



**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

<b>Název:</b>	Linky - webový portál
<b>Student:</b>	Tomáš Vošický
<b>Vedoucí:</b>	Ing. Jiří Chludil
<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 2018/19

### **Pokyny pro vypracování**

Projekt Linky je interaktivní světelná instalace na budově ČVUT FEL.

1. Analyzujte problém (aktuální situaci) správy fasády Linek.
2. Pomocí metod softwarového inženýrství navrhnete nový webový portál pro veřejnost podporující správu uživatelů, akcí, aplikací, atd.
3. Navrhnete uživatelsky přívětivé a responsivní administrační rozhraní pro správu portálu.
4. Navržené řešení implementujte. Při implementaci použijte technologii NodeJS. Pro nasazení použijte Continuous Integration a verzovací systém GIT.
5. Prototyp portálu podrobte vhodným testům.

### **Seznam odborné literatury**

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.  
děkan

V Praze dne 14. prosince 2017





**FAKULTA  
INFORMAČNÍCH  
TECHNOLÓGIÍ  
ČVUT V PRAZE**

Bakalářská práce

## **Linky – webový portál**

*Tomáš Vošický*

Katedra softwarového inženýrství

Vedoucí práce: Ing. Jiří Chludil

13. května 2019



---

## Poděkování

Děkuji své rodině a přátelům za podporu. Především bych chtěl poděkovat vedoucímu práce Ing. Chludilovi za velkou pomoc, podporu a poskytnutí cloudového úložiště.



---

# 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 mojí 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. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 13. května 2019

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2019 Tomáš Vošický. 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 a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Vošický, Tomáš. *Linky – webový portál*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.



---

# Abstrakt

Bakalářská práce se zabývá webovým portálem pro správu a propagaci interaktivní světelné instalace Linky umístěné na budově Fakulty elektrotechnické Českého vysokého učení technického. Správcům webového portálu je umožněno měnit obsah, vytvářet akce, spravovat registrované uživatele a programy a řídit běh instalace. Vývojáři mohou nahrávat své vytvořené programy. Návštěvníkům je zpřístupněna prezentační část portálu.

Práce popisuje proces vývoje webového portálu od analýzy a návrhu přes implementaci až po testování. V analýze jsou zmapována existující řešení a použité technologie. Za pomoci prostředků softwarového inženýrství byl navržen webový portál a následně byl implementován jeho prototyp. Veškerý kód byl při implementaci verzován systémem Git a nasazován pomocí průběžné integrace a průběžného nasazení.

**Klíčová slova** projekt Linky, návrh a analýza, webový portál, interaktivita, světelná instalace, budova FEL ČVUT, TypeScript

---

# Abstract

Bachelor thesis is focused on the web portal for propagation and management of Linky interactive light installation, which is located on the facade of Faculty of Electrical Engineering at Czech Technical University. Administrators have access to modify the content, create events, manage registered users and programs and control scheduled programs. Developers are able to upload their programs. Visitors can only show content.

The thesis describes process of analysis, design, implementation and testing of web portal. In analysis part, existing solutions and used technologies were mapped. After design, prototype was implemented. Continuous integration and continuous delivery were used while implementing. Source code was developed under the Git version control system.

**Keywords** Linky project, design and analysis, web portal, interactivity, light installation, FEE CTU building, TypeScript

---

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Cíl práce</b>	<b>3</b>
<b>2 Analýza</b>	<b>5</b>
2.1 Popis projektu . . . . .	5
2.2 Existující řešení . . . . .	6
2.3 Funkční a nefunkční požadavky . . . . .	8
2.4 Vybrané technologie . . . . .	10
2.5 Použité aplikace a nástroje . . . . .	11
2.6 Ochrana osobních údajů . . . . .	12
<b>3 Návrh</b>	<b>13</b>
3.1 Koncept Linky . . . . .	13
3.2 Případy užití . . . . .	15
3.3 Diagram tříd . . . . .	17
3.4 Aplikační rozhraní . . . . .	17
3.5 Uživatelské rozhraní . . . . .	19
3.6 Ochrana osobních údajů . . . . .	20
3.7 Bezpečnost . . . . .	20
<b>4 Implementace</b>	<b>23</b>
4.1 Použité technologie . . . . .	23
4.2 Kód . . . . .	24
4.3 Serverová část . . . . .	25
4.4 Klientská a serverová část . . . . .	26
4.5 Continuous Integration a verzování . . . . .	28
4.6 Nastavení vývojového prostředí . . . . .	30
<b>5 Testování</b>	<b>33</b>

5.1	Automatické testy . . . . .	33
5.2	Nástroje pro automatické testování . . . . .	34
5.3	Testování responzivity . . . . .	34
5.4	Testování použitelnosti . . . . .	36
<b>6</b>	<b>Budoucí práce</b>	<b>39</b>
6.1	Funkce . . . . .	39
6.2	Virtualizace . . . . .	39
	<b>Závěr</b>	<b>41</b>
	<b>Literatura</b>	<b>43</b>
	<b>A Seznam použitých zkratk</b>	<b>47</b>
	<b>B Obsah příložené SD karty</b>	<b>49</b>
	<b>C Wireframy</b>	<b>51</b>

---

## Seznam obrázků

3.1	Koncept Linky . . . . .	14
3.2	Diagram případů užití programu . . . . .	15
3.4	Ukázkový wireframe detailu programu . . . . .	20
3.3	Diagram tříd . . . . .	21
4.1	Adresářová struktura projektu . . . . .	25
5.1	Porovnání responzivního zobrazení na různých velikostech obrazovky	35
C.1	Wireframe seznamu programů . . . . .	51
C.2	Wireframe detailu programu . . . . .	52
C.3	Wireframe přihlašovací stránky . . . . .	52
C.4	Wireframe plánu akcí . . . . .	53
C.5	Wireframe seznamu stránek . . . . .	53
C.6	Wireframe formuláře pro úpravu statické stránky . . . . .	54
C.7	Wireframe uživatelského profilu . . . . .	54
C.8	Wireframe formuláře pro nahrání programu . . . . .	55
C.9	Wireframe formuláře pro nastavení instalace . . . . .	55



---

## Seznam tabulek

3.1	Přístupové body API stránek . . . . .	18
3.2	Přístupové body API akcí . . . . .	18
3.3	Přístupové body API souborů . . . . .	18
4.1	Proměnné v souboru <code>.env</code> . . . . .	31





---

# Úvod

Na fasádu budovy Fakulty elektrotechnické byla nainstalována sada světelných LED segmentů. Instalace však pouze přehrává předpřipravené animace a proto je načase tento nedostatek změnit a vyzvat vývojáře, aby vytvořili pro instalaci interaktivní programy.

Náhodný kolemjdoucí se může zapojit do fungování instalace a přes mobilní telefon nebo počítač hrát hry nebo do fungování zasahovat jiným způsobem. Vytvořený prototyp webového portálu umožňuje instalaci nastavovat, přidávat a schvalovat programy a informovat širokou veřejnost o aktualitách.

Téma práce bylo zvoleno pro zvýšení povědomí o práci autorů Linek. Autor se tímto zúčastní týmové práce na projektu univerzitního významu.

V práci se autor zabývá analýzou problému správy Linek, návrhem a implementací prototypu webového portálu a nakonec průběhem testování.

Tato práce je součástí trojice bakalářských prací a navazuje na závěrečnou práci kolegy, která se zabývá analýzou, návrhem a implementací jádra světelné instalace řídicí samotný běh. Na jádro Linek navazuje závěrečná práce druhého kolegy, zabývající se grafickou knihovnou pro vykreslování scény na světelnou instalaci.



---

## Cíl práce

Cílem rešeršní části práce je získat povědomí o problému správy a přínosu projektu Linek okolí. V rešeršní části je popsán samotný projekt Linky a byla analyzována existující řešení ostatních světelných instalací a také webového portálu Linek. Byly zjištěny funkční a nefunkční požadavky a zmapovány aktuální trendy tvorby webového portálu pomocí technologie Node.js.

Cíl praktické části je pomocí metod softwarového inženýrství navrhnout webový portál a implementovat prototyp, který podpoří propagaci a správu světelné instalace. Webový portál bude fungovat jako jedna. Při vývoji se využije průběžné integrace (*anglicky Continuous Integration*) a verzovacího nástroje Git. Vhodné části budou otestovány.



---

# Analýza

Analýza je základem při vývoji kvalitního softwarového projektu. Cílem analýzy je pochopit daný problém, zmapovat již existující řešení a zhodnotit, jaké technologie a nástroje se hodí k řešení daného problému. Před samotnou analýzou webového portálu je ale třeba se seznámit se s projektem Linky, pochopit jeho cíle a myšlenky.

## 2.1 Popis projektu

Tato sekce seznámí čtenáře s projektem Linky. Sekce popisuje, kde se projekt nachází, jak vypadá a kdo za ním stojí. Další část se věnuje krátké historii a na závěr je shrnuto, jaký význam má projekt pro okolí.

### 2.1.1 Popis světelné instalace

Linky je světelná instalace umístěná na jihovýchodní straně budovy Fakulty elektrotechnické Českého vysokého učení technického a je viditelná z velké části Vítězného náměstí v Praze. Do prostorů mezi okny na budově je svisle připevněno pět sloupců, kde v každém sloupci je postupně vyskládáno 17 panelů po 12 barevných LED svítidel. Z toho dva panely v každém sloupci vystupují nad střechu. Jedná se tedy o displej velikosti  $5 \times 204$  pixelů.

### 2.1.2 Historie

Provoz Linek byl zahájen 25. února 2016 na základě konceptu tvůrčího týmu ve složení: Marián Karel a Josef Šafařík, Roman Berka, Jakub Hybler a Hana Chmelenská [1]. Během let 2016–2019 se konalo několik jednorázových akcí. Představení Linek započal program Creative Colours of FEL s interaktivním programem. Linky mohli zájemci ovládat přes mobilní aplikaci. V roce 2019 byla na Linkách spuštěna vizualizace při příležitosti uctění památky Jana Palacha. Na státní svátky zdobí světelnou instalaci vlajka České republiky.

