síť Instagram.
2. Josef potřebuje pomocí aplikace zjistit, kde jsou v areálu festivalu nejbližší záchody.
3. Adéla by si ráda vytvořila vlastní program z filmů, které ji zaujmou, a následně chce zkontrolovat, jestli jí program nekoliduje.
4. Adéla si chce vytvořit vlastní program ze severských filmů, které budou na festivalu promítány.
5. Josef potřebuje zjistit adresu konkrétního kina.
6. Adéla potřebuje zjistit, v jakém kině se bude promítat film, který si přidala do svého programu, a jaká je přesná adresa tohoto kina.

### 7.3.3 První iterace testování

Problémy ve všech iteracích byly definovány při vyhodnocení záznamů testování a jsou hodnoceny na škále **1 (nejméně důležitý)** až **5 (nejvíce důležitý)**. Přes toto hodnocení byla snaha vyřešit všechny problémy. V průběhu první iterace byly objeveny nejzávažnější problémy, které ztěžovaly testujícím uživatelům použití aplikace. Následuje jejich výčet:

- Chybí možnost mazání položek přímo ze sekce *můj program* (**5**; nutno přidat tlačítko na odebrání).
- Formulář vyhledávání po zmáčknutí klávesy *enter* přenačte celou stránku, místo aby vyhledával (**5**; odebrat automatické odeslání na stisk klávesy *enter*).
- Chybí odkaz na adresu tracku v mapě z každé položky popisující konkrétní *session* (**5**; nutno přidat).
- Chybí odkaz na adresu tracku v mapě z *nejbližšího programu* (**4**; mohlo by se hodit).
- Chybí filtrování seznamu performerů přes jejich tagy (**5**; bylo uvedeno v návrzích, nutno přidat).
- Je nutné zlepšit chování filtru performerů (**4**; po stisku *enter* zaměřit (*focus* efekt) na formulář).
- *Vyzkoušet možnost rychlého přidání performerera do mého programu přímo ze seznamu performerů, pokud má pouze jednu session* (**3**; mohlo by se hodit)

### 7.3.4 Druhá iterace testování

Před začátkem druhé iterace testování byly vyřešeny problémy nalezené v první iteraci. Testující uživatelé následně opět našli několik problémů, které stály za vyřešení. Také bylo nutné otestovat možnost rychlého přidání performerera do *mého programu*. Následuje výčet problémů nalezených v druhé iteraci:

- Uživatelé nechápou, že je ikona pinu u položek v *mém programu* odkaz na mapu (**5**; změnit na název tracku s ikonou).
- Chybí barevné odlišení položek, které má uživatel v *mém programu* (**4**; zvýší přehlednost).
- Na stránce s *performery* je špatný poměr prostoru pro seznam performerů a detail performerera (**2**; zvýší přehlednost).
- V seznamu performerů chybí u položek tagy, podle kterých lze seznam filtrovat (**5**; důležitá informace pro pochopení vyhledávání, přidat).

- Text v poli pro vyhledávání nedostatečně informuje o možnostech vyhledávání (**5**; důležitá informace pro pochopení vyhledávání, nutno změnit *placeholder*).
- Vyhledávání v informacích není dostatečně pochopitelné (**2**; přidat vhodný *placeholder* text do pole vyhledávání).
- Rychlé přidání performerera je nekonzistentní, není jasné, co provádí a proč chybí, pokud má performer více session (**4**; zkusit odebrat a znovu testovat bez).

### 7.3.5 Třetí iterace testování

Problémy nalezené v druhé iteraci bylo nutné vyřešit a otestovat. Také bylo potřeba vyřešit použití funkce rychlého přidání performerera do *mého programu*, a proto byla provedena také třetí iterace.

Uživatelé v této iteraci nenalezli žádné závažné problémy a spíše komentovali aplikaci dle svých preferencí. Problémy nalezené v předchozích iteracích už je nerozptylovaly a práce s aplikací pro ně byla víceméně bezproblémová. Z těchto důvodů už nebyly prováděny další iterace.

### 7.3.6 Shrnutí výsledků testování

Již první iterace testování použitelnosti pomohla odhalit závažné problémy v návrhu uživatelského rozhraní aplikace. Následující iterace poté pomohly otestovat úpravy uživatelského rozhraní a doladit ho tak, aby poskytovalo uživateli kvalitní uživatelský zážitek. Uživatelé se často na problémech shodli a proto nebylo těžké odhadnout, které je potřeba vyřešit.

Uživatelé také navrhli funkci pro rychlé přidání performerera do *mého programu* v případě, že má pouze jednu session. Tato funkce se na první pohled zdála být výhodná, nakonec se ale ukázalo, že je zavádějící a je lepší ji nepoužít. V případě, že měl performer více session, vedla na stejná dvě kliknutí, která bylo nutné provést i bez ní. V případě, že měl performer pouze jednu session, nebylo dostatečně jasné, jaký je rozdíl mezi session a performerem a která z těchto dvou entit se přidává do *mého programu*, což bylo matoucí.

Obecně lze říci, že se uživatelské rozhraní aplikace a funkce, které aplikace poskytuje, nijak moc nezměnily. Byly však doladěny zásadní problémy a nedostatky, které sice neohrožovaly funkčnost aplikace, ale použitelnost se po jejich vyřešení velice zlepšila.

### 7.3.7 Další možnosti vylepšení

Již v analýze byly objeveny některé funkce, které by mohla webová klienta aplikace poskytovat, aby lépe plnila uživatelské cíle. Aplikace však musí plnit

nejen uživatelské cíle, ale také business cíle vývojářů a pořadatelů, což sice aktuální stav splňuje, ale uživatelské cíle jsou citelně potlačeny. Potvrzením nedostatečného zaměření na uživatelské cíle je fakt, že si uživatelé v průběhu testování často stěžovali na věci, které byly záměrně z návrhu vynechány.

V případě, že by se aplikace stala plnohodnotným řešením, by bylo vhodné provést také následující úpravy:

- Optimalizovat uživatelské rozhraní pro mobilní zařízení, protože by aplikace měla sloužit také pro platformy, které mobilní klientská aplikace nepodporuje
- Přidat seznam tracků a jejich detaily, protože nalézt informace o tracku ve stávající verzi je obtížné (záměrně).
- Přidat možnost vytvoření uživatelského profilu a synchronizace mého programu mezi webovou a mobilní klientskou aplikací, protože scénář použití webové klientské aplikace je z pohledu uživatele jiný, než u mobilní klientské aplikace. Uživatelé by webovou klientskou aplikaci použili spíše k plánování vlastního programu z pohodlí domova před návštěvním událostí.

### 7.4 Shrnutí kapitoly

Výsledkem fáze vývoje popsané v této kapitole jsou **sumativně otestované wireframy** uživatelského rozhraní mobilní a webové klientské aplikace, které byly vytvořeny v předchozích fázích. Dále jsou výsledkem **mockupy** uživatelského rozhraní mobilní klientské aplikace, které byly testovány pomocí techniky **Guerilla testování použitelnosti** a opraveny. V neposlední řadě je výsledkem fáze technikou **Guerilla testování použitelnosti** otestovaná implementace uživatelského rozhraní webové klientské aplikace, která byla opět opravena. Obě testování použitelnosti odhalila závažné i méně závažné problémy, díky čemuž bylo možné zvýšit kvalitu použitelnosti a zlepšit tak uživatelský zážitek. **Opravené návrhy uživatelských rozhraní lze nalézt v příloze E a také na přiloženém médiu v podobě mockupů pro mobilní klientskou aplikaci a finální implementace pro webovou klientskou aplikaci.**

Z testování prototypu uživatelského rozhraní mobilní klientské aplikace a testování implementace uživatelského rozhraní webové klientské aplikace je patrný rozdíl mezi těmito dvěma stavy návrhu. Testování prototypů je obtížnější a náročnější na řízení a také na uživatelovu představivost. Moderátor musí často do průběhu testování zasahovat a pomáhat uživateli. Testování implementace je přímočaré, protože se jedná o praktické použití uživatelského rozhraní a není nutné ani vhodné do testování více zasahovat.

---

# Vliv uživatelsky zaměřeného návrhu

## 8.1 Nevýhody

Nevýhody uživatelsky zaměřeného návrhu se zdají být jasné z průběhu praktické části práce. Je jimi zejména časová a finanční náročnost. Řádně provedený uživatelský průzkum pomocí přímých rozhovorů s uživateli může zabrat desítky hodiny pro realizaci, pokud jeden rozhovor trvá průměrně hodinu. Další hodiny může trvat vyhodnocení výsledků rozhovorů. Uživatele je nutné nějakým způsobem motivovat, což znamená nemalou finanční investici, pokud se jako designéři nebudeme pohybovat pouze v okruhu svých přátel. Testování s uživateli například pomocí testování použitelnosti je podobně časově i finančně náročné. Je nutné domluvit schůzky s uživateli, provést samotná testování a výsledky vyhodnotit, což může zabrat v průměru desítky hodin práce. Opět je nutné uživatele nějakým způsobem motivovat, a tak provést finanční investici.