### 2.1.3 Sociální dopad

#### Atmosféra a využití

Ing. Jakub Hybler, člen tvůrčího týmu, ve svém článku, kromě technických detailů také popisuje jaké je autory zamýšlené využití instalace a jak Linky mohou informovat nebo zabavit kolemjdoucí:

*„Realizovaný koncept poskytuje dostatečný stupeň variability tak, aby mohla vznikat škála barevných kreací reagujících na nejrůznější podněty. Těmito podněty mohou být údaje o hluku a znečištění vzduchu vně budovy, množství dat proudících do budovy a ven nebo interaktivní vstupy kolemjdoucích z chytrých mobilních telefonů. Světelný systém může na tyto podněty reagovat změnami obrazu v závislosti na právě aktivní aplikaci (např. během dne reaguje na vnější podněty, večer se stane součástí nějakého představení a v noci ustoupí do pozadí snížením intenzity světla nebo úplně zhasne, aby nerušil noční klid). Budova, která bývá obcházena bez povšimnutí, tak získává schopnost, která není obvyklá – reaguje.“ [1]*

#### Zájem vývojářů a veřejnosti

Pro Linky je vytvořen webový portál, který má instalaci prezentovat veřejnosti a přivést potencionální vývojáře z řad středních, ale i vysokých škol k vývoji interaktivních programů a graficky zajímavých vizualizací. Vytvářené projekty mohou také například vznikat jako semestrální práce softwarových projektů.

#### Akce

V Praze se během roku konají různé akce a festivaly. Instalace Linky má potenciál se do těchto projektů zapojit. Oživeny mohou být například farmářské trhy, které se konají přímo před budovou každý týden. Návštěvníky festivalů jako Signal Festival nebo Open House může světelná instalace zaujmout doplňujícími animacemi.

## 2.2 Existující řešení

Při průzkumu existujících řešení se autor rozhodl své výsledky rozdělit do dvou kategorií. První kategorie se zaměřuje na jiné interaktivní instalace ve světě i v České republice. Druhá kategorie shrnuje realizovaná řešení a pokusy přímo pro projekt Linky.

### 2.2.1 Portál pro interaktivní instalaci

Na světě existuje celá řada světelných instalací na veřejných místech jako jsou například budovy, ale málo z nich poskytuje interakci s veřejností. Autor se zaměřil pouze na ty, které lze nebo šlo ovládat prostřednictvím internetu.

## luminous

Na jednu z budov v části Darling Harbour australského města Sydney byla v roce 2011 nasazena největší permanentní světelná instalace na světě „luminous“. Pro tuto instalaci byla vytvořena webová stránka se simulátorem. Pokud návštěvník otevře webovou stránku na svém mobilním telefonu, získá tak přístup k ovládání předpřipravených her. Webová stránka dále umožňuje nakreslit jakýkoliv obrazec a zobrazit ho v určenou dobu. [2]

Návštěvníci mohou použít pouze předpřipravené programy a narozdíl od Linek není možné vytvářet nové. Vytvořené programy mohou posloužit jako inspirace vývojářům programů pro Linky. Zajímavostí je, že interaktivní obsah je zpřístupněn pouze večer ve vybrané dny, kdy má světelná instalace nejlepší viditelnost.

## Vánoční stromek

Na katedře řídicí techniky FEL ČVUT ve spolupráci se Slow TV a Dopravním podnikem hlavního města Prahy vznikl na vánoce roku 2015 svítící vánoční strom, který byl umístěn na udržbovou mazací tramvaj. Osvětlení vánočního stromu mohl kdokoli vzdáleně ovládat přes webovou stránku <http://stromek.felk.cvut.cz>. Kliknutím na jakoukoliv část vánočního stromku mohl návštěvník webu rozsvítit příslušné místo osvětlení. Výsledek bylo možné živě sledovat na streamu natáčeného kamerou umístěnou na mazací tramvaji, přitom šlo zjistit aktuální polohu tramvaje na mapě. [3]

Na tomto projektu je zajímavé, že se ho mohli zúčastnit i uživatelé, kteří samotnou instalaci nikdy naživo neviděli. Akce vybízející k činnosti uživatelů je na webu viditelná hned na první pohled.

### 2.2.2 Webové prezentace pro Linky

Pro Linky již byla v minulosti vytvořena statická webová prezentace s neveřejnou dokumentací API, ale obojí není již dostupné. První pokusy o interaktivitu byly zaznamenány v prosinci 2016, kdy byly představeny celkem tři aplikace. Jako první byl neinteraktivní přehrávač audiovizuální kompozice Josefa Šafaříka a Adama Sporky. Druhou aplikací byla hra PONG inspirovaná tenisovými počítačovými videohrami. Třetí aplikace VOTE vybízela přítomné diváky k odpovídání různých otázek. Hry mohli přítomní ovládat pomocí mobilní a webové aplikace. Tyto hry nejsou již v provozu a mobilní aplikace také není dostupná. [4]

Na začátku roku 2019 byly Centrem znalostního managementu spuštěny dvě webové aplikace [5], kterými bylo uvedeno několik ukázkových interaktivních programů a simulátor pro vývojáře. Dokumentace umístěná na prezentačním webu obsahovala popis jednoduchého API pro komunikaci programů a fasády. V době psaní práce ze zmíněného API fungoval pouze jeden přístupový bod.

### 2.3 Funkční a nefunkční požadavky

Na základě zadání byly sestaveny požadavky. Funkční požadavky definují základní funkčnosti, které by měl portál umět. Nefunkční požadavky kladou na systém důležité vlastnosti a omezení týkající se kvality a prostředí.

#### 2.3.1 Funkční požadavky

##### Správa

###### FR1 Správa obsahu

Správci webového portálu budou schopni měnit obsah jednotlivých stránek bez nutnosti zásahu programátora či stahování nebo instalace jakýchkoliv vývojářských nástrojů. Do stránek bude možné nahrávat obrázky a vkládat videa.

###### FR2 Správa programů

Vývojářům bude umožněno nahrávat své programy na webový portál. Tyto programy budou moci správci schválit, upravit detaily a mazat je.

###### FR3 Správa uživatelů

Správčům webového portálu bude umožněno spravovat uživatele s přidělováním rolí.

###### FR4 Správa prostředí

Portál správčům umožní práci s prostředím pro plány běhů.

###### FR5 Správa akcí

Do portálu bude možné přidávat a upravovat plánované akce.

###### FR6 Ovládání instalace

Instalaci půjde měnit různá nastavení. Z provozních důvodů mohou správci potřebovat upravit celkový jas, rozlišení obrazu nebo instalaci ručně vypnout.

##### Přihlášení

###### FR7 Přihlášení uživatele

Uživatelé se budou moci do portálu přihlašovat. Přihlašování bude probíhat pomocí přihlašovacích bran Facebook Login a Google Sign-In. Přihlášení do webového portálu by mělo sloužit převážně vývojářům programů a správčům Linek. Přihlásit se bude moci i běžný uživatel.

###### FR8 Odhlášení uživatele

Uživatelé se budou moci odhlásit.



### **FR9 Profil**

Každý uživatel bude mít dostupný svůj profil, kde bude mít možnost zobrazit a upravit uložené údaje.

### **Prezentace**

#### **FR10 Zobrazení statické stránky**

Portál bude zobrazovat veřejnosti uložené statické stránky.

#### **FR11 Zobrazení programu**

Na portálu bude vyvěšen seznam všech spustitelných programů.

#### **FR12 Zobrazení plánu**

Uživatelé uvidí na portále naplánované programy a vypsané akce.

#### **FR13 Simulátor**

Portál umožní zobrazit simulátor k zobrazení aktuální scény Linek.

### **2.3.2 Nefunkční požadavky**

#### **NR1 Bezpečnost**

Budou přijata opatření na zajištění bezpečnosti portálu proti útokům.

#### **NR2 Kód**

Pro dlouhodobé fungování a udržování portálu je důležité, aby byl kód přehledný a strukturovaný. Při vývoji bude použit verzovací systém Git.

#### **NR3 Responzivita a přístupnost**

Webový portál bude optimalizovaný pro zobrazení z jakéhokoliv zařízení bez ohledu na velikost a typ. V případě webového portálu Linky je tento požadavek obzvláště důležitý, jelikož cílovou skupinou prezentační části portálu jsou právě náhodní kolemjdoucí v okolí Linek.

#### **NR4 Uživatelská přívětivost**

Ovládání portálu musí být jednoduché a průměrný návštěvník by měl být schopen základní navigace. Správcům nesmí systém klást větší překážky.

#### **NR5 Ochrana osobních údajů**

Jelikož portál bude nakládat s osobními údaji pro identifikaci uživatelů, musí být při ukládání těchto dat dodržen platný zákon o ochraně osobních údajů.

### 2.4 Vybrané technologie

V analytické části se autor práce zamyslel nad použitými technologiemi, které se podle něj hodí pro vývoj webového portálu popisovaného v této práci. K jednotlivým technologiím a jazykům jsou sepsány i výhody oproti konkurenci.

#### TypeScript

Jazyk TypeScript je rozšířením standardu ECMAScript uvedený v roce 2012 společností Microsoft. Hlavní funkcí tohoto jazyka je statická kontrola typů. Díky této kontrole je odhalena spousta programátorských chyb již v době psaní. Typová kontrola umožňuje zdrojový kód jednoduše refaktorovat a dokumentovat. Jedním z cílů autorů je udržovat funkce jazyka v souladu se standardem ECMAScript. Kód TypeScriptu lze navíc přeložit do jazyka JavaScript v čisté a čitelné podobě s možností výběru verze ECMAScriptu. Pokud vývojář použije do svého projektu knihovnu, která není v TypeScriptu napsána nebo nemá definiční soubor, může si pro svůj projekt typové definice sám předeepsat.

#### Výhody

- Použití typů není povinné.
- Podpora ve velkém množství vývojových prostředí.
- Díky přehledu o typech je možné refaktorovat kód v rámci projektu.
- TypeScript má téměř stejnou syntaxi jako JavaScript oproti jazykům Dart nebo CoffeeScript. Díky tomu není třeba se učit zcela novému jazyku.
- Přejít na TypeScript v existujícím projektu by měl být bezproblémový, jelikož i čistý JavaScript je platný TypeScript.

#### MongoDB

MongoDB je nerelační dokumentově orientovaná databáze. Narozdíl od SQL-like databáze, kde se informace ukládají jako řádky do tabulek, do MongoDB se informace ukládají jako dokumenty do kolekcí. Díky přizpůsobivému modelu není potřeba předem deklarovat strukturu dokumentů a v dokumentu je možné přidat novou informaci i bez zásahu do jiného dokumentu v kolekci. [6]

MongoDB byla zvolena pro tento projekt, protože umožňuje rychlé nasazení a nevyžaduje žádnou práci s konverzí reprezentace uložených dat.

### Framework

Psaní webové aplikace bez použití frameworku je časově náročné a navíc nebezpečné. Webový framework za programátora řeší spoustu záležitostí důležitých pro vývoj a provoz webu. Podle zadání má být využita technologie Node.js, proto se krátká analýza frameworků zaměří pouze na ty pracující s touto technologií.

Nejznámějšími webovými frameworky jsou v současné chvíli AngularJS, Next.js a Nuxt.js. Autor se rozhodl využít framework Next.js jelikož se zdá být pro vývoj webové aplikace vhodnou volbou díky výhodám uvedeným níže. Next.js je framework pro tvorbu single-page aplikací. Narozdíl od frameworku Angular není tak komplexní a je více konfigurovatelný. Pro zobrazování obsahu využívá Next.js knihovnu React s níž má autor práce osobní zkušenosti, proto nebyl zvolen Nuxt.js založený na Vue.js.

### Výhody Next.js

- Vykreslování na straně serveru.
- Projekty lze psát i v TypeScriptu.
- Lze jednoduše měnit předpřipravenou konfiguraci podle potřeb.
- Mnoho velkých firem si Next.js zvolilo na produkční nasazení svých projektů. [7]

### Express

Express je minimalistický a rychlý webový framework pro Node.js. [8] Pro implementaci serverové části byl zvolen pro jeho jednoduché API a také pro jeho velkou oblíbenost v komunitě [9]. Pro tento framework je dostupné velké množství doplňků a rozšíření.

## 2.5 Použité aplikace a nástroje

K řešení a usnadnění práce se autor rozhodl použít nástroje popsané v této sekci. U každého nástroje je popsáno, k čemu byl v projektu použit.

### Visual Studio Code

Visual Studio Code je editor pro psaní zdrojového kódu, vyvíjený společností Microsoft. Jelikož je postaven na JavaScriptu, je multiplatformní a je lépe provázaný s vývojem webových technologií. Pro implementaci prototypu webového portálu byl zvolen hlavně díky výborné podpoře jazyka TypeScript.

### Sketch

Sketch je grafický nástroj pro operační systém macOS se zaměřením na vektorovou grafiku. Kromě tvorby plakátů, log a jiné vektorové grafiky je vhodný také na návrh aplikací a webových stránek, pro které lze vytvářet interaktivní prototypy. V práci byl použit na tvorbu wireframu a byl v něm vytvořen základní návrh vzhledu jednotlivých částí a stránek portálu.

### GitUp

GitUp je jednoduchý grafický klient pro verzovací nástroj Git. Stěžejní funkcí této aplikace je přehledná vizualizace struktury repozitáře pro snadnou orientaci se zabudovaným mechanismem pro odvolání příkazů.

### draw.io

Autor shledává většinu dostupných programů pro tvorbu diagramů za nešťastné. Jedním z lepších nástrojů je online nástroj draw.io, použitý pro nakreslení diagramů v této práci. Nástroj nabízí širokou škálu typů diagramů s možností vytvoření vlastních stylů. Velkou výhodou je automatické ukládání do cloudových služeb a export do různých formátů.

## 2.6 Ochrana osobních údajů

Jelikož ochrana osobních údajů je jedním z nefunkčních požadavků, je třeba se seznámit s jedním z nejznámějších nařízení Evropské unie, které je shrnuto v této sekci.

25. května 2018 vstoupilo v platnost Obecné nařízení na ochranu osobních údajů (*zkratka GDPR*) a týká se všech subjektů, které nakládají s osobními údaji občanů Evropské unie. Osobní údaje jsou podle GDPR informace, které vedou k identifikaci konkrétní osoby jako je jméno, e-mail, ale i IP adresa nebo historie prohlížení. Mezi další povinnosti, které GDPR specifikuje, patří vhodné zabezpečení dat, aby případné riziko úniku bylo minimální. [10]

K tomu aby správce (ten kdo nakládá s osobními údaji) mohl zpracovávat osobní údaje musí mít oprávněný zájem ke zpracování nebo souhlas dotyčného subjektu. V okamžiku získání údajů musí správce poskytnout svou totožnost, účely zpracování a případné třetí strany. Dále je nutné subjekt seznámit s právy o odvolání souhlasu a možnosti podat stížnost u dozorového úřadu. Subjekt má právo na požadavek ke stažení osobních údajů, jejich výmaz nebo opravu. [11]

---

## Návrh

Tato kapitola popisuje návrh řešení problému správy fasády Linek. Čtenář se seznámí s celkovým konceptem Linek, tak jak byl navržen skupinou jejíž součástí byl i autor této práce. Další část kapitoly se zabývá případy užití, návrhem uživatelského rozhraní a návrhem aplikačního rozhraní. Konec kapitoly je věnován návrhu, jak řešit některé nefunkční požadavky jako je bezpečnost a ochrana osobních údajů.

### 3.1 Koncept Linky

Autoři světelné instalace Linky již od začátku počítali s interaktivitou. Cílem konceptu není jenom zapojit do dění Linek kolemjdoucí, ale i začínající a pokročilé programátory. Řešení tohoto problému bylo zadáno pracovní skupině v předmětu *Softwarový týmový projekt 1*. V týmovém projektu byl problém rozdělen na části a byl navržen způsob, jak mezi sebou budou jednotlivé části komunikovat [12]. Tři z těchto částí jsou podrobněji řešeny a implementovány v rámci bakalářských prací.

Před vysvětlením navrženého konceptu je třeba si nejdříve ujasnit několik následujících pojmů:

**Program**

Implementace, vytvořena vývojáři a spouštěna na světelné instalaci. Může se jednat například o hru nebo animaci.

**Aplikace**

Aplikace pro mobilní telefon nebo webová aplikace.

**Fasáda**

Světelná instalace.

**Core Server**

Jádro Linek, jehož součástí je Core API.

### Grafický Server

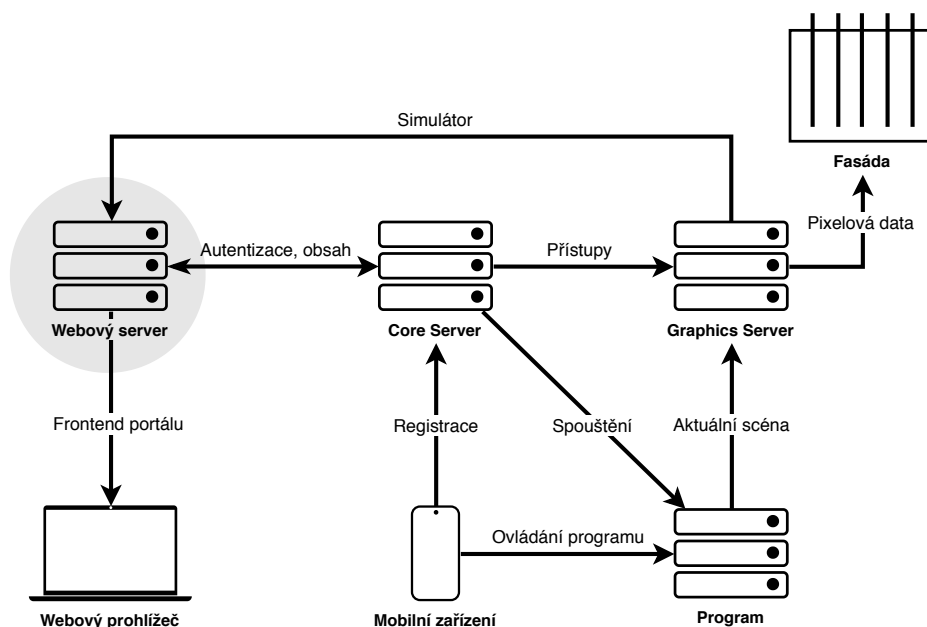
Server zpracovávající grafická data, jehož součástí je grafická knihovna.

Celý navržený koncept je graficky znázorněn v diagramu 3.1. Po technické stránce oproti starému řešení zůstane pouze část Fasády. Do Fasády budou zasílána pixelová data z Grafického serveru, která se mají na obrazovce zobrazovat. Tato data bude Grafický server získávat z dat poslaných z programů ve formě informací o tvarech a barvách. Grafický server bude zároveň poskytovat data pro simulátor, využitelný na webovém portálu.

Hlavní součástí Linky je Core Server, který vše řídí. Core Server má uloženou databázi uživatelů, programů, zařízení, běhových prostředí a výsledků her. Dále zajišťuje komunikaci s programy, webovým portálem, uživatelskými zařízeními a Grafickým serverem. Core Server určuje, kdy bude jaký program spuštěn. Jakmile je program spuštěn, může zasílat do Grafického Serveru data aktuální scény, kterou chce zobrazit.

Pokud je program interaktivní nebo se jedná o hru, bude přijímat příkazy, odesílané z připojených uživatelských aplikací. Tyto příkazy bude zpracovávat, jak uzná za vhodné. Po skončení nebo v průběhu může program do Core Serveru zaslat výsledky z jednotlivých „zápasů“.

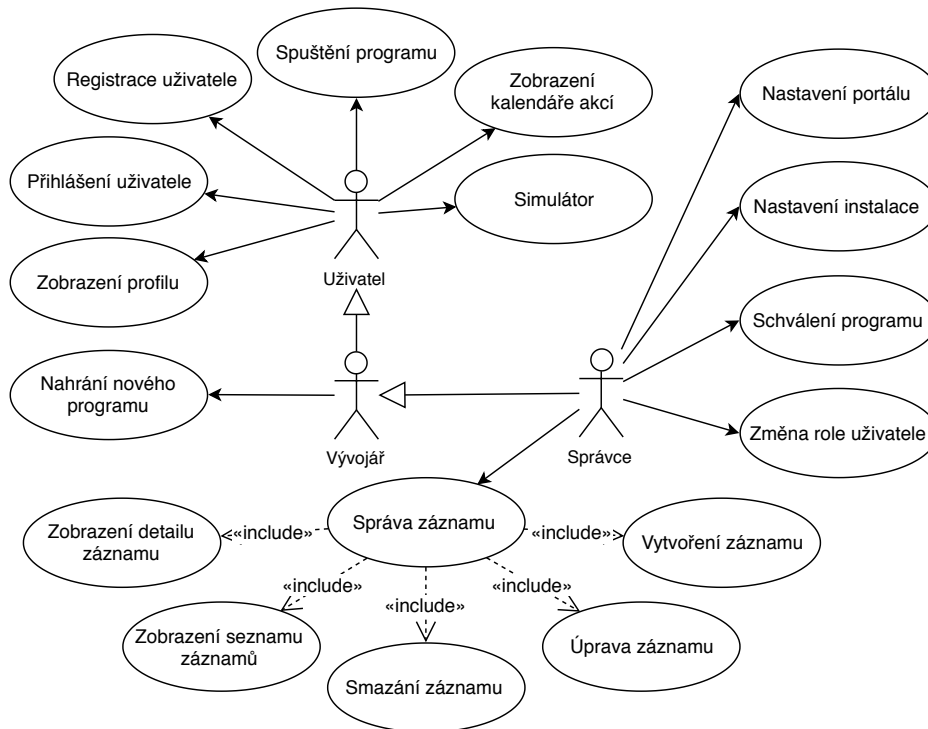
Další stěžejní částí Linky je webový portál, který umožňuje ovládat Core Server a prezentuje Linky široké veřejnosti. Webový portál může přistupovat k datům Grafického serveru pro prezentační účely nebo účely odladění programů vývojáři.