## 8.2 Přínosy

Jak vyplývá z teoretického rozboru, výše zmíněné nevýhody jsou značně kompenzovány přínosy, které uživatelsky zaměřený návrh má. Počáteční vyšší investice se sice může zdát jako nevýhoda, ale pokud provedeme uživatelský průzkum řádně, můžeme si být jisti, že vytvoříme produkt, který bude pomáhat plnit cíle uživatelů. V případě vynechání uživatelského průzkumu jsme odsouzeni k hádání, které může znamenat investici do tvorby softwaru, který nebude chtít nikdo používat a bude ztrátový. Podobně to platí také o testování s uživateli. Pokud bychom vynechali testování s uživateli, ochudili bychom se o možnost již v raném stádiu odhalit chyby a problémy, které jsme schopni odhalit pouze sledováním uživatelů při práci. Tím, že budeme testovat s uživa-

vateli, jsme schopni odhalit špatně navržené části softwaru například již ve fázi návrhu, což může pomoci nejen zlevnit a zrychlit vývoj, ale také vytvořit použitelný software. O použitelnosti softwaru nerozhodují implementované funkce, ale kvalita uživatelského rozhraní. Vytvořením nepoužitelného softwaru se ochuzujeme o uživatele, kteří rozhodují o jeho úspěšnosti či neúspěšnosti.

Pokud není ani jeden z výše uvedených argumentů dostačující, pak se lze odkázat na práci Toma Gilba, podle které se každý \$1 investovaný do *User Experience designu* vrátí přibližně \$10–\$100. [7]

### 8.3 Zhodnocení vlivu na praktickou část práce

Pokud bychom měli srovnávat původní a nově vytvořenou verzi uživatelských rozhraní (lze srovnávat záznam původních návrhů aplikací ekosystému na přiloženém médiu a obsah přílohy E), pak došlo ke mnoha změnám, které by měly vést k lepšímu plnění uživatelských cílů. V případě mobilní klientské aplikace došlo zejména k rozdělení programu, který v podobě *dnešního programu, mého programu* a *časové osy* poskytuje lepší přehled v průběhu události. Také byly výrazně přepracovány *novinky*, které kladou větší důraz na sociální funkce a přehlednost. I v *diskuzi* byly rozšířeny sociální funkce a byla zlepšena přehlednost. Nový návrh klade větší důraz na použití mobilní klientské aplikace pro orientaci na události vyzdvižením funkce plánování trasy. Použitelnost byla zvýšena také zjednodušením odesíláním *dotazníků*, což by mělo zpříjemnit již tak neoblíbenou aktivitu. V neposlední řadě je jednou z výrazných změn poskytnutí volnosti uživateli možností volby výchozí obrazovky.

Jak bylo zmíněno již v kapitole Testování, je stávající verze webové klientské aplikace mířena spíše na business cíle vývojářů, což úplně neplní její scénář použití definovaný uživateli. Přesto bylo i zde provedeno nemálo změn, které by měly zvýšit použitelnost a lépe plnit uživatelské cíle. *Domovská stránka* se více zaměřuje na program a práci s ním. *Časová osa* byla změněna na vertikální, protože vertikální pohyb je pro uživatele na myši (pohyb pomocí kolečka myši) daleko přirozenější. Byla zvýšena přehlednost *seznamu performerů* a poskytnuto jejich vyhledávání, což by mělo zjednodušit tvorbu vlastního programu. *Mapy* jsou více interaktivní a přes absenci seznamu tracků poskytuje webová klientská aplikace uživateli alespoň informace o jejich adresách.

Vliv uživatelsky zaměřeného návrhu na praktickou část práce byl tedy bezesporu kladný. **Uživatelský průzkum** pomohl stanovit požadavky vycházející z cílů uživatelů a business cílů ostatních zainteresovaných stran. Tyto požadavky následně pomohly definovat funkcionalitu klientských aplikací a také návrh uživatelských rozhraní, která by pomocí plnění požadavků měla pomáhat plnit stanovené uživatelské a business cíle. Bez uživatelského průzkumu bychom byli odsouzeni k hádání. Techniky návrhu následně pomohly vytvo-

řit **prototypy**, na kterých bylo možné odladit základní nedostatky ve stádiu, kdy jsme ještě neprovedli nemalé časové investice do implementace. Už návrhy v tomto stavu by měly plnit uživatelské cíle daleko lépe, než původní verze uživatelského rozhraní. Samotná **implementace** byla pouhým převedením návrhu uživatelského rozhraní a funkcionality do hotového softwaru pro danou platformu v podobě webové klientské aplikace. Přesto se snaží plnit business cíle vývojářů tím, že poskytuje jednoduchý způsob generování verzí pro nové události. **Testování použitelnosti** s uživateli následně pomohlo odhalit problémy v návrhu uživatelských rozhraní jak mobilní, tak webové klientské aplikace, čímž jsme vytvořili uživatelská rozhraní, která jsou pro uživatele dobře použitelná (vyplývá z poslední iterace testování a teoretického rozboru) a která by měla pomáhat plnit jeho cíle nejlepším možným způsobem. **Takto vytvořená uživatelská rozhraní poskytují kvalitnější uživatelský zážitek, což plní cíle uživatelsky zaměřeného návrhu v podobě User Experience design.**





---

# Závěr

Tato práce se zabývala problematikou uživatelsky zaměřeného návrhu a jeho vlivem na vývoj softwaru. Poznatky získané v teoretické části práce byly aplikovány na návrh uživatelských rozhraní mobilní a webové aplikace ekosystému pro události. Cílem práce bylo pochopit význam uživatelsky zaměřeného návrhu, prakticky jej použít a následně zhodnotit, jaký vliv měl takto zaměřený návrh na průběh a výsledky práce, jaké jsou jeho přínosy a nevýhody a jaké závěry z toho plynou. Mezi cíle práce také patřilo implementovat webovou aplikaci.

V kapitolách *User Experience design* a *User-Centered engineering* byl proveden rozbor uživatelsky zaměřeného návrhu a jeho vlivu na vývoj softwaru. V rámci kapitol byl definován obecný model vývoje, který byl použit v praktické části. Taktéž zde byly rozebrány techniky, které lze pro uživatelsky zaměřený návrh v rámci vývoje softwaru využít, a některé z nich byly v praktické části použity. V kapitole *Ekosystém aplikací pro události* je čtenář seznámen se zadaným ekosystémem, který byl použit v praktické části práce. V rámci praktické části popsán v kapitolách *Analýza, Návrh uživatelského rozhraní, Implementace webového klientů aplikace a Testování* byl analyzován kontext a definovány požadavky na aplikace pro události. Tyto požadavky byly transformovány do funkcí a následně do konkrétních návrhů uživatelských rozhraní realizovaných jako prototypů a také jako finální implementace v případě webové aplikace. Vytvořená uživatelská rozhraní v podobě prototypů a implementace byla řádně otestována, díky čemuž bylo možné vytvořit taková rozhraní, která jsou pro uživatele lépe použitelná a navíc pomáhají plnit jejich cíle tím, že poskytují adekvátní formu, obsah a chování. V poslední kapitole *Vliv uživatelsky zaměřeného návrhu* byl shrnut vliv uživatelsky zaměřeného návrhu na praktickou část práce a poznatky z toho plynoucí.

Výsledkem práce jsou, kromě souhrnu poznatků o uživatelsky zaměřeném návrhu a jeho vlivu na vývoj softwaru (viz. kapitola *Vliv uživatelsky zaměřeného návrhu*), také úspěšně navržená a otestovaná uživatelská rozhraní mo-

bilní a webové aplikace ekosystému pro události (viz. příloha E a příložené médium), která byla vytvořena se zřetelem na poznatky získané v teoretické části práce. Takto vytvořená uživatelská rozhraní plní stanovené požadavky (pokud stav návrhu dovoluje, mobilní aplikace nebyla implementována) a měla by být pro uživatele lépe použitelná, pomáhat plnit jeho cíle a poskytovat kvalitní a pozitivní uživatelský zážitek. Díky uživatelsky zaměřenému návrhu je tedy možné vytvořit software, se kterým se bude uživateli dobře pracovat a který tak bude úspěšný. Dalším z výsledků práce je také plně funkční webová aplikace (viz. příložené médium), pro kterou již stačí vytvořit pouze grafický návrh. Z těchto informací plyne, že se cíle práce podařilo splnit.

Pokud by byla ochota ze strany vývojářů ekosystému měnit jeho stávající architekturu a strukturu, bylo by možné použitelnost a uživatelský zážitek ještě více zlepšit. V takovém případě by bylo vhodné rozdělit analýzu na konkrétní druhy událostí, provést ji pro každý druh nezávisle a následně ze získaných informací usoudit, zda by nebylo vhodnější kostry aplikací rozdělit pro každý druh události samostatně. Z praktické části práce totiž vyplývá, že by se uživatelská rozhraní v takovém případě mohla lépe zaměřovat na konkrétní cílovou skupinu uživatelů, protože každá událost vyžaduje trochu jiné funkce a některé funkce jsou pro ni zbytečné.

Z uživatelského průzkumu vyplynulo, že scénáře použití mobilní a webové aplikace jsou rozdílné. V tomto případě by bylo vhodné webovou aplikaci přizpůsobit zejména pro tvorbu vlastního programu a plánování návštěvy události z domova, protože tímto způsobem ji chtějí uživatelé používat. Následně by bylo vhodné poskytnout jednoduché řešení synchronizace programu, který si uživatel vytvořil ve webové aplikaci, do mobilní aplikace.

Jak bylo uvedeno výše, zmíněné úpravy by pomohly zlepšit uživatelský zážitek tím, že by lépe podporovaly cíle uživatelů (zejména díky synchronizaci programu) a poskytovaly lepší použitelnost (vynecháním funkcí nedůležitých pro konkrétní druh události).

---

## Literatura

- [1] NORMAN, D. A.: *Design pro každý den*. Dokořán, první vydání, 2010, ISBN 978-80-7363-314-1.
- [2] RICHTER, M.; FLÜCKIGER, M.: *User Centered Engineering: Creating Products for Humans*. Springer, 2014, ISBN 978-3-662-43989-0.
- [3] NORMAN, D. A.: *Emotional Design: Why we love (or hate) everyday things*. Basic Books, 2004, ISBN 0-465-05135-9.
- [4] COOPER, A.; REIMANN, R.; CRONIN, D.; aj.: *About Face: The Essentials of Interaction Design*. John Wiley & Sons, Inc., čtvrté vydání, 2014, ISBN 978-1-118-76658-3.
- [5] NEISSER, U.: *Cognitive Psychology*. Century psychology series, Appleton-Century-Crofts, 1967.
- [6] MATHIS, L.: *Designed for Use: Create Usable Interfaces for Applications and the Web*. Pragmatic Bookshelf, 2011, ISBN 978-1-93435-675-3.
- [7] GILB, T.: *Principles Of Software Engineering Management*. Pearson Education Ltd, 1988, ISBN 0-201-19246-2.
- [8] KADLEC, V.: *Agilní programování: Metodiky efektivního vývoje softwaru*. Computer Press, a.s., první vydání, 2004, ISBN 80-251-0342-0.
- [9] CUNNINGHAM, W.: Manifesto for Agile Software Development [online]. [cit. 2017-03-05]. Dostupné z: <http://agilemanifesto.org/>
- [10] GOTHELF, J.: *Lean UX*. O'Reilly Media, Inc., první vydání, 2013, ISBN 978-1-449-31165-0.
- [11] BULEY, L.: *The User Experience Team of One: A Research and Design Survival Guide*. Rosenfeld Medias, 2013, ISBN 978-1-933820-18-7.

- [12] NIELSEN, J.: 10 Usability Heuristics for User Interface Design [online]. [cit. 2017-03-09]. Dostupné z: <https://www.nngroup.com/articles/ten-usability-heuristics/>
- [13] KRUG, S.: *Nenutte uživatele přemýšlet!* Computer Press, a.s., druhé vydání, 2006, ISBN 80-251-1291-8.
- [14] FIELDING, R. T.: *Architectural Styles and the Design of Network-based Software Architectures*. Dizertační práce, University of California, 2000.
- [15] Refsnes Data: HTML Introduction [online]. [cit. 2017-04-20]. Dostupné z: [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)
- [16] W3C: Document Object Model [online]. [cit. 2017-04-20]. Dostupné z: <https://www.w3.org/DOM/>
- [17] Refsnes Data: JavaScript HTML DOM [online]. [cit. 2017-04-20]. Dostupné z: [https://www.w3schools.com/js/js\\_htmlDOM.asp](https://www.w3schools.com/js/js_htmlDOM.asp)
- [18] Refsnes Data: CSS Introduction [online]. [cit. 2017-04-20]. Dostupné z: [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)
- [19] Refsnes Data: CSS3 Media Queries [online]. [cit. 2017-04-20]. Dostupné z: [https://www.w3schools.com/css/css3\\_mediaqueries.asp](https://www.w3schools.com/css/css3_mediaqueries.asp)
- [20] CATLIN, H.; WEIZENBAUM, N.; EPPSTEIN, C.: Sass (Syntactically Awesome StyleSheets) [software]. Cit. 2017-04-20].
- [21] Refsnes Data: JavaScript Introduction [online]. [cit. 2017-04-20]. Dostupné z: [https://www.w3schools.com/js/js\\_intro.asp](https://www.w3schools.com/js/js_intro.asp)
- [22] Refsnes Data: JavaScript Versions [online]. [cit. 2017-04-20]. Dostupné z: [https://www.w3schools.com/js/js\\_versions.asp](https://www.w3schools.com/js/js_versions.asp)
- [23] Refsnes Data: AJAX Introduction [online]. [cit. 2017-04-20]. Dostupné z: [https://www.w3schools.com/js/js\\_ajax\\_intro.asp](https://www.w3schools.com/js/js_ajax_intro.asp)
- [24] Babel team and community: Babel is a JavaScript compiler [software]. [cit. 2017-04-21]. Dostupné z: <http://babeljs.io/>
- [25] Facebook Inc.: React: A JavaScript Library for Building User Interfaces [software]. [cit. 2017-04-21]. Dostupné z: <https://facebook.github.io/react/>
- [26] ABRAMOV, D.: Redux [software]. [cit. 2017-04-21]. Dostupné z: <http://redux.js.org/docs/introduction/>
- [27] CHACON, S.: *Pro Git*. CZ.NIC, z. s. p. o., první vydání, 2009, ISBN 978-80-904248-1-4.

- [28] JANČA, M.: Webpack - the module bundler | 3. část (+ yarn) [online]. [cit. 2017-04-21]. Dostupné z: <https://medium.com/p/2c27a8c2bbc1>
- [29] Webpack team and community: Webpack: Bundle Your Assets [software]. [cit. 2017-04-21]. Dostupné z: <https://webpack.js.org/>
- [30] JANČA, M.: Webpack - the module bundler [online]. [cit. 2017-04-21]. Dostupné z: <https://medium.com/p/933ae83101db>
- [31] JANČA, M.: Webpack - the module bundler | 2. část [online]. [cit. 2017-04-21]. Dostupné z: <https://medium.com/p/fbbb2805a55a>



## Seznam použitých zkratk

- AJAX** Asynchronous JavaScript and XML
- API** Application Programming Interface
- CSS** Cascading Style Sheets
- DOM** Document Object Model
- ES** ECMAScript
- GUI** Graphical User Interface
- HTML** Hyper Text Markup Language
- HTTP** HyperText Transfer Protocol
- ISO** International Organization for Standardization
- IT** Informační technologie
- JSON** JavaScript Object Notation
- REST** Representational State Transfer
- SASS** Syntactically Awesome StyleSheets
- SME** Subject Matter Experts
- UCE** User-Centered engineering
- UI** User Interface
- UX** User Experience
- UxD** User Experience design
- XML** eXtensible Markup Language





---

# Uživatelský průzkum

## B.1 Respondent č.1

### Obecné informace

Studentka ČVUT, 23 let. Pravidelně navštěvuje hudební i filmové festivaly a zajímá se o kulturní události.

### Informace získané v průzkumu

- Chodí na festivaly kvůli lidem, kvůli hudbě, poslechnout si hudbu a užít si hezký víkend. Chodí na konkrétní promítání na filmovém festivalu. Na filmové festivaly nechodí tolik kvůli lidem.
- Program na festivalu ji zajímá, ale hudební festivaly tolik neplánuje. Filmové festivaly je podle ní lepší plánovat dopředu, protože to je jiný koncept než hudební festivaly.
- Hudební festivaly mají výhodu, že pokud ji něco nezajímá, tak může odejít, změnit plány nebo se jen věnovat přátelům.
- Filmové festivaly bere spíše tím způsobem, jako by šla do kina. Také ji více zajímá, jak jdou jednotlivé filmy po sobě.
- Je pro ni rozdíl mezi tím, jestli jde na festival nebo na konkrétní představení.
- Použila papírový program. Použila ho na přehled o kapelách, kde kdo hraje, na hudebním festivalu. Na filmovém festivalu věci plánuje hodně dopředu a proto takový program tolik nepotřebuje.
- Papírový program podle ní musí obsahovat všechny nejdůležitější informace, které potřebuje pro hladkou návštěvu festivalu.

- Mobilní aplikaci by použila, ale musí mít minimálně všechny informace, co papírový program a něco navíc.
- Líbily by se jí realtime notifikace o změnách v programu a různých dalších důležitých informacích, které nebyly známy před začátkem festivalu a vznikly až v průběhu.
- Poskytnutí zpětné vazby by spíše nevyužila. Nerada vyplňuje dotazníky. Kdyby to šlo jednoduše, tak by to asi udělala. Obecně čím jednodušší způsob, tím spíše by jí nevadilo zpětnou vazbu poskytnout.
- Hodnotit vystoupení kapely jí nedává smysl, hodnocení filmů by dělala, kdyby to mělo smysl třeba pro výběr nejlepšího filmu.
- Pokud by byla možnost výběru mezi webovou a mobilní aplikací, tak by si vybrala mobilní. Nedává jí smysl použít webovou aplikaci přímo na festivalu.
- Pokud by si mohla vytvořit program ve webové aplikaci a následně jej synchronizovat do mobilní, tak by takovou funkci určitě využila. V takovém případě by podle ní mělo i smysl hodnotit přímo filmy v aplikaci.
- Socializaci v aplikaci by nevyužila, není to podle ní to, co by měla aplikace nutně obsahovat.
- Využila by nějaký chat či diskuzi, kde by mohla rychle kontaktovat organizátory, pokud by to nutně potřebovala.

## B.2 Respondent č.2

### Obecné informace

Studentka architektury, 23 let. Pravidelný návštěvník hudebních festivalů se zájem o hudbu a kulturu obecně.

### Informace získané v průzkumu

- Na festivaly chodí kvůli lidem, kvůli hudbě, nemusí chodit na jednotlivé koncerty, vyhovuje jí větší koncentrace vystoupení.
- Už jela na festival i tak, že vůbec nic neplánovala, ale je to jen na určité festivaly.
- Na filmových festivalech by určitě vybírala, na které filmy půjde, a plánovala si program.
- Použila papírový program. Zajímá ji časový program festivalu, co kde je, informace o interpretech atd.