Obrázek 3.1: Koncept Linky

## 3.2 Případy užití

Případy užití a jejich účastníci jsou shrnuty v diagramu 3.2. Webový portál budou využívat běžní neregistrovaní návštěvníci (v diagramu jako Uživatel), Vývojáři programů a přístup k celému portálu budou mít Správci.



Obrázek 3.2: Diagram případů užití programu

### Správa záznamů

Tato část případů užití je určena pouze pro správce a všechny tyto případy budou dostupné po přihlášení. Správou záznamů se rozumí správa uživatelů, programů, statických stránek, akcí a prostředí.

#### UC1 Zobrazení seznamu záznamů

Seznam záznamů bude možné prohlížet a vyhledávat v přehledné tabulce. Tabulku bude možné řadit podle sloupců a při více záznamech bude možné data procházet mezi stránkami.

#### UC2 Zobrazení detailu záznamu

Některé typy záznamů (například stránky nebo soubory) bude možné prohlížet, aby správce viděl výsledek vytvářeného záznamu.

### 3. NÁVRH

---

#### **UC3 Úprava záznamu**

Po otevření editační stránky ze seznamu záznamů bude správce moci upravit hodnoty daného záznamu a uložit je. Hodnoty mohou být jakéhokoliv typu. Veškerý formátovaný obsah bude psán v jazyce Markdown. Do některých záznamů bude možné nahrávat soubory.

#### **UC4 Vytvoření záznamu**

Záznamy bude možné vytvářet formulářem podobným nebo dokonce stejným jako při úpravě.

#### **UC5 Smazání záznamu**

Z tabulky bude možné záznamy mazat.

### **Program**

#### **UC6 Nahrání nového programu**

Po přečtení dokumentace a vytvoření programu bude vývojář moci nahrát svůj program vyplněním formuláře.

#### **UC7 Schválení programu**

Jakmile vývojář nahraje program do portálu, správce Linek program může schválit nebo smazat.

### **Uživatel**

#### **UC8 Přihlášení uživatele**

Registrovaný uživatel se bude moci přihlásit stisknutím tlačítka na přihlašovací obrazovce.

#### **UC9 Registrace uživatele**

Pokud uživatel není uložen v databázi, bude zaregistrován prvním přihlášením.

#### **UC10 Zobrazení profilu**

Po přihlášení se uživateli zobrazí profil s jeho údaji. Odsud může provádět uživatelské akce.

#### **UC11 Změna role uživatele**

Správce Linek může povýšit registrované uživatele na vývojáře a na další správce.

### **Prezentace**

#### **UC12 Zobrazení kalendáře naplánovaných programů a akcí**

Uživatelé si budou moci prohlížet programy přidané ve frontě programů pomocí týdenního kalendáře.



**UC13 Zobrazení simulátoru**

Vývojářům bude k dispozici simulátor zobrazující aktuální scénu Linek pro odladění jejich programů. Simulátor bude také dostupný veřejnosti a ten bude zrcadlit aktuální scénu instalace.

**Nastavení****UC14 Nastavení instalace**

Správce může světelné instalaci ovládat jas, či měnit velikost obrázků.

**UC15 Nastavení portálu**

Správce může měnit navigaci webového portálu.

### 3.3 Diagram tříd

Z navržených případů užití lze již navrhnout diagram tříd. Diagram tříd 3.3 představuje, jaké objekty se v systému při implementaci použijí, obsahuje vztahy mezi nimi a definuje jejich atributy. Diagram tříd je možné použít pro vytvoření typů a databázového schéma v implementační části.

### 3.4 Aplikační rozhraní

Pro získávání a manipulaci se všemi daty klientskou částí portálu se použijí aplikační rozhraní (*anglicky Application Programming Interface*). Jelikož REST API využívá již Core Linek, bude vhodné když tento architektonický styl bude zachován. Autor práce zvažoval využít místo REST API řešení s dotazovacím jazykem GraphQL, které poskytuje volnější přístup k datům, ale to by dle jeho názoru přinášelo plno zbytečné complexity vzhledem k velikosti navrhovaného API.

API Linek jsou rozděleny na tři části. Pro autentizaci uživatelů, správu programů a další režii s programy slouží Linky Core API. Pro práci s obsahem webového portálu slouží Web Portal API. Koncept Linek ještě počítá s třetím API a to pro získávání obrazových dat – Graphics API. Návrh ani implementace Linky Core API není součástí této práce, ale pro funkci webového portálu je Linky Core API nezbytné.

Aplikačním rozhraním pro webový portál se zabývá následující sekce. Detailní a kompletní dokumentace API webového portálu je sepsána v externí službě SwaggerHub dostupná na adrese <https://app.swaggerhub.com/apis/jerrytm/Linky> a její offline verze je nahrána na přiloženou SD kartu.

### 3. NÁVRH

URL	Metoda	Oprávnění	Krátký popis
/pages	GET	Správce	Vrátí seznam stránek.
/pages	POST	Správce	Vytvoří novou stránku.
/pages	DELETE	Správce	Smaže více stránek najednou.
/pages/:id/edit	GET	Správce	Vrátí obsah stránky určený pro úpravu.
/pages/:id	PUT	Správce	Uloží stránku.
/pages/:id	DELETE	Správce	Smaže stránku.
/pages/:slug	GET	-	Vrátí obsah stránky určený pro veřejnost.

Tabulka 3.1: Přístupové body API stránek

URL	Metoda	Oprávnění	Krátký popis
/events/:date?	GET	Správce	Vrátí seznam akcí na týden.
/events	GET	Správce	Vrátí seznam akcí.
/events	POST	Správce	Vytvoří novou akci.
/events	DELETE	Správce	Smaže více akcí najednou.
/events/:id/edit	GET	Správce	Vrátí obsah akce určený pro úpravu.
/events/:id	PUT	Správce	Uloží akci.
/events/:id	DELETE	Správce	Smaže akci.
/events/:id	GET	-	Vrátí obsah akce určený pro veřejnost.

Tabulka 3.2: Přístupové body API akcí

URL	Metoda	Oprávnění	Krátký popis
/files	PUT	Vývojář, Správce	Nahraje nový soubor.
/files	GET	Správce	Vrátí seznam souborů.
/files	DELETE	Správce	Smaže více souborů najednou.
/files/:id/edit	GET	Správce	Vrátí obsah souboru určený pro správu.
/files/:id	DELETE	Správce	Smaže soubor.

Tabulka 3.3: Přístupové body API souborů

Veškerá komunikace probíhá pomocí HTTP požadavků, jejichž tělo je ve formátu JSON. Některé přístupové body vyžadují přihlášení uživatele s dostatečnými právy. Autentizace uživatelů probíhá pomocí hlavičky `x-Token` v požadavku. API server v odpovědi vrátí požadovaný výsledek, případně vrátí chybu doplněnou o odpovídající stavový HTTP kód.

```
{
  "title": "Technická bez aut",
  "visible": true,
  "startDate": "2019-05-03T10:52:34.099Z",
  "endDate": "2019-05-03T10:52:34.099Z",
  "url": "https://cvut.cz",
  "color": "success",
  "author": {
    "id": 1,
    "username": "PanSpravce",
    "imageUrl": "https://graph.facebook.com/01234567898765432/
      picture"
  },
  "description": "Linky jsou nyní dostupnější."
}
```

Vzor API požadavku vytvoření akce

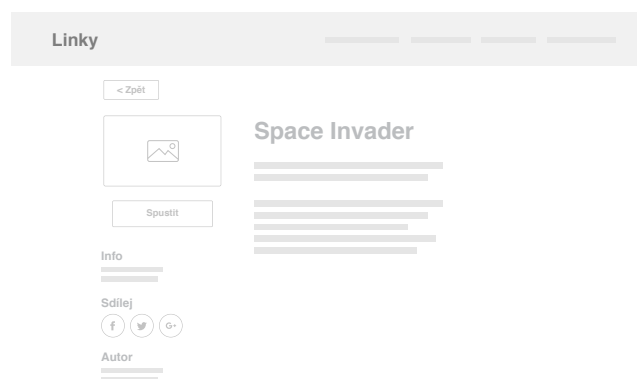
```
{
  "code": 500,
  "name": "ServerError",
  "message": "Error occured"
}
```

Vzor API odpovědi s chybou serveru

Kromě výše uvedených přístupových bodů poskytuje API webového portálu i přístup ke Core API. Pokud přijímaný požadavek nevyhovuje žádnému přístupovému bodu webového portálu, je žádost přesměrována na Core API. Dokumentace Core API a jeho přístupových bodů je sepsána v [13].

## 3.5 Uživatelské rozhraní

Při návrhu portálu je vhodné si rozvrhnout části webového portálu vytvořením wireframu. Manipulace s jednoduchým wireframem je mnohem jednodušší než s hotovým grafickým návrhem. Cílem wireframu je specifikovat základní strukturu stránek a přechody mezi nimi. Všechny návrhy wireframů jsou umístěny v příloze C na konci textu této práce.



Obrázek 3.4: Ukázkový wireframe detailu programu

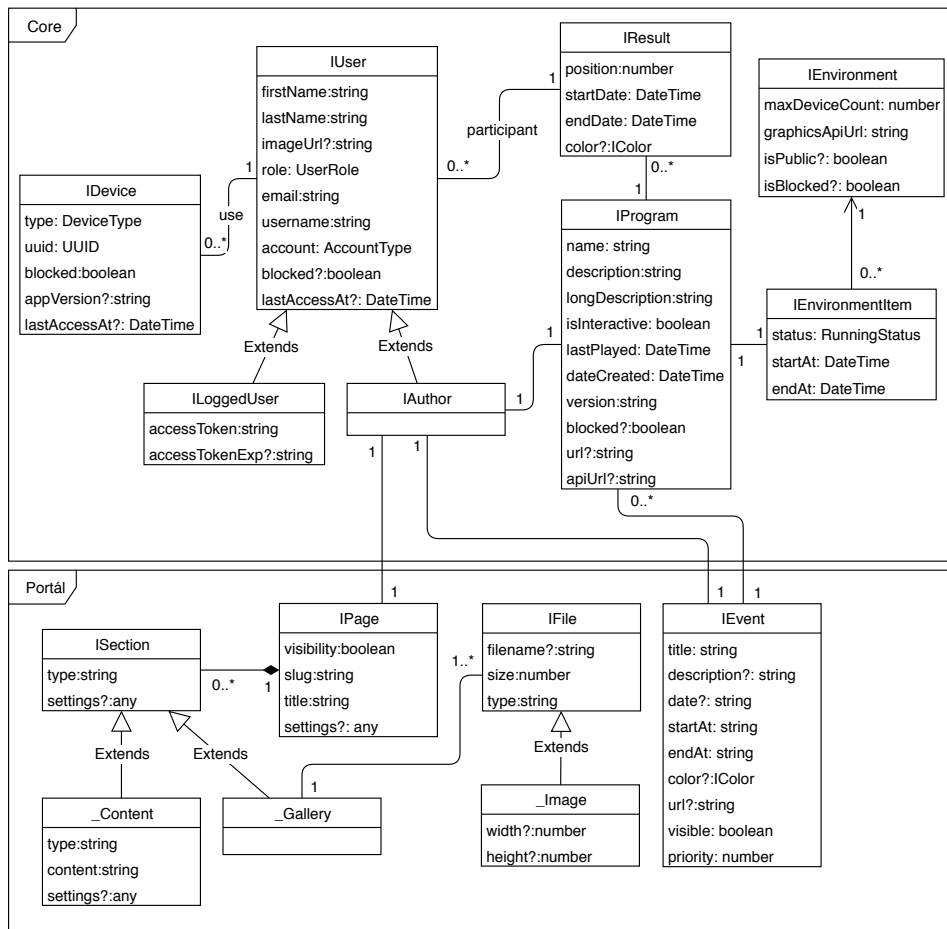
## 3.6 Ochrana osobních údajů

Neregistrovaný návštěvník nebude nijak sledován a webový portál o něm nebude ukládat žádná data. Informační povinnost pro registrované návštěvníky plynoucí z nařízení GDPR spočívá v sepsání dokumentu „Zásady ochrany osobních údajů“ osobou pověřenou správou obsahu. Pro tento účel je vyhrazena statická stránka dostupná z patičky webového portálu.

Práva pro stažení dat nebo jejich výmaz mohou být uspokojena v profilu uživatele. Tuto funkcionalitu webový portál pouze zprostředkovává a implementace je zodpovědností Core API, které shromažďuje osobní data uživatelů.

## 3.7 Bezpečnost

Autor při nasazení prototypu využívá zabezpečeného protokolu HTTPS, takže je veškerá komunikace s uživateli šifrována. Proti technice XSS, kde dochází ke vložení cizího kódu [14], je portál na většině místech chráněn ošetřováním vstupů knihovnou React. Výjimku tvoří formátovaný obsah, pro jehož vkládání je nutné dostatečné oprávnění. Jelikož API webového portálu nepoužívá SQL databázi, nemůže tak docházet k útokům typu SQL injection [15]. Nicméně podobnou techniku lze provést i u dokumentově orientované databáze. Tato hrozba by měla být odvrácena použitím knihovny Mongoose, která vstupní data ošetřuje.



Obrázek 3.3: Diagram tříd



---

# Implementace

Kapitola věnovaná implementaci začíná seznámením s použitými technologiemi a knihovnamí. V kapitole je představen styl kódu a rozvržení adresářové struktury. Zbytek kapitoly je věnován popisu vytvořených React komponent a průběžné integraci. Je třeba poznamenat, že vytvořený prototyp webového portálu je v době psaní práce dostupný na adrese <https://linky.sic.cz>.

## 4.1 Použité technologie

V analytické části byly rozebrány a představeny technologie, které se autor rozhodl použít při implementaci prototypu webového portálu. Tato sekce se věnuje představení technologií vyplývajících z analýzy.

### Node.js

Node.js je prostředí umožňující spouštět JavaScript mimo webový prohlížeč. JavaScript se díky tomuto prostředí dá používat i na serveru a ne jen u klienta. [16] Kromě využití pro běh serverů jej lze využít nejen pro vývoj desktopových aplikací a nástrojů pro příkazovou řádku.

### React

React je knihovna pro vytváření uživatelského rozhraní vyvíjená společností Facebook a veřejnou komunitou. Používá se převážně na tvorbu webových aplikací, ale v kombinaci s dalšími knihovnamí může být použita i pro vývoj nativních mobilních aplikací nebo aplikací pro televize. [17] Největší výhodou Reactu je tzv. virtuální DOM. Zjednodušeně řečeno si React udržuje strom prvků, které mají být vykresleny. Pokud ve stromu nastane změna, React se tyto změny co nejefektivněji pokusí zpropagovat do hostitelských instancí. Takovými instancemi mohou být DOM uzly ve webovém prohlížeči nebo UIView v operačním systému iOS. [18]

### JSX

S použitím knihovny React je pro přehlednost kódu téměř nezbytné využít JSX. Díky tomuto rozšíření jazyka JavaScript (resp. TypeScript) lze zapisovat definice komponent Reactu ve formátu podobném jazyku XML. Kód pak připomíná HTML v JavaScriptu.

## 4.2 Kód

### 4.2.1 Styl kódu

Při implementaci prototypu webového portálu bylo dbáno pravidel psaní zdrojového kódu, která jsou shrnuta na Git repozitáři společnosti Airbnb dostupném na adrese <https://github.com/airbnb/javascript>. Dodržování těchto pravidel je důležité pro čitelnost a jednotnost zdrojového kódu. V projektu webového portálu je použit nástroj `tslint`, který se stará o dodržování daných pravidel.

### 4.2.2 Adresářová struktura

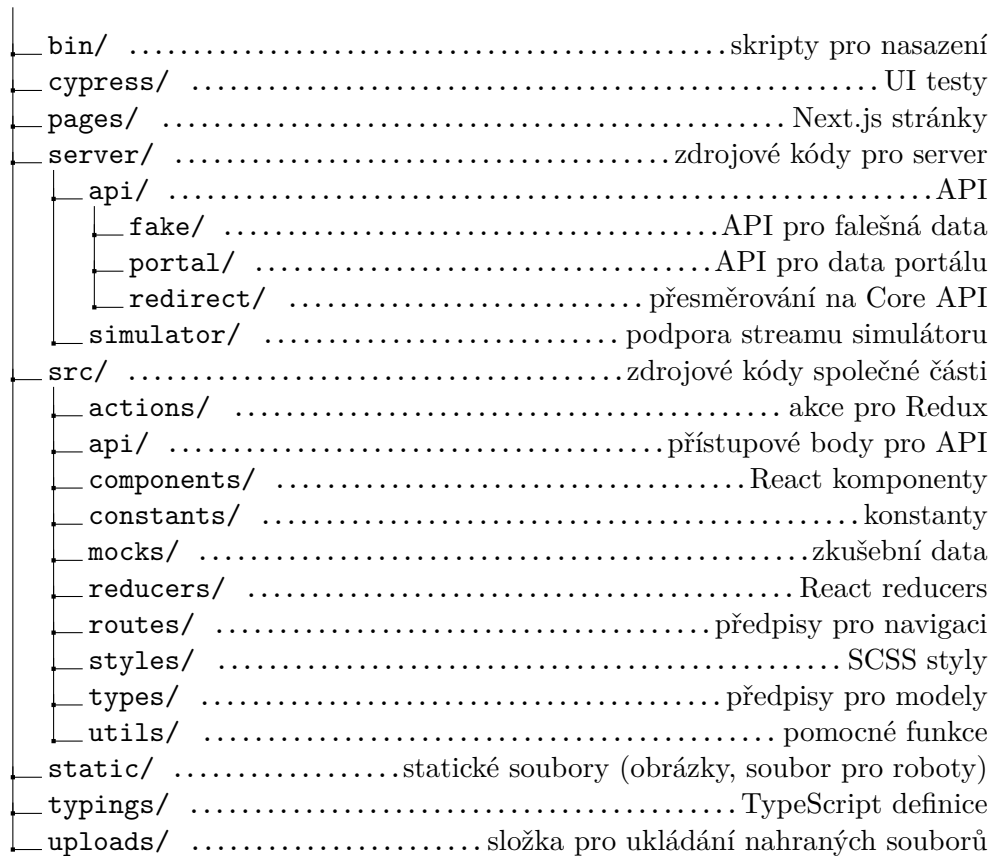
Základ aplikace vychází z předpřipravené kostry vytvořené autory frameworku Next.js. Tento framework navíc vyžaduje, aby se jednotlivé stránky nacházely ve složce `pages` a statické soubory ve složce `static`. Na zbytek adresářové struktury nejsou kladeny žádné nároky a struktura je zcela na vývojáři. Zdrojový kód samotného portálu je rozdělen na dvě části – serverovou a společnou.

První úroveň serverové části tvoří rozdělení API na typy. Druhá úroveň portálového API je tvořena složkami entit.

Ve společné části se nachází složka `components`. Tato složka obsahuje znovupoužitelné React komponenty. Složka `pages`, obsahuje všechny ostatní React komponenty určené pro jednotlivé stránky. Složka `routes` slouží pro definice navigace. Soubory pro knihovnu Redux jsou uloženy ve složkách `actions` a `reducers`.

V kořenové složce projektu jsou složky pro UI testy, statické soubory a skripty pro nasazování.





Obrázek 4.1: Adresářová struktura projektu

### 4.2.3 NODE\_PATH

Ke většině souborům v rámci projektu se přistupuje absolutní cestou se základem ve složce `src`. Toto nastavení se dá změnit proměnnou `NODE_PATH`, která je definována v souboru `.env`. O tomto souboru pojednává jedna ze sekcí v této práci. Výhodou absolutního adresování je hlavně čitelnost importovacích příkazů a orientace v projektu.

## 4.3 Serverová část

### 4.3.1 API

Podle návrhu je implementováno API pro stránky, akce, soubory a nastavení. Jednotlivé entity jsou rozděleny do svých složek. Každá složka obsahuje řadič (controller), databázové schéma s validačními pravidly (model) a směrovač (router). Každý řadič obsahuje funkce odpovídající REST API metodám,

volané ze směrovače. Pokud API požadavku nevyhovuje žádná funkce, je požadavek přeposlán do Linky Core API.

Většina implementovaných funkcí odpovídá operacím BREAD (zkratka pro operace seznamu, čtení, úpravy, přidání a mazání záznamů). K běžným BREAD operacím přibyla funkce pro nahrávání souborů a funkce pro zobrazení akcí do kalendáře.

Operace pro získání seznamu záznamů přijímají v požadavku parametry pro řazení, vyhledávání a stránkování. Řadit lze pomocí parametru `sortBy` obsahujícím klíče, podle kterých se má řadit. Vyhledávání proběhne v klíčích podle dané entity při specifikování parametru `search`. Stránkování je stejné pro všechny entity a funguje podobně jako v relačních databázích s parametry `limit` a `offset`. Tyto parametry zpracovává pomocná funkce `getFilters`.