- Na hudebním festivalu ji více zajímá časová osa, na filmovém by ji více zajímaly konkrétní promítání a jak z nich vytvořit svůj vlastní program.
- Mobilní aplikaci by použila, líbila by se jí možnost vytvoření vlastního programu a informace z něj plynoucí (notifikace o začátcích představení atd.).
- Na papírovém programu ji vadí občasná nepřehlednost časového programu jednotlivých představení, což si myslí, že lze v aplikaci udělat lépe.
- Líbila by se jí možnost plánování trasy a to hlavně u filmových festivalů, které jsou navíc někde, kde to nezná.
- Využila by poskytnutí zpětné vazby, baví ji vyplňovat dotazníky. Líbilo by se jí, kdyby šlo poskytnout zpětnou vazbu jednoduchým způsobem.
- Nestahovala by si aplikaci, pokud by si nebyla jistá, že na festival půjde.
- Hodnocení filmů jí smysl dává, hodnocení hudební podle ní smysl nemá. Mít hodnocení pouze v mobilní aplikaci podle ní nemá takový smysl, lepší by bylo, kdyby se tyto informace také zobrazovaly třeba v programu, který má festival na svých webových stránkách.
- Nestáhla by si mobilní aplikaci, pokud by šla jen na jedno představení.
- Určitě by spíše použila mobilní než webovou aplikaci, pokud by si mohla vybrat, a to hlavně kvůli datovém připojení.
- Propojení webové a mobilní aplikace s možností synchronizace svého vytvořeného programu by využila. Líbí se jí možnost vytvořit si program z pohodlí domova na počítači s velkou obrazovkou. Na festivalu by pak už pracovala pouze s mobilní aplikací.
- Pokud by měla možnost pomocí diskuze či chatu rychle komunikovat s organizátory a zjistit nějaké důležité informace, tak by ji využila.

## B.3 Respondent č.3

### Obecné informace

Student ČVUT, 24 let. Pravidelně navštěvuje různé kulturní události, mezi které patří mimo jiné i hudební a filmové festivaly.

### Informace získané v průzkumu

- Na hudební festivaly chodí na muziku, na kapely a za přáteli. Na filmové festivaly chodí na konkrétní filmy.
- Vidí rozdíl mezi návštěvou filmového a hudebního festivalu. Na filmový by šel objevovat.
- Filmový by plánoval, protože nechce odcházet z kina, když ho film nebaví, protože už si jej zaplatil. Na hudebním může odejít, pokud ho něco nebaví, takže tolik neplánuje.
- Na hudebním festivalu má vybraných pár kapel dopředu a zbytek dovyplní podle aktuální situace.
- Na hudební jde přímo na ten festival, na filmový jde na konkrétní promítání.
- Na hudebním festivalu ho před festivalem zajímá zejména kdo tam bude, kde to bude, kolik to bude stát, kde bude bydlet. Jde spíše o odpočinek.
- Na filmovém festivalu ho před festivalem zajímá primárně co tam bude za filmy, ale i další informace jako na hudebním.
- V papírovém programu nejvíce hledá, co zrovna hraje. Zajímá ho průběh kapel, kdo hraje na které stagi. Zajímá ho hlavně časová osa.
- Také v papírovém programu pravidelně používá mapu areálu na hudebním festivalu.
- Obecně papírový program používá pro získání informací o organizaci festivalu.
- Mobilní aplikaci už použil, také ji použil na program, na časovou osu, a přijde mu lepší časová osa v telefonu, protože je interaktivní.
- Aplikaci také použil na notifikace o aktualitách od pořadatelů v průběhu festivalu, které nelze dát do papírového programu.
- Také je podle něj lepší aplikace díky možnosti rychlého vyhledání kapely na internetu atd.
- Aplikaci by použil spíš až na festivalu, ne před ním.
- Plány by dělal předem, na webu festivalu, spíše než pomocí mobilní aplikace, protože je pro něj pohodlnější dělat takové věci na počítači.

- V případě použití webové aplikace by se mu líbilo, kdyby si mohl vytvořit svůj program ve webové aplikaci a potom jej synchronizovat do mobilní aplikace. Webovou aplikaci by použil před festivalem a mobilní v průběhu festivalu.
- V mobilní aplikaci očekává časovou osu a aktuální informace získané pomocí notifikací či novinek.
- Plánování trasy v aplikaci by se mu hodilo, pokud by město, ve kterém je, neznal.
- Zpětnou vazbu festivalu nikdy nedával. Kdyby možnost měl, tak by to udělal pouze v případě, že by mu něco vadilo. Pokud by přišla notifikace a bylo by poskytnutí zpětné vazby jednoduché, pak by to udělal. Nechtěl by otevřené otázky v dotazníku.
- Hodnocení filmů, které by vedlo k vybrání nejlepšího filmu, by dělal. Hodnocení by dělal, až by věděl, který film se mu nejvíce líbil, třeba na konci festivalu.
- Použil by sociální funkce v rámci aplikace jako je komentování novinek či diskuze v rámci festivalu.
- Líbila by se mu možnost ovlivnění festivalu, kdyby mohl třeba vybrat promítání konkrétního filmu.
- Informace o festivalu by hledal hlavně na webových stránkách festivalu, takže pokud by mu někdo neřekl o existenci webové aplikace a možnosti synchronizace s mobilní aplikací, tak by ho to ani nenapadlo hledat.

## B.4 Respondent č.4

### Obecné informace

Studentka chemie, 22 let. Zajímá se o kulturní dění, navštěvuje hudební festivaly a jiné kulturní události.

### Informace získané v průzkumu

- Na festivaly chodí, protože tam chodí její kamarádi. Na hudební si chodí odpočinout, není to jen o hudbě.
- Bere jinak filmové a hudební festivaly. Na filmové chodí jen na pár konkrétních promítání.
- Papírový program použila. Nejvíce jí tam zajímalo, jak jí navazují představení, kam stihne přejít.

## B. UŽIVATELSKÝ PRŮZKUM

---

- Na filmovém už má v plánu, co chce vidět. Hudební řeší spíše až na místě, má pár věcí co chce vidět, což se ale často mění, protože se například rozhodne odpočívat.
- Obecně filmové festivaly plánuje daleko více než hudební festivaly a to často dopředu. Filmový je pro ni více sofistikovaná akce.
- Dopředu zjišťuje, kdo na hudebním festivalu bude, nezjišťuje vedlejší akce.
- Informace o organizaci festivalu nezjišťuje moc dopředu, ale hledala by je v programu.
- Mobilní aplikaci festivalu už použila, primárně ji zajímá program.
- Informace o interpretech v aplikaci moc nezjišťuje, jen pokud by danou kapelu/film neznala.
- Informace pro plánování návštěvy festivalu by hledala hlavně na webu, ale pokud by byla propagována aplikace, tak i v aplikaci.
- Vybrala by si spíše mobilní aplikaci než papírový program, protože v mobilní aplikaci si mohla vytvořit vlastní program.
- Kdyby měla možnost vytvořit si vlastní program dopředu, tak by to udělala. Na filmovém by si určitě program připravila.
- Možnost informování o začátku představení dopředu by se jí líbilo. Také by se jí líbilo, kdyby aplikace měla další realtime funkce.
- Líbilo by se jí plánování trasy z aplikace, což by jí dávalo smysl spíše u filmových festivalů nebo obecně festivalů, které jsou na více místech.
- Zmiňuje, že by se jí v aplikaci hodila mapa kin.
- Je podle ní velmi důležité, jakým způsobem je aplikace a její funkcionality propagována, protože některé funkce by ji v aplikaci hledat nenašlo.
- Kdyby v aplikaci byly novinky s možností sociálních funkcí, tak by je využila.
- Obecně by sociální funkce v aplikaci nejspíše využila.
- Zpětnou vazbu by podle ní účastníci poskytli, ale zejména pokud by z toho měli nějaké výhody, i když je to podle ní o povaze konkrétního účastníka.
- Hodnocení filmů by provedla, pokud by jí přišla notifikace o tom, že to může udělat, nebo nějaká jiná informace o této možnosti.

- Nechápe, na co by měla použít webovou aplikaci, pokud má už mobilní aplikaci. Pokud by měla možnost si dopředu vytvořit ve webové aplikaci svůj program a pak si ho synchronizovat do mobilní, tak by to u filmového festivalu s vysokou pravděpodobností použila.

## B.5 Respondent č.5

### Obecné informace

Studentka architektury, 23 let. Pravidelný a aktivní návštěvník hudebních i filmových festivalů se zájmem o festivalové dění.

### Informace získané v průzkumu

- Na hudební festivaly chodí trávit čas s přáteli, bere je jako dovolenou, chodí na své oblíbené kapely. Na filmové festivaly jezdí, protože má ráda filmy obecně, dívat se na filmy.
- Mezi filmovými festivaly a hudebními festivaly je podle ní velký rozdíl. Hudební festivaly jsou volnější. Filmové festivaly plánuje více dopředu.
- Plánovala by obecně více festivaly, kde by kupovala vstupenky na představení, než festivaly, kde kupuje vstupenku na celý festival.
- Na hudebních festivalech se jí líbí, že může vyplňovat čas kapelami, které nezná, a může tak objevit něco nového zajímavého.
- Použila už papírový program a to jak na hudebním, tak na filmovém festivalu.
- Použila již mobilní aplikaci a pokud ji na festivalu mají, tak nepoužívá papírový program.
- Vyhovují jí informace o tom, co hraje, jednoduché zjištění informací o tom, kdo hraje, a jestli tam má cenu vůbec jít.
- Papírový program používá zejména na program, mapu areálu a informace o kapelách či filmech.
- Mobilní aplikaci by použila, kdyby měla možnost. Vytvoření vlastního programu by se jí hodilo. Použila by to i na filmovém i na hudebním festivalu.
- Informace o festivalu před začátkem zjišťuje a to zejména: kdy to je, kde to je, jak se tam dostane, kde bude bydlet atd.
- Doprovodný program akce ji vždy zajímá, ale zjišťuje si takové věci až na festivalu.

## B. UŽIVATELSKÝ PRŮZKUM

---

- Od aplikace by očekávala program, vlastní program, věci co jsou v papírovém programu a že jí budou chodit notifikace o různých změnách či začátcích jejího programu.
- Filmový festival má podle ní jinou cílovou skupinu a je vhodné podle toho aplikaci upravit.
- Nevadilo by jí registrování do aplikace, které by mělo nějakou přidanou hodnotu.
- Sociální funkce v rámci aplikaci jako je diskuze nebo okomentování novinek by nejspíše využila, ale není to pro ni primární cíl použití aplikace.
- Zpětnou vazbu by poskytla, spíše do diskuze než vyplňování nějakých dotazníků. Myslí si, že obecně lidé neradi vyplňují dotazníky a raději by to dělali za nějakou odměnu.
- Aplikace by podle ní měla pomáhat k tomu, aby si festival užila na sto procent.
- Hodnocení hudebníků jí nedává smysl, je to něco jiného než filmy, více subjektivní. Hodnocení filmů dává smysl třeba pro diváckou soutěž.
- Mobilní aplikaci by si stáhla až když by věděla, že na festival pojede.
- Návštěvu by plánovala dopředu pomocí počítače a webových stránek festivalu. Raději používá počítač, protože má větší obrazovku a lépe si jí v tom vyznává.
- Možnost vytvoření vlastního programu ve webové aplikaci a synchronizace do mobilní aplikace by se jí líbila, protože by to mohla udělat na velké přehledné obrazovce počítače, ale musela by se o této možnosti nějakým způsobem dozvědět od pořadatelů.



---

## Konfigurace Webpacku pro generování webových klientských aplikací

```
1  const path = require('path');
2
3  const webpack = require('webpack');
4  const merge = require('webpack-merge');
5  const Extract = require('extract-text-webpack-plugin');
6  const HtmlWebpackPlugin = require('html-webpack-plugin');
7  const CleanWebpackPlugin = require('clean-webpack-plugin');
8
9  const jsonImporter = require('node-sass-json-importer');
10
11  const appConfig = require('./config/app_config.json');
12  const packages = require('./package.json');
13
14  const enviroment = process.env.NODE_ENV;
15
16  const PATHS = {
17    'src': path.join(__dirname, 'src'),
18    'build': path.join(__dirname, appConfig.folder)
19  };
20
21  const FILENAMES = {
22    entry: './index.js',
23    build: 'js/app.js',
24    vendor: 'js/vendors.js',
25    html: 'index.html',
26    template: 'templates/index.ejs',
27    sass: 'css/styles.css',
28    images: 'images/[hash].[ext]',
29  };
30
31  console.log('Building application with enviroment "${enviroment}"');
32
33  let config = {
34    context: PATHS.src,
35    entry: {
36      app: FILENAMES.entry,
```

## C. KONFIGURACE WEBPACKU PRO GENEROVÁNÍ WEBOVÝCH KLIENTSKÝCH APLIKACÍ

---

```
37     vendor: Object.keys(packages.dependencies),
38   },
39   output: {
40     path: PATHS.build,
41     filename: FILENAMES.build,
42     publicPath: '/'
43   },
44   module: {
45     rules: [
46       {
47         test: /\.js$/,
48         exclude: /node_modules/,
49         use: [
50           {
51             loader: 'babel-loader',
52             options: {
53               presets: ['es2015', 'react', 'stage-0']
54             }
55           }
56         ]
57       },
58       {
59         test: /\.scss$/,
60         loader: Extract.extract({
61           fallback: 'style-loader',
62           use: [
63             {
64               loader: 'css-loader'
65             },
66             {
67               loader: 'sass-loader',
68               options: {
69                 importer: jsonImporter,
70               }
71             }
72           ]
73         })
74       },
75       {
76         test: /\.png$/,
77         use: [
78           {
79             loader: 'url-loader',
80             options: {
81               name: FILENAMES.images,
82               mimetype: 'image/png',
83               limit: 10000
84             }
85           }
86         ]
87       },
88     ]
89   },
90   plugins: [
91     new webpack.EnvironmentPlugin([
92       'NODE_ENV',
93     ]),
94     new webpack.optimize.CommonsChunkPlugin({
95       name: 'vendor',
96       filename: FILENAMES.vendor
97     }),
98     new CleanWebpackPlugin(appConfig.folder),
```

---

```
99     new Extract({
100         filename: FILENAMES.sass,
101     }),
102     new HtmlWebpackPlugin({
103         filename: FILENAMES.html,
104         template: FILENAMES.template,
105         title: AppConfig.name,
106     })
107 ]
108 };
109
110 let envConfig = {};
111
112 if (environment === 'production') {
113     envConfig.plugins = [
114         new webpack.optimize.OccurrenceOrderPlugin(),
115         new webpack.optimize.UglifyJsPlugin({
116             minimize: true,
117         })
118     ];
119     envConfig.devtool = 'source-map';
120 } else {
121     envConfig.devServer = {
122         port: 8080,
123         historyApiFallback: true,
124     };
125     envConfig.devtool = 'eval';
126 }
127
128
129 module.exports = merge(config, envConfig);
```

Zdrojový kód C.1: Konfigurace Webpacku pro generování webových  
klientských aplikací.



---

# Testování použitelnosti

Jedná se pouze o výpisky ze záznamů pořízených při testování použitelnosti. Výpisky z poslední iterace nejsou uvedeny, protože na jejich základě nebyly prováděny žádné změny. Všechny záznamy lze nalézt na přiloženém CD.

## D.1 Testování prototypu mobilní klientské aplikace

### D.1.1 První iterace

#### Uživatel č.1

Muž, marketingový zástupce, 27 let.

- V *seznamu performerů*, pokud má performer pouze jednu session, by jej automaticky přidal do oblíbených bez pop-upu.
- *Můj program* by rozdělil po dnech a poté až po tracích.
- Zvýraznil by na všech obrazovkách lépe informaci o tom, že má uživatel session v *mém programu*.
- Změnil by ikonu pro plánování trasy.
- V *detailu novinky* by zvýřanil, zda je či není novinka hodnocena jako „líbí se mi“.
- V *detailu novinky* by přidal možnost označit komentář jako „líbí se mi“ či vytvořit vlákno.
- V *dotaznících* by po každé vybrané odpovědi nějakým způsobem upozornil na tlačítko pro odeslání.
- V *diskuzi* by přidal možnost označení příspěvku jako „líbí se mi“ a možnost vytvoření vlákna odpovědi.

### Uživatel č.2

Žena, grafická designerka, 28 let.

- V *dnešním programu* jí chybí prokliky na *detail performeru*.
- Mate ji terminologie *můj program*, *můj dnešní program*, *dnešní program* atd.
- Nepochopila zvýraznění *session* v *časové ose*, je potřeba vyřešit zvýraznění *session* z *mého programu*.
- Změnila by terminologii formuláře pro odeslání komentáře v *detailu novinky* a příspěvku v *diskuzi*.
- V *dotaznících* by lépe upozornila na skupinové odesílání.

### Uživatel č.3

Muž, student vysoké školy, 24 let.

- V *dnešním programu* v sekci *můj dnešní program* mu chybí informace o tracku.
- V *mém programu* by upravil rozdělení na dny a poté až tracky.
- Není mu jasné, co znamená zvýraznění *session* v *časové ose*.
- V *detailu novinky* by přidal označení komentářů jako „líbí se mi“ a jejich řazení dle hodnocení a času vytvoření.
- Dotazníky by raději odesílal až poté, co vybere odpovědi, a to hromadně. Vyhovuje mu, že může odeslat jen to, na co chce odpovědět.
- V *diskuzi* by přidal možnost odpovědět na příspěvek a řazení odpovědí do vlákna pod příspěvek pro lepší přehlednost.

### D.1.2 Druhá iterace

#### Uživatel č.4

Muž, programátor, 27 let.

- Přidal by proklik na úvodní obrazovku přes logo v menu.
- Upravil by ikonu pro přidání do *mého programu*.
- Změnil by ikonu pro plánování trasy.
- Odstranil by označení „líbí se mi“ přímo z *novinek*.

- Formulářové pole pro přidání komentáře či příspěvku by se mělo zvětšovat s množstvím textu.
- V *diskuzi* mu přijde špatný poměr velikostí textů ku velikosti obrazovky.
- V *diskuzi* mu chybí vizualizace odpovídání na cizí příspěvek.

#### **Uživatel č.5**

Žena, vysokoškolská studentka, 22 let.

- Přidala by plánování trasy i přímo z pinu na mapě.
- Odstranila by řazení tracků dle abecedy v *seznamu tracků*.
- Plete ji ikona oka pro skrytí všech tracků u vyhledávání v *seznamu tracků*.
- Vadí jí poměr velikosti textů v *diskuzi* vůči velikosti obrazovky.
- Filtr měst v *časové ose* by zobrazila jen pokud je událost ve více městech.
- Přidala by proklik na *detail tracku* přímo z *časové osy*.
- Řadila by tracky v *časové ose* dle počtu session z *mého programu*, které obsahují.
- Přidala by informaci o konci session.

#### **Uživatel č.6**

Žena, projektová manažerka, 27 let.

- Chybí jí uvedení konců session.
- Chybí jí informace o tom, jestli se session přidaná do *mého programu* kryje s jinou session.
- Není jí jasné, co udělá ikona oka v *seznamu tracků*.
- Nerozumí odesílání dotazníků.
- Líbilo by se jí vyhledávání v *diskuzi*.
- Chybí jí vizualizace odpovídání na cizí příspěvek v *diskuzi*.

## D.2 Testování implementace webové klientské aplikace

### D.2.1 První iterace

#### Uživatel č.1

Muž, marketingový zástupce, 27 let.