Pro testovací účely byla vytvořena část falešného API, mapující Linky Core API. Tato část nijak nezasahuje do reálného serveru a kromě testování ji lze využít i při vývoji webového portálu.

### 4.3.2 Definice schémat

Podle diagramu tříd 3.3 byla odvozena schémata pro databázový framework Mongoose. Podle těchto schémat jsou pak ukládána data do databázových kolekcí. Ke schématům jsou přidána validační pravidla zajišťující integritu dat.

### 4.3.3 Uživatel

Po přihlášení je uživateli do počítače uložena cookie s unikátním klíčem přiděleným Core API Linek. Každý požadavek vyžadující vyšší uživatelské oprávnění je ověřen. Zjištění aktuální role uživatele probíhá přes přístupový bod `/users/current` Core API Linek.

## 4.4 Klientská a serverová část

Jelikož framework Next.js podporuje vykreslování stránek na straně serveru i klienta, bude většina zdrojového kódu společná pro obě části. React komponenty musí být přizpůsobené pro běh i mimo webový prohlížeč.

### 4.4.1 SSR

Celý portál až na malé výjimky podporuje načítání na straně serveru (Server Side Rendering). To znamená, že se návštěvníkovi portálu bez zapnutého JavaScriptu nezobrazí pouze bílá stránka, ale web bude stále dostupný a je navíc čitelný pro internetové vyhledávače.

### 4.4.2 `getInitialProps`

K načítání asynchronních dat zobrazitelných na straně serveru je frameworkem Next.js určena funkce `getInitialProps`. V rámci sjednocení načítání dat, byla vytvořena funkce `getFetchData`. Tato funkce jako argumenty přijímá asynchronní funkce, které mají být přijímány. Při načtení stránky tato funkce zjistí, jakou akci má provést. Podle typu akce načte data a vrátí v jednotném formátu. Pokud při načítání dojde k jakékoliv chybě bude ošetřena.

### 4.4.3 `Redux State`

Přihlašování a informace o aktuálně přihlášeném uživateli se ukládají do globálního stavu aplikace, odkud jsou dostupné všem komponentám v Reactu. Globální stav aplikace je řešen knihovnou Redux.

### 4.4.4 `Router`

Pro odkazování na jinou stránku umístěnou na webovém portálu slouží komponenta `Link`, která má stejnou funkčnost jako v knihovně `next-routes` (<https://github.com/fridays/next-routes>). Odkazy z této komponenty při kliku nezpůsobí znovunačtení celé stránky. `Link` dostane jednu resp. dvě vlastnosti `route` resp. `params`. Veškeré cesty webového portálu jsou definovány v souboru `src/routes.ts` a jazyk TypeScript hlídá, aby se do vlastnosti `route` nedostaly neexistující cesty. Do vlastnosti `params` patří parametry, které jsou následně předány do předpisu cesty. Komponenta `Link` z těchto dvou vlastností dokáže vygenerovat pro uživatele čitelnou URL adresu.

### 4.4.5 `Datová tabulka`

Pro manipulaci se záznamy slouží základní komponenta `Table`. Vstupními vlastnostmi takové komponenty jsou stránkovaná data získaná z API s informacemi o aktuální stránce a stavu řazení. Potomci v definici komponenty určují, jaké sloupce budou zobrazeny. Data v buňkách tabulky se získávají podle jména sloupce, popřípadě vlastní funkcí.

`Table` podporuje i dávkové operace. K tomu je třeba, aby byl tabulce předán atribut `batch` a mezi potomky se nacházel sloupec `TableCheckColumn`, umožňující označit řádky. Příklad definice datové tabulky je v ukázce kódu 4.1.

### 4.4.6 `Formik`

Práci s formuláři zjednodušuje knihovna `Formik`. Zrychluje vytváření formulářů, zajišťuje jejich validaci a zachovává čitelnost jejich definic. Pro účely webového portálu byla vytvořena komponenta `Formiko`, která rozšiřuje chování knihovny `Formik` o odesílání dat formulářů na API. Formulářová pole byla rozšířena o pole pro nahrání souboru a pole propojující Markdown editor.

```
<Table<IPage> data={data} parameters={parameters} baseUrl={pageUrl}>
  <TableCheckColumn>#</TableCheckColumn>
  <TableColumn
    name="title"
    sortable
    onClick={this.handleClick}
    render={(page: IPage) => <strong>{page.title}</strong>>
    Název
  </TableColumn>
  <TableColumn
    name="createdAt"
    sortable
    render={(item: IPage) => formatDate(item.createdAt)}>
    Datum vytvoření
  </TableColumn>
  <TableColumn
    name="author"
    sortable
    render={(item: IPage) => getFullName(item.author)}>
    Autor
  </TableColumn>
  <TableActionsColumn />
</Table>
```

Ukázka kódu 4.1: Použití komponenty Table

### 4.4.7 Simulátor

Autor práce si nad rámec zadání přidal mezi funkční požadavky zobrazení simulátoru. Implementovaný simulátor má zrcadlit aktuální scénu Linek v reálném čase. Simulátoru lze přepínat typy získávání dat a několik typů je v portále implementováno.

Přehrávání živé scény je dosaženo díky protokolu WebSockets. Při získání dat ze zdroje jsou ihned vysílána všem sledujícím divákům. Pixelová data jsou zasílána ve formátu pole barev po čtyřech bajtech a pro jednoduchost nejsou nijak komprimována. V případě změny programu nebo potřeby zaslání zprávy divákům lze zaslat data ve formátu JSON. V budoucnu lze implementovat i přepínání běhových prostředí Linek a náhled scény programů, které na Linkách neběží.

Pro náhled, jak by na světelné instalaci vypadal obrázek, je vytvořen simulátor umožňující nahrát soubor v obrazovém formátu.

## 4.5 Continuous Integration a verzování

Při vývoji softwarového projektu se určité činnosti opakují. Pro ušetření času a úsilí je více než vhodné si některé činnosti zautomatizovat. Jejich automatizaci se věnuje tato sekce.

### 4.5.1 Verzování

Každá změna by se měla při vývoji portálu ukládat do tzv. revize pomocí verzovacího nástroje. Prototyp webového portálu je verzován v Git repozitáři uloženém na serveru Gitlab FIT ČVUT. Je zvykem [19], že se v životním cyklu softwaru udržují dvě základní větve:

- `develop` - Do této hlavní větve se přidávají nové funkce, opravují chyby apod. Automatické testy nemusí nutně procházet.
- `master` - Produkční větev se ve vhodných okamžicích ideálně spojuje s důkladně otestovanou větví `develop`.

### 4.5.2 Průběžná integrace

Průběžná integrace je proces softwarového vývoje, kde členové vývojového týmu často integrují kód. Každý takový příspěvek je automaticky zkontrolován neobsahuje-li chyby či jiné nedostatky. Tato praktika pomáhá snížit množství problémů při společném vývoji softwaru. [20] V případě prototypu webového portálu se průběžnou integrací rozumí sestavení aplikace a její otestování.

### 4.5.3 Průběžné nasazení

Jak je popsáno v předchozí podsekci, při každém odeslání revize na server je spuštěna sada úkolů, které portál sestaví a otestují. Pokud předchozí úkoly proběhnou v pořádku, spustí se další úkol, který zajistí odeslání nové verze na vývojový (resp. produkční) server a portál automaticky spustí. Pokud jakákoliv část selže, je o tom vývojář ve většině systémů ihned zpraven e-mailem. Tomuto procesu se říká Continuous Delivery (průběžné nasazování).

### 4.5.4 Soubor pro automatizaci

Součástí služby Gitlab, použité pro verzování, je nástroj Gitlab CI umožňující spouštět průběžnou integraci a průběžné nasazení. Nastavení tohoto nástroje se realizuje souborem `.gitlab-ci.yml`, který je napsán v jazyce YAML. Tento jazyk se převážně používá v konfiguračních souborech. Do tohoto souboru vývojář specifikuje příkazy, která se mají během průběžné integrace spustit.

Základ souboru tvoří tzv. úkoly (*anglicky jobs*). Úkoly mohou být rozděleny do fází. Fázemi obvykle bývá `build`, `test` a `deploy`. Každý úkol musí obsahovat příkazy, které se mají vykonat. Příkazy se vykonávají postupně a pokud jakýkoliv příkaz skončí chybou, skončí celá fáze. Spuštění úkolů lze omezit pouze na určité větve. V úkolech se dají také specifikovat složky, které se mají uchovat pro další běhy nebo složky, které se mají uchovat jako artefakty. Artefakty mohou být například videa pořízená při testování. [21]

### 4.6 Nastavení vývojového prostředí

Tato sekce seznamuje čtenáře s postupy, jak nainstalovat potřebné nástroje, spustit vývojový server na svém zařízení a pokračovat tak v dalším rozvoji prototypu webového portálu.

#### 4.6.1 Stažení projektu

Zdrojový kód byl postupně umisťován do Git repozitáře dostupného na adrese <https://gitlab.fit.cvut.cz/linky/web-portal> a odsud je možné stáhnout nejnovější verzi a pokračovat tak v dalším vývoji. Zdrojový kód je také umístěn na příložené médium.

#### 4.6.2 Instalace závislostí

Pro spuštění serveru webového portálu Linky je potřeba nainstalovat následující potřebné nástroje a technologie:

- Node.js (verze alespoň 10.8.0)
- MongoDB
- yarn

Postup instalace ponechává autor této práce na čtenáři, avšak do Git repozitáře byl sepsán stručný výtah instalačních příkazů určených pro spuštění na operačním systému Linux Debian verze Jessie. Tento výtah se nachází v souboru `diary.md` v kořenové složce repozitáře.

#### 4.6.3 Proxy server

Pro současný běh více serverů na jedné adrese, možné rozložení zátěže a další správu je využít proxy server `nginx`. Na produkčním serveru portálu zajišťuje `nginx` přeměrování HTTP požadavků na bezpečnější protokol HTTPS, spravuje SSL certifikáty a vystavuje Linky Core API pod svou doménou.

#### 4.6.4 SSL certifikáty

Webový portál lze spustit i bez nutnosti HTTPS, žádné certifikáty nejsou třeba. Pro lokální vývoj je tato možnost tedy postačující. Na druhou stranu je výrazně doporučeno využívat zabezpečeného protokolu HTTPS pro produkční nasazení, kde jsou certifikáty nedílnou součástí. Autor práce využil certifikační autority Let's Encrypt (<https://letsencrypt.org>). Certifikáty jsou předány do konfiguračního souboru serveru `nginx`.