1. Nenapadlo jej přímo použít vyhledávání, což by udělal pouze v případě velkého množství informací na stránce.
2. Hledal nejprve v *informacích*, poté na *mapě*.
3. Vybírání a přidávání filmů do *mého programu* rozumí a našel bez problémů. Informace o překrytí session z *mého programu* našel rychle v *časové ose*.
4. Snažil se vyhledávat v *seznamu performerů*. Nedíval se, že se seznam automaticky filtruje, odeslal formulář, což vyvolalo přenačtení stránky.
5. Adresu našel přes *časovou osu*, hledal by také v seznamu session v detailu performerera, kde však tato informace chybí.
6. Opět šel přes *časovou osu*, jinak by šel přes seznam tracků, který ale v aplikaci není.

#### Uživatel č.2

Žena, grafická designerka, 28 let.

1. Nepoužila vyhledávání. Stejně jako předchozí testující by ho použila v případě velkého množství informací.
2. Hledala v *mapách* a našla v obrázkových.
3. Má problém s vyhledáváním *severských* filmů. Chtěla by filmy přidávat rovnou do *svého programu* z jejich seznamu.
4. Adresu kina, kde se provádí promítání, našla bez problému.
5. Hledala adresu kina přes detail nějaké session u jména tracku a také v *nejbližším programu*. Přes *časovou osu* by nešla.



### Uživatel č.3

Muž, student vysoké školy, 24 let.

1. Našel bez problému v *informacích*. Vyhledávání by opět použil pouze při vyšším množství dat.
2. Našel bez problému na *mapě*.
3. Bez problému vytvořil svůj program a poté v *časové ose* zjistil, jestli mu program nekoliduje.
4. Při mazání předchozího programu mu vadí, že musí provádět vše přes detail performerera a není možnost přímo z *mého programu*. Filtroval by filmy po kliku na tag v detailu. Omylem odeslal filtr zmáčknutím **enter**, což vyvolalo přenačtení stránky.
5. Šel hledat přes detail performerera, kde není odkaz na pin v *mapách*. Poté šel přes mapy a nakonec přes *můj program*.
6. Hledal nejdříve v *nejbližším programu* a poté přímo přes *mapy*. Přes *časovou osu* by nešel.

### D.2.2 Druhá iterace

#### Uživatel č.4

Muž, programátor, 27 let.

1. Ručně vyhledal přes *informace*. Vyhledávání by zde použil v případě, že by nebylo jasné, kde konkrétně hledat.
2. Hledal nejprve v *informacích* a poté v *mapách*.
3. Zmatený z rychlého přidání performerera do *mého programu* a chybějící možnosti pro přidání v případě, že má performer více session.
4. Zvládl najít přes tag uvedený v detailu konkrétního performerera, na který klikl. Raději by měl tagy uvedené i v listu performerů, aby bylo lépe jasné, přes co vyhledávat.
5. Hledal přes detail performerera. Nevšiml si pinu u položky v *mém programu*.
6. Našel přes *nejbližší program*, chybí mu seznam tracků.

### Uživatel č.5

Žena, vysokoškolská studentka, 22 let.

1. Hledala v *informacích*, nenapadlo ji filtrovat. Při zkoušení filtru ji filtr mate, odesílá pomocí **enter** a nic se neděje. Myslela si, že vyhledávání je v obsahu konkrétní zobrazené informace a ne ve všech informacích.
2. Hledala v *informacích* a poté našla v *mapách*.
3. Neví, proč se u performerera, který má více session, nezobrazí tlačítko pro rychlé přidání do *mého programu*. Myslí si, že je to chyba.
4. Všimla si tagu a klikla na něj. Raději by měla tagy na filtrování přímo u položek v listu performerů, aby bylo jasné, že pomocí nich lze vyhledávat.
5. Nenapadlo ji kliknout na velkou ikonu pinu v *mém programu*. Nevšimla si jí.
6. Hledala přes mapu, pak přes *nejbližší program* na *homepage*. Chybí jí stránka se seznamem tracků.

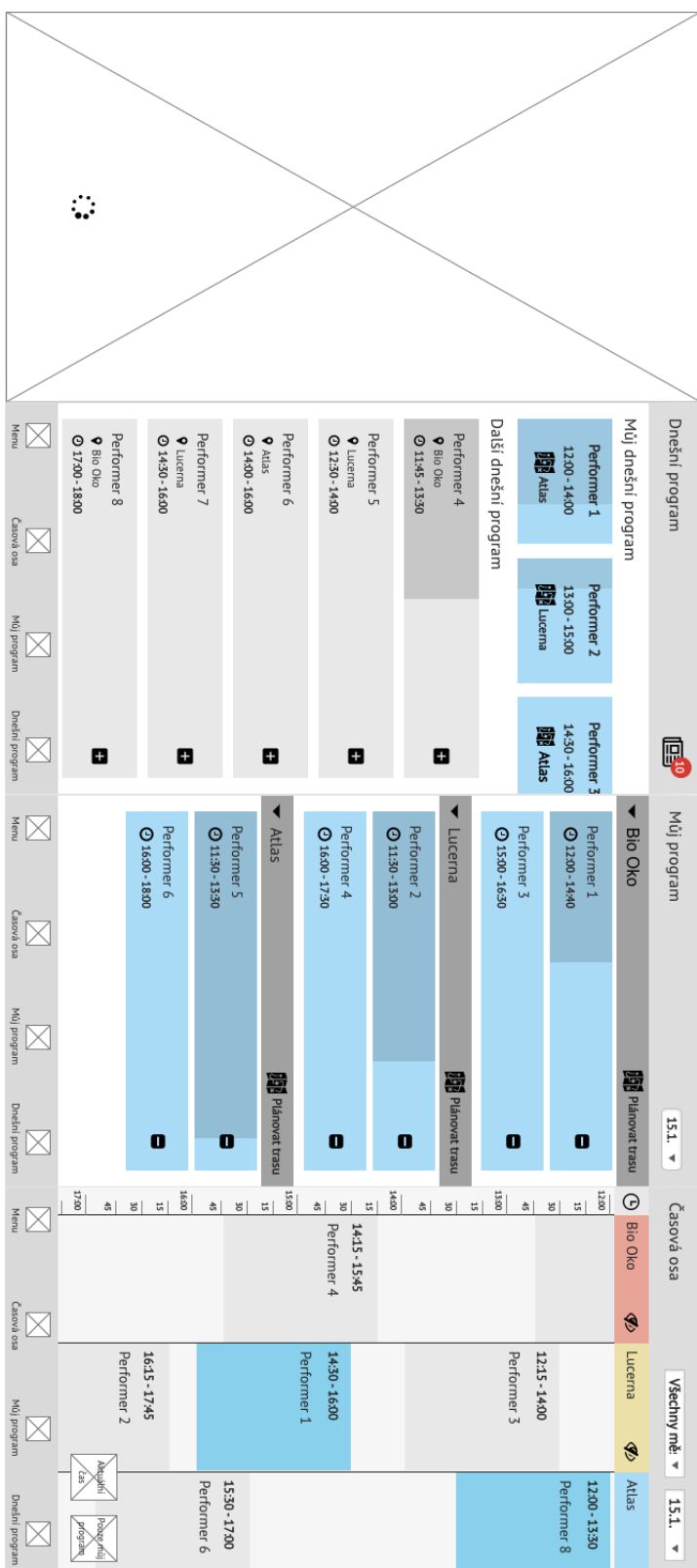
### Uživatel č.6

Žena, projektová manažerka, 28 let.

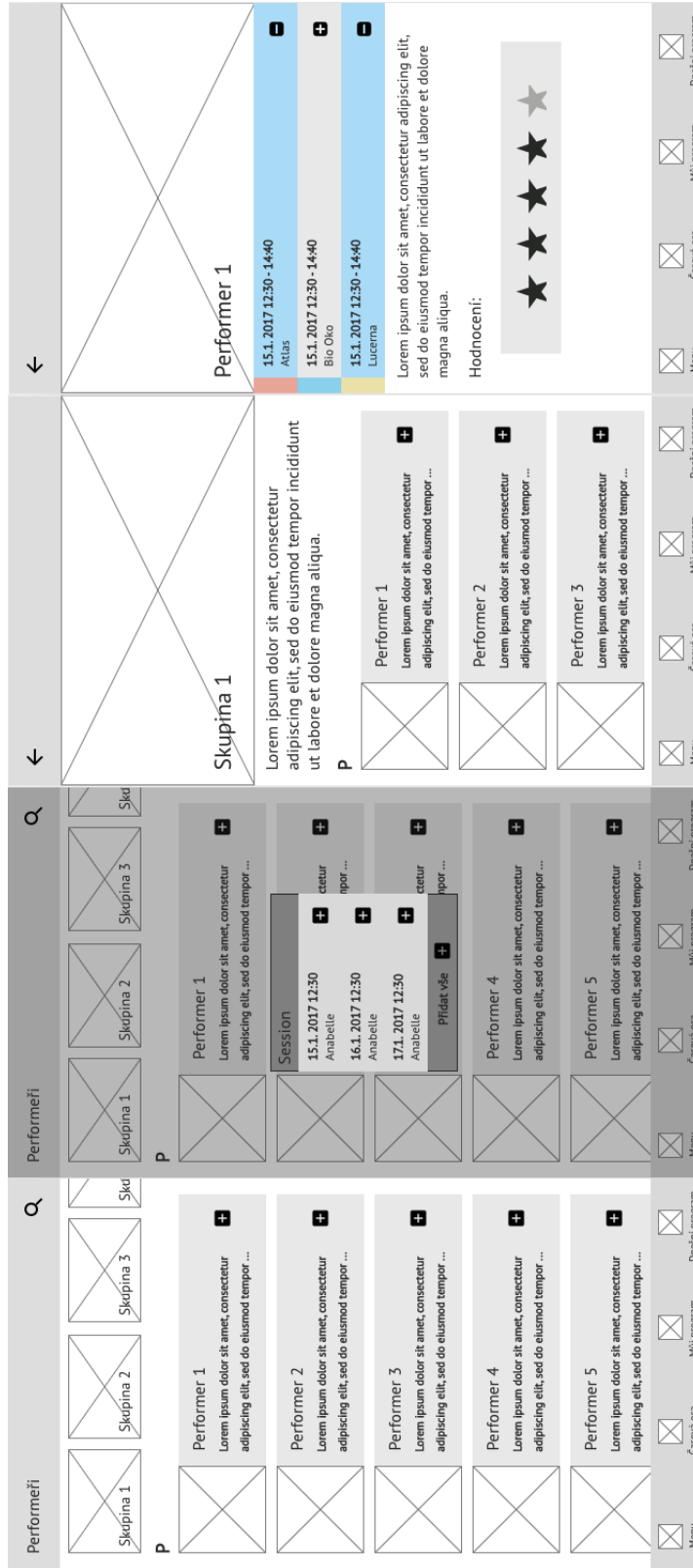
1. Použila hledání, nedívala se, že se filtruje přímo při psaní, zmáčkla **enter** a zmátlo ji, že se nic nestalo.
2. Bez problému našla v mapě areálu, ale uvažovala i o hledání v *informacích*.
3. Rychlé přidání přes *seznam performerů* ji zmátlo, protože bylo vynecháno v případě, že měl performer více session.
4. Nevěděla, že by mohla filtrovat podle tagů u performerů.
5. Našla bez problémů přes detail performerera.
6. Našla rychle přes *nejbližší program*.