### 4.6.5 Soubor `.env`

Soubor `.env` obsahuje konfigurační proměnné pro aktuální běhové prostředí. Nastavení tohoto souboru spočívá v přepísování připravené šablony `.env.example` a upravení proměnných podle aktuální situace. Z bezpečnostních důvodů není doporučeno tento soubor nikde sdílet, jelikož obsahuje citlivá data. V UI testech spouštěných při průběžné integraci webového portálu Linky se pro účely testování spouští server, jehož konfigurace vychází ze souboru `.env.test`. Všímavý čtenář zjistí, že v tomto souboru chybí `DATABASE_PASSWORD`. Tuto proměnnou přidává služba GitLab z nastavení projektu. Význam proměnných v konfiguračním souboru je vysvětlen v tabulce 4.1.

Proměnná	Příklad	Vysvětlení
<code>NODE_ENV</code>	<code>production</code>	Typ prostředí
<code>NODE_PATH</code>	<code>./src</code>	Cesta ke zdrojovým souborům. Používá se jako základ při načítání modulů v Node.js.
<code>ENVIRONMENT_URL</code>	<code>https://linky.cz</code>	Veřejná adresa webového portálu bez lomítka.
<code>API_URL</code>	<code>http://192.168.0.2</code>	Adresa na Core API.
<code>DATABASE_URL</code>	<code>192.168.0.3</code>	Adresa k MongoDB.
<code>DATABASE_NAME</code>	<code>linky</code>	Název databáze.
<code>DATABASE_USER</code>	<code>linky</code>	Uživatel s přístupem k uvedené databázi.
<code>DATABASE_PASSWORD</code>	<code>kobylamamalybok</code>	Heslo do databáze.
<code>DATABASE_AUTH_SOURCE</code>	<code>admin</code>	Autorizační databáze.
<code>PORT</code>	<code>3000</code>	Port webového portálu.
<code>HOST</code>	<code>localhost</code>	Adresa pro naslouchání.
<code>UPLOADS_DIR</code>	<code>uploads/</code>	Složka pro ukládání nahraných souborů.
<code>USE_FAKE_API</code>	<code>false</code>	Příznak, má-li se použít falešné Core API místo reálného.

Tabulka 4.1: Proměnné v souboru `.env`

### 4.6.6 MongoDB

Novou databázi není třeba nijak zvlášť konfigurovat. Kolekce se vytvoří až za běhu v případě potřeby. Jedinou výrazně doporučenou konfigurací po instalaci je pouze zabezpečení databáze. Pro běh portálu je nutná pouze jedna databáze, avšak je vhodné mít pro každé prostředí jednu databázi s uživatelem a unikátním heslem. Pro přihlášení do databáze se využívá tzv. autorizační databáze, která má uloženy informace o uživateli celé databáze.

### 4.6.7 Spuštění

Pokud jsou všechny závislosti správně nainstalovány a nastaveny, je potřeba nainstalovat balíčky z repozitáře npm (balíčkovací systém) pomocí příkazu `yarn install`. Po nainstalování závislostí by již mělo být možné spustit vývojovou verzi portálu příkazem `yarn dev`. Pro odladění API se může hodit příkaz `yarn dev:api`, který nespouští celý webový portál, ale pouze API. Spuštění produkční verze portálu lze po sestavení `yarn build` spustit příkazem `yarn start`. Spouštění na produkčním serveru zajišťuje průběžné nasazování.



---

# Testování

Tato kapitola se zaměřuje na vhodné techniky testování jak zdrojového kódu, výsledného portálu, ale i závěrečného testování použitelnosti. Použití testů je důležitá součást při vývoji softwaru a má za cíl snížit riziko chyb v produkční verzi softwaru a zvýšit jeho celkovou kvalitu.

## 5.1 Automatické testy

### 5.1.1 Jednotkové testy

Jednotkové testy jsou prvními prováděnými testy zdrojového kódu. Cílem jednotkových testů je otestovat samostatné části systému nezávisle na sobě. Jednotkové testy v ideálním případě píše vývojář před samotným vývojem jednotky. Takovému způsobu vývoje se říká „Test driven development“. Jednotkou obvykle bývají funkce, metody nebo komponenty a testy by měly pokrývat co nejvíce možných stavů. V projektu webového portálu jsou jednotkové testy připraveny převážně pro komponenty Reactu.

### 5.1.2 UI testy

Testy uživatelského prostředí probíhají v reálném prostředí na reálném zařízení. Jejich cílem je simulací reálného chování uživatele otestovat veškeré interakce aplikace. V projektu webového portálu UI testy pokrývají případy užití sestavené v návrhové části této práce.

### 5.2 Nástroje pro automatické testování

#### TypeScript

TypeScript není přímo testovacím nástrojem, avšak jazyk JavaScript nepodporuje statické typování a může tak docházet k často člověkem nerozeznatelným chybám. Díky TypeScriptu lze vypustit psaní testů věnovaných na kontrolu typů.

#### Jest

Pro snadné spuštění jednotkových testů je použit framework Jest. Jest je vyvíjen společností Facebook a poskytuje tak mnoho nástrojů pro jednoduché testování React aplikací. [22]

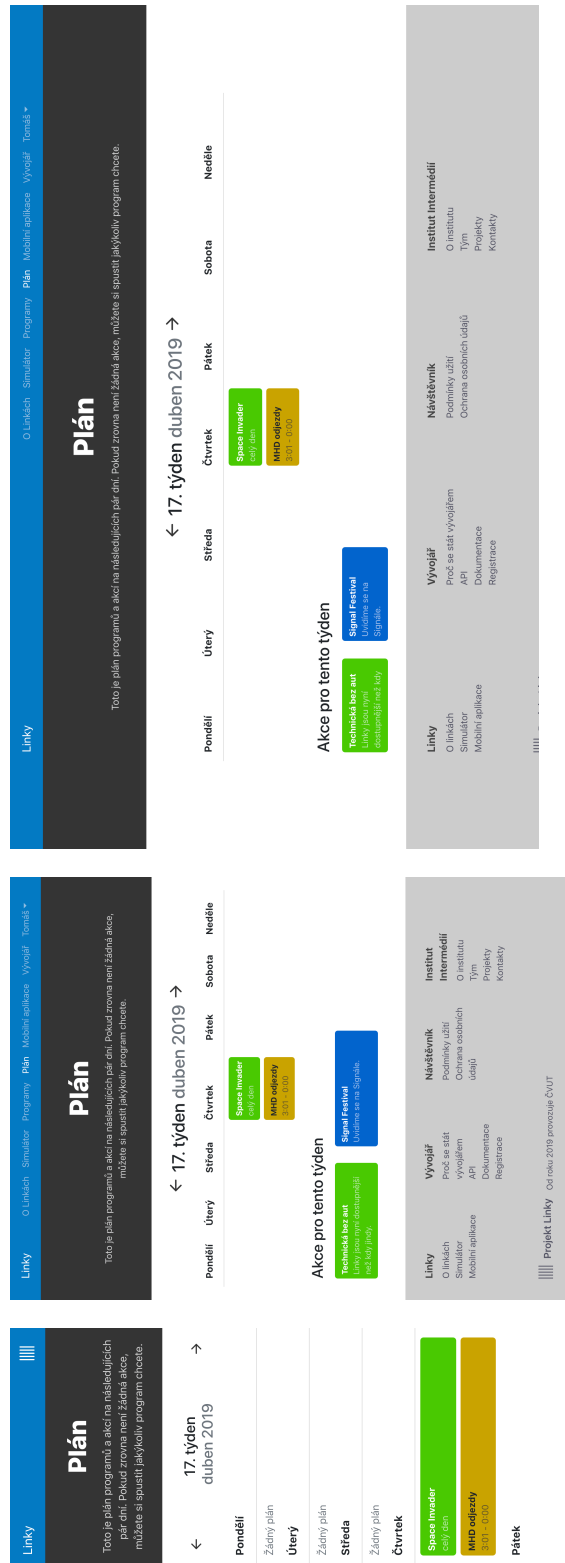
#### Cypress

Cypress je nástroj na testování webových aplikací. Nástroj dokáže testovat reálné použití aplikace ve webovém prohlížeči a z testování pořizuje snímky a videa. Tyto testy lze pouštět také při průběžné integraci. V době psaní této práce nástroj zatím dokáže provádět testy pouze v prohlížeči Chrome. [23]

### 5.3 Testování responzivity

Jeden z nefunkčních požadavků v analytické části požaduje přizpůsobení pro všechny velikosti obrazovek. K tomuto požadavku bylo při implementaci přihlédnuto. Otestování responzivity je však složitější než zmíněné jednotkové testování, jelikož vyžaduje ruční kontrolu a nelze jej jednoduše automatizovat.

Mnoho problémům s responzivním zobrazením lze předejít použitým CSS frameworkem Bootstrap a jeho systémem mřížkového rozložení prvků. Další možné problémy lze odhalit při UI testování, kde se testy pouštějí při různých velikostech obrazovky. Zmíněný nástroj na UI testování Cypress provádí kontrolu viditelnosti a pokud není testovaný prvek vidět, nástroj tento problém odhalí. Všechny ostatní problémy lze zkontrolovat ručně při závěrečném testování.



Obrázek 5.1: Porovnání respzivního zobrazení na různých velikostech obrazovky

### 5.4 Testování použitelnosti

#### 5.4.1 Heuristická analýza

Testovat prototyp lze také bez využití uživatelů pomocí heuristické analýzy, což je soubor poznatků a zkušeností, jak jsou uživatelé zvyklí se systémem pracovat a co od něho očekávají. Nejznámější heuristickou analýzou je Nielsenova heuristická analýza. Tato metoda tvoří deset základních pravidel, které pomáhají vytvořit systém jednoduchý a efektivní na práci. [24] Těchto deset pravidel je shrnuto v následujícím seznamu [25]:

- 1. Viditelnost stavu systému** Uživatel by měl v jakémkoliv stavu aplikace vědět, co se děje. Pokud je například potřeba načíst data ze vzdáleného serveru, je vhodné včas zobrazit ukazatel průběhu načítání.
- 2. Propojení systému a reálného světa** Aplikace by měla obsahovat součásti na které jsou uživatelé zvyklí. Například pohyby objektů by měly být znázorněny animacemi.
- 3. Uživatelská kontrola a svoboda** Aplikace by měla poskytovat uživatelům volnost a umožnit odvolat akce.
- 4. Konzistence a pravidla** Aplikace by měla držet jednotný jazyk pochopitelný a blízký uživatelům.
- 5. Prevence chyb** Systém by měl předcházet chybám a nedovolit uživatelům dostat se do stavu do kterého nechtějí.
- 6. Rozpoznávání místo vzpomínání** Systém by neměl uživatele nutit pamatovat si jakékoliv postupy nebo informace. Tam, kde je potřeba by mělo být vše potřebné vysvětleno.
- 7. Flexibilní a efektivní použití** Zkratky, které nemusí být na první pohled viditelné, mohou zrychlit práci zkušenějším uživatelům. Dobrým příkladem jsou klávesové zkratky.
- 8. Minimalistický design** Uživatel by neměl být zahlcován zbytečnými informacemi. Uživatelské prostředí by mělo být navrženo tak, aby neodvádělo pozornost od své primární akce.
- 9. Pomoc uživatelům pochopit a dostat se z chyb** Pokud dojde k jakékoliv chybě, měla by být uživateli dostatečně vysvětlena. Uživateli například nemusí stačit zobrazení chybové hlášky „500 Internal Server Error“.
- 10. Náповěda a návody** I když je aplikace sebevíc intuitivní a nevyžaduje návodu, je důležité zprostředkovat uživatelům pomoc. Taková pomoc ve formě dokumentace by měla být stručná a prohledávatelná.