# **Finální verze uživatelských rozhraní**

## E. FINÁLNÍ VERZE UŽIVATELSKÝCH ROZHRAŇÍ

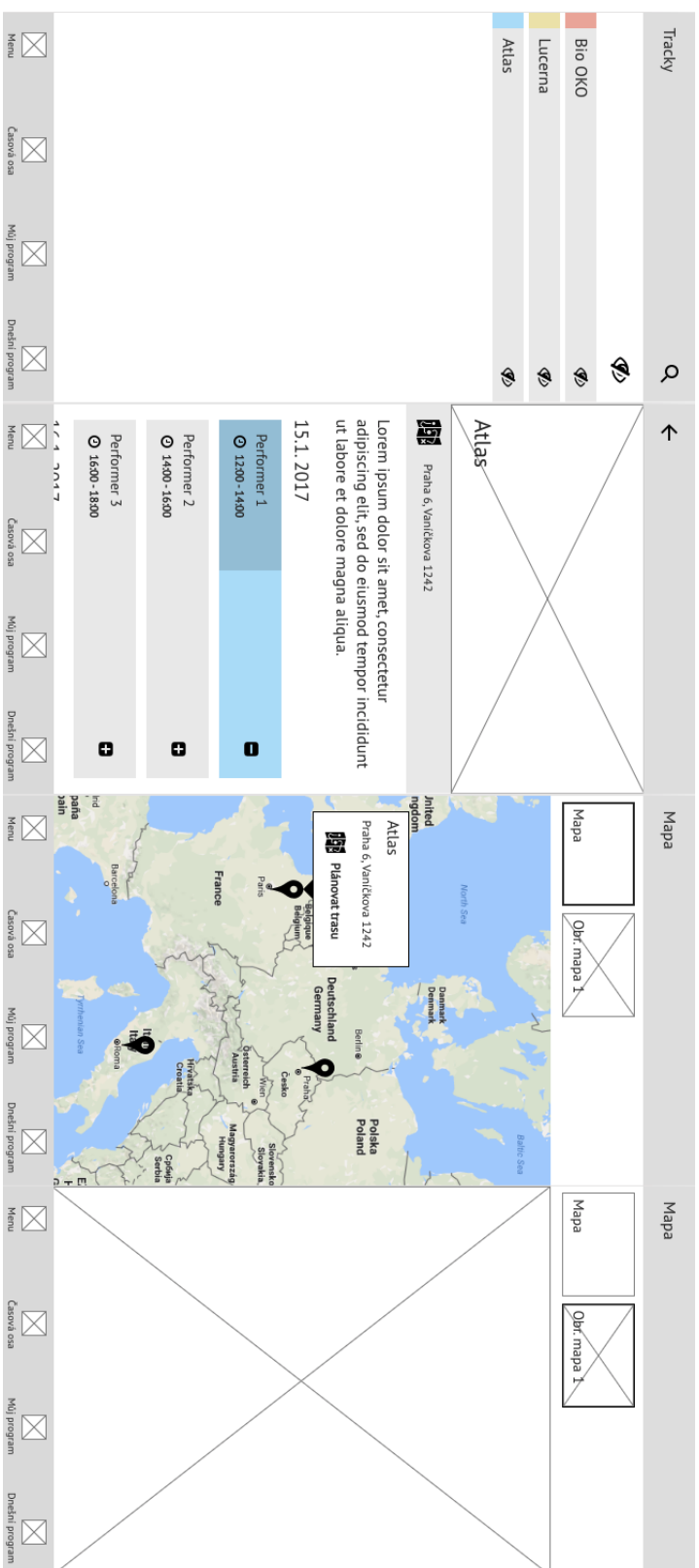


Obrázek E.1: Mobilní aplikace: loading screen, dnešní program, můj program, časová osa

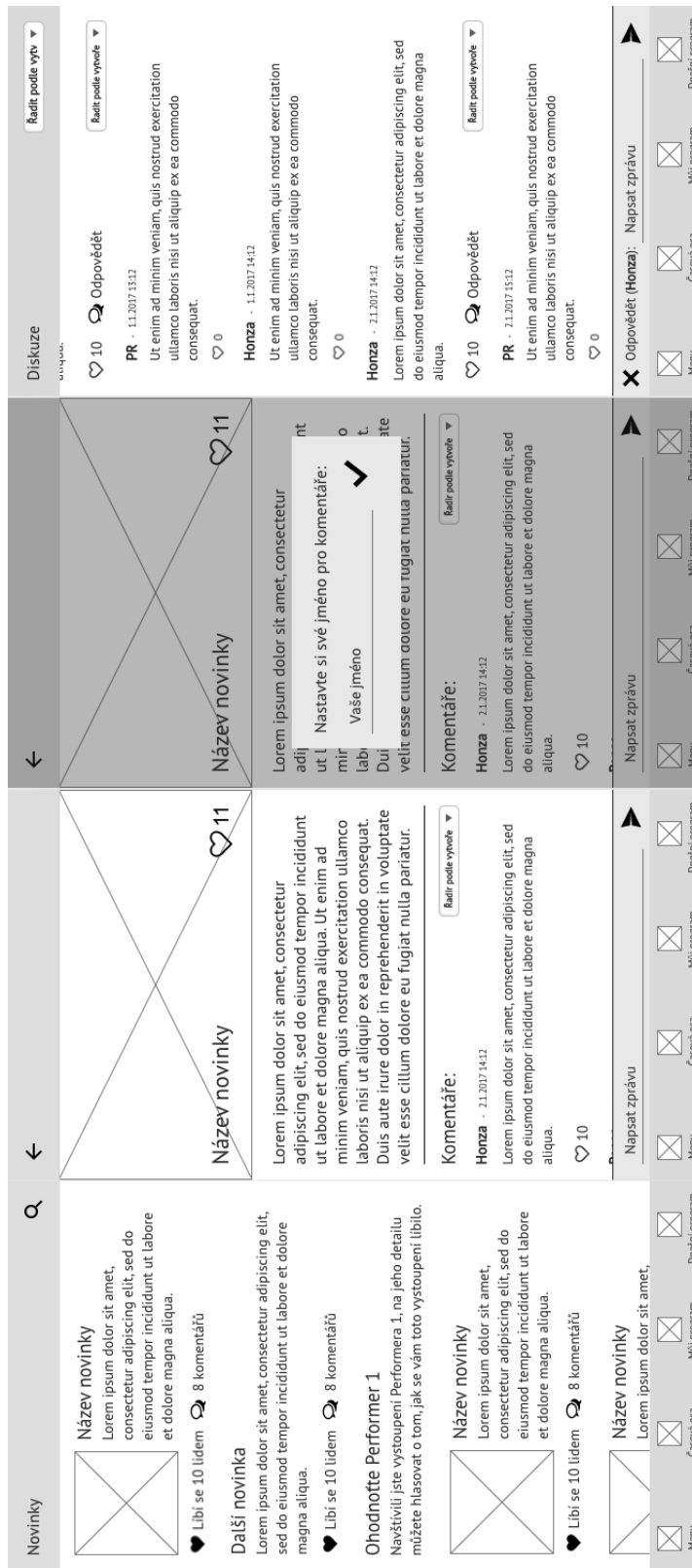


Obrázek E.2: Mobilní aplikace: seznam performerů, list performerů pop-up, skupina performerů, detail performerů

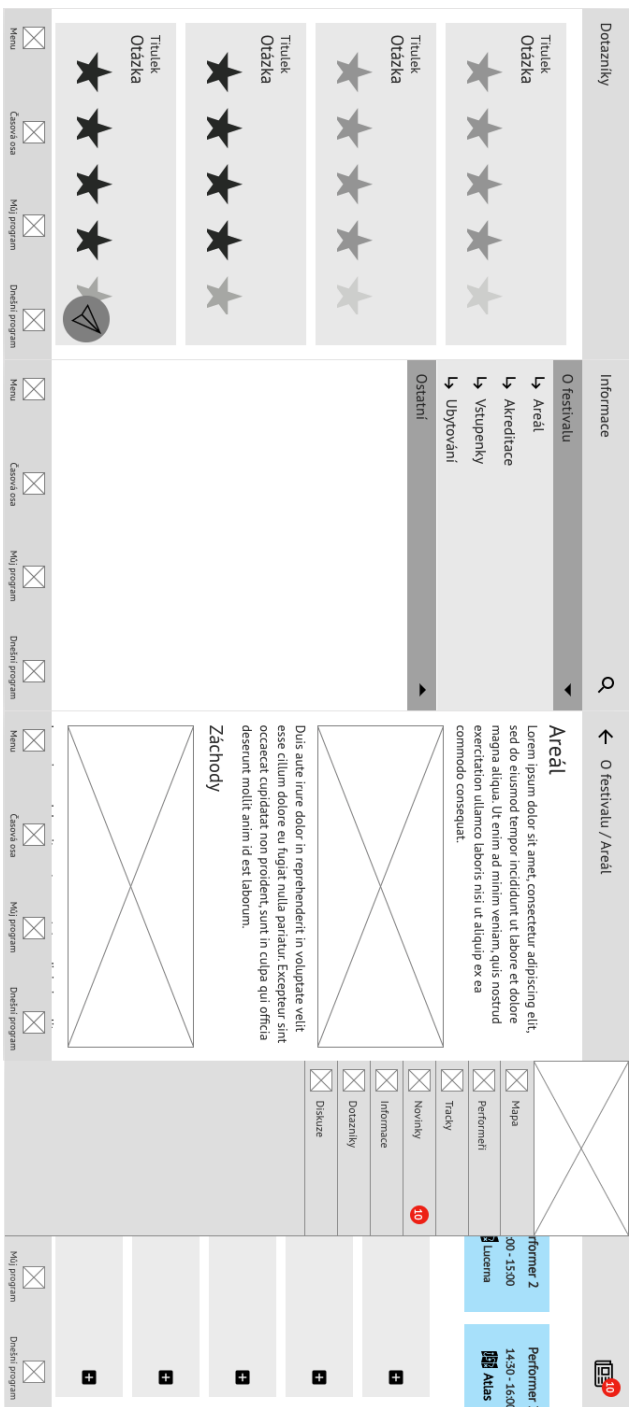
## E. FINÁLNÍ VERZE UŽIVATELSKÝCH ROZHRAŇÍ



Obrázek E.3: Mobilní aplikace: seznam tracků, detail tracku, interaktivní mapa, obrázková mapa



Obrázek E.4: Mobilní aplikace: seznam novinek, detail novinky, detail novinky nastavení jména, diskuze



Obrázek E.5: Mobilní aplikace: dotazníky, informace, detail informace



# Testing festival

Domů Časová osa Filmy Mapy Informace

## Můj program

**Trainspotting**  
↳ Bio OKO  
🕒 03.09.2017 20:00

**T2 Trainspotting**  
↳ Lucerna  
🕒 03.09.2017 22:00

## Program pro nejbližších 24 hodin

**Bio OKO**  
Boj sněžného pluhu s mafií  
🕒 03.09.2017 14:00

**Lucerna**  
Povídky ze Stockholmu  
🕒 03.09.2017 16:00

**Atlas**  
Sámská krev  
🕒 03.09.2017 18:00

## Novinky

**Další filmy**  
Přidali jsem další filmy a jejich promítání.

**První filmy**  
Přidali jsme první filmy k festivalu.

**Festival začíná**  
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure

Obrázek E.6: Webová aplikace: domovská stránka

## E. FINÁLNÍ VERZE UŽIVATELSKÝCH ROZHRANÍ

	Bio OKO	Lucerna	Atlas
14:00	Boj snežného...		
15:00			
16:00		16:00 Povídky ze St...	
17:00			
18:00			18:00 Sámská krev
19:00			
20:00	20:00 Trainspotting		
21:00			
22:00		22:00 T2 Trainspoti...	
23:00			

Obrázek E.7: Webová aplikace: časová osa

## Testing festival

Domů Časová osa Filmy Mapy Informace

### Anthropoid

**drama**

Snímek Anthropoid vychází z výjimečného skutečného příběhu vojenské operace za druhé světové války, jejímž cílem byl atentát na obergruppenführera SS Reinharda Heydricha. Heydrich byl po

### Boj sněžného pluhu s mafií

**krimi** **seversky**

Nils se se svým sněžným pluhem prohání zasněženými cestami v okolí chladného norského městečka. Proráží cesty pro své spoluobčany, což považuje za velmi zodpovědnou práci, kterou

### Povídky ze Stockholmu


**seversky**

Životní osudy pěti lidí se zkříží během několika deštivých dní ve stockholmské metropoli. Johan se ze všech sil snaží o vydání svého rukopisu. Thomas dělá postoka na Ministerstvu financí. Asociální

### Sámská krev

**seversky**

Čtrnáctiletá Elle Marja je Sámská dívka, která se věnuje chovu sobů. Vystavena rasismu éry třicátých let minulého století a ponižujícím vyšetřením na své internátní škole, začne snít o novém



### Povídky ze Stockholmu

**Promítání**

**Lucerna**

03.09.2017 16:00

**Lucerna**

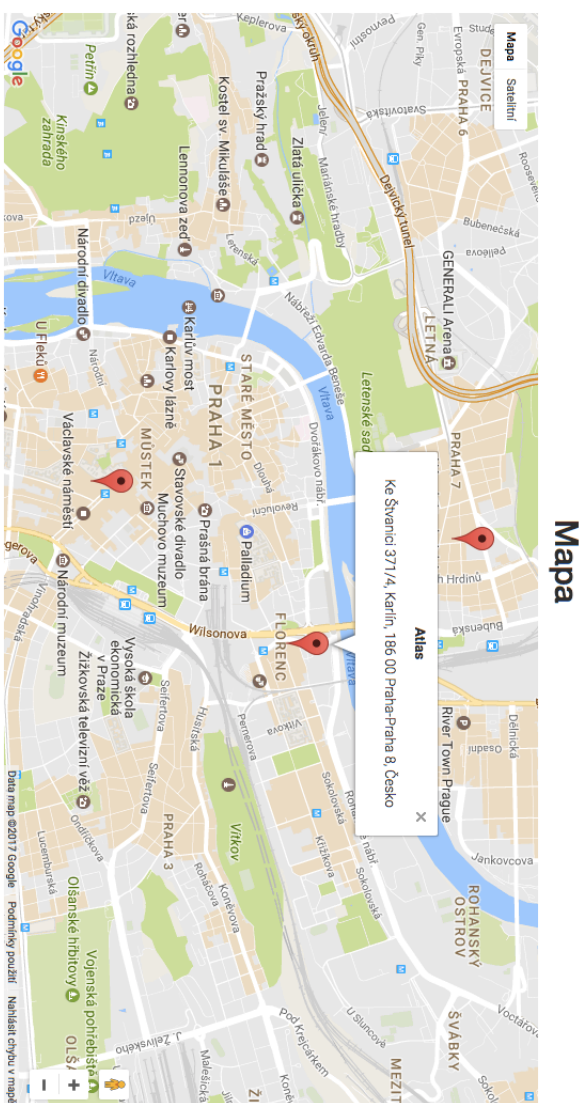
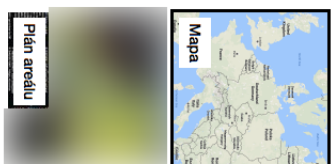
04.09.2017 19:00

**Délka** 97 min  
**Rok** 2013

Obrázek E.8: Webová aplikace: seznam performerů

## Testing festival

Domů Časová osa Filmy Mapy Informace



Obrázek E.9: Webová aplikace: mapy

## Testing festival

The image shows a horizontal navigation menu for a website titled "Testing festival". The menu items are: "Domů", "Časová osa", "Filmy", "Mapy", and "Informace". The "Informace" item is highlighted with a dark background. Below the menu is a search bar with the placeholder text "Vyhledávat v obsahu informací...". To the right of the search bar, there are several menu items: "O festivalu", "Vstupenky", "Applikace", "Festival", "Další", and "Sponzoři". The "Applikace" item is highlighted with a dark background. Below the "Applikace" item, there is a sub-menu with the following items: "O festivalu / Aplikace", "Aplikace", and "Festival má skvělou mobilní a webovou aplikaci.".

Domů Časová osa Filmy Mapy **Informace**

Vyhledávat v obsahu informací...

**O festivalu**  
Vstupenky  
**Applikace**  
Festival  
**Další**  
Sponzoři

**O festivalu / Aplikace**  
**Aplikace**  
Festival má skvělou mobilní a webovou aplikaci.

Obrázek E.10: Webová aplikace: informace



---

## Obsah přiloženého média

README.md	popis obsahu média a instalační příručka
prototypes	prototypy uživatelských rozhraní webové a mobilní aplikace
_ mockups	mockupy vyexportované do HTML
_ wireframes	naskenované papírové wireframy
recordings	záznamy testování použitelnosti
src	zdrojové kódy
_ app	zdrojové kódy webové aplikace
_ build	vygenerovaná webová aplikace pro testovací událost
_ thesis	zdrojová forma práce ve formátu $\text{L}^{\text{T}}\text{E}^{\text{X}}$
screens	záznamy původních návrhů aplikací ekosystému
text	
_ thesis.pdf	text práce ve formátu PDF
_ zadani.pdf	zadání práce ve formátu PDF