Hodnocení Nielsenovy heuristiky by měli v ideálním případě provádět odborníci na použitelnost, avšak tato možnost může být nákladná. S větším počtem hodnotitelů se zvyšuje počet nalezených problémů. Doporučuje se 3 až 5 osob. Přínos dalších osob se postupně snižuje. [26]

### 5.4.2 Zhodnocení

Autor si na základě případů užití vytvořil scénář průchodu, podle kterého má být prováděno uživatelské testování. Scénář se skládá z úkolů, které mají být vykonány. Jak mají být vykonány, je na testovacím subjektu. Testování provedl autor sám a zaměřil se na výše uvedená pravidla heuristické analýzy. Získaná data s návrhy na opravu jsou sepsána v části s nedostatky.

#### Scénář průchodu pro správce

1. Přihlásit se pomocí Facebooku.
2. Upravit nadpis na hlavní stránce a uložit.
3. Posunout akci „Farmářské trhy“ na další týden.
4. Smazat několik souborů.
5. Odhlásit se.
6. Zjistit informace o programu „Space Invaders“.
7. Získat seznam všech hratelných programů.
8. Najít akce na příští týden.

#### Nedostatky

- **Při načítání chybí ukazatel načítání**  
Při procházení webu není uživateli k dispozici žádný ukazatel, že se obsah načítá a při pomalém internetovém připojení se dlouhou dobu nic neděje.
- **Neužitečná hláška o ukládání**  
Na stránce s úpravou záznamu je informace o času posledního uložení „Naposledy uloženo 23. 4. 2019 20:12:53“. Uživatel by v tuto chvíli čekal intuitivnější hlášku „Naposledy uloženo před 5 minutami“.
- **Klávesové zkratky**  
Web nepodporuje žádné klávesové zkratky pro urychlení práce. Hodila by se například klávesová zkratka pro uložení záznamu, na kterou jsou uživatelé zvyklí z jiných aplikací.

## 5. TESTOVÁNÍ

---

- **Informace o aktuální poloze v navigaci**  
Hlavní navigace nezvýrazňuje aktuální položku, kde se uživatel nachází.
- **Potvrzení destruktivní akce**  
Uživatel po kliknutí na tlačítko s destruktivní akcí musí pro potvrzení na tlačítko kliknout znovu. O nutnosti kliknutí znovu je dostatečně informován.
- **Tlačítko „Uložit“ je vždy, když je potřeba, vidět**  
Nejdůležitější akcí při úpravě záznamu je uložení. Spuštění této akce by mělo být vždy dostupné.
- **Okamžitá odezva o platnosti formulářového pole**  
Pokud se uživatel pokusí odeslat nesprávně vyplněný formulář, zapne se okamžitá odezva o platnosti formulářových polí.

Na výše uvedené nedostatky bylo přihlédnuto a byly v prototypu opraveny.

---

## Budoucí práce

V této práci byl analyzován, navržen webový portál pro správu světelné instalace Linky a následně byl implementován jeho prototyp. Již od začátku se počítalo s budoucím rozšiřováním funkcí. Tato kapitola shrnuje, co by se dalo do webového portálu přidat a co zlepšit.

### 6.1 Funkce

#### 6.1.1 Články a dokumentace

V budoucnu může vzniknout potřeba vydávat aktuality či tiskové zprávy. Pro tento účel by se hodilo lepší prostředí než pro vytváření statických stránek. Stejná potřeba může vzniknout i pro dokumentaci API programů, Core Linek a grafické knihovny.

#### 6.1.2 Výsledky z programů

Vyvojáři programů mohou pro Linky vytvářet kromě různých vizualizací i hry. Ke zvýšení soutěživosti může být přidána funkce ukládání a zobrazování výsledků z her. Výsledky pak mohou být veřejně umístěny na webovém portálu ve formě výsledkové tabule.

### 6.2 Virtualizace

Pro vývoj, testování a nasazení by se jistě hodilo použití virtualizačního softwaru. Představou autora práce je použít nástroj Docker a vytvořit izolované prostředí se všemi nainstalovanými a připravenými závislostmi a zjednodušit tak zprovoznění běhového prostředí pro webový portál. K realizaci však v době začátku psaní práce nedošlo, jelikož znalosti autora nebyly dostatečné k vytvoření virtualizačního kontejneru a spojení s webovým portálem.





---

## Závěr

Cílem práce bylo zanalyzovat problém správy světelné instalace Linky, navrhnout a implementovat webový portál a následně funkce portálu otestovat. Portál umožňuje správcům instalaci jednoduše ovládat, vývojářům nahrávat své programy a propagovat veřejnosti světelnou instalaci. Kromě požadavku navrhnout přívětivé a responzivní administrační rozhraní pro správu portálu se autor práce zaměřil i na bezpečnost portálu.

Analýza existujících řešení ukázala, že myšlenka portálu pro interaktivní instalaci není tak unikátní, jak se zdála. Na druhou stranu je zajímavé, že na danou situaci autor práce pohlížel velmi podobně, jako autoři luminous ještě před tím, než byla provedena analýza.

Webový portál bude třeba udržovat pouze aktualizací obsahové části pro návštěvníky a schvalováním nových programů. Řešení by se jistě dalo využít i na podobných instalacích umístěných ve veřejných prostorách.

Autor práce při řešení využil své základní znalosti nabitě při studiu. Analýzou a zkoušením nových technologií navíc získal cenné zkušenosti vývoje webového portálu.



---

## Literatura

- [1] HYBLER, J.: Světelný design v kostce – Část 24 – Světelná fasáda – projekt Linky [online]. *Časopis Světlo - Odborné časopisy*, 2019, [cit. 2019-02-18]. Dostupné z: <http://www.odbornecasopisy.cz/svetlo/clanek/svetelny-design-v-kostce-cast-24-svetelna-fasada-projekt-linky--1715>
- [2] iion: Darling Quarter Luminous [online]. 2011, [cit. 2019-04-22]. Dostupné z: <http://www.myinteractivecity.com>
- [3] KUŽNÍK, J.: Na pražské tramvaji jezdí zvláštní vánoční strom. Rozsvítit ho musíte vy [online]. *Technet - iDNES.cz*, 2015, [cit. 2019-04-03]. Dostupné z: [https://www.idnes.cz/technet/technika/mazaci-tramvaj-ovladej-stromecek-cvut-zemanek.A151208\\_084124\\_tec\\_technika\\_kuz](https://www.idnes.cz/technet/technika/mazaci-tramvaj-ovladej-stromecek-cvut-zemanek.A151208_084124_tec_technika_kuz)
- [4] FEL ČVUT: Fakulta elektrotechnická ČVUT v Praze představila dvě interaktivní aplikace na svítící fasádě Linky [online]. *FEL aktuality*, 2019, [cit. 2019-04-03]. Dostupné z: <https://www.fel.cvut.cz/cz/aktuality/2016/linky>
- [5] ČVUT FEL IoT Workshop: ČVUT FEL - Linky [online]. 2019, [cit. 2019-04-21]. Dostupné z: <https://linky.fel.cvut.cz>
- [6] MongoDB: The most popular database for modern apps [online]. [cit. 2019-04-03]. Dostupné z: <https://www.mongodb.com>
- [7] ZEIT Inc.: Showcase | Next.js [online]. 2019, [cit. 2019-04-26]. Dostupné z: <https://nextjs.org/showcase>
- [8] Node.js Foundation: Express [online]. 2019, [cit. 2019-04-03]. Dostupné z: <https://expressjs.com>

- [9] GitHub, Inc.: Topic: nodejs [online]. 2019, [cit. 2019-05-01]. Dostupné z: <https://github.com/topics/nodejs?o=desc&s=stars>
- [10] Breezy s.r.o.: Jak připravit web na GDPR [online]. *Solidpixels Academy*, 2018, [cit. 2019-04-24]. Dostupné z: <https://academy.solidpixels.cz/blog/jak-pripravit-web-na-gdpr>
- [11] VOLLMER, N.: Článek 13 EU obecné nařízení o ochraně osobních údajů [online]. *Privazy Plan*, 2018, [cit. 2019-04-24]. Dostupné z: <http://www.privacy-regulation.eu/cs/13.htm>
- [12] PLÍŠEK L. a spol.: BI-SP1: Projekt Linky [nepublikováno]. [cit. 2019-05-08].
- [13] PLÍŠEK, L.: Linky - Core [online]. [cit. 2019-05-13]. Dostupné z: <https://ctulinkycore.docs.apiary.io/>
- [14] OWASP Foundation: Cross-site Scripting [online]. *Open Web Application Security Project*, 2018, [cit. 2019-05-08]. Dostupné z: [https://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [15] OWASP Foundation: SQL Injection [online]. *Open Web Application Security Project*, 2016, [cit. 2019-05-08]. Dostupné z: [https://www.owasp.org/index.php/SQL\\_Injection](https://www.owasp.org/index.php/SQL_Injection)
- [16] MÁČA, J.: Úvod do Node.js [online]. *itnetwork*, 2018, [cit. 2019-04-07]. Dostupné z: <https://www.itnetwork.cz/javascript/nodejs/uvod-do-nodejs>
- [17] React - A JavaScript library for building user interfaces [online]. [cit. 2019-04-03]. Dostupné z: <https://reactjs.org>
- [18] ABRAMOV, D.: React as a UI Runtime [online]. 2019, [cit. 2019-04-26]. Dostupné z: <https://overreacted.io/react-as-a-ui-runtime>
- [19] DRIESSEN, V.: A successful Git branching model [online]. *nvie*, 2010, [cit. 2019-04-07]. Dostupné z: <https://nvie.com/posts/a-successful-git-branching-model>
- [20] FOWLER, M.: Continuous Integration [online]. [cit. 2019-02-16]. Dostupné z: <https://www.martinfowler.com/articles/continuousIntegration.html>
- [21] GitLab: GitLab CI/CD Pipeline Configuration Reference [online]. 2019, [cit. 2019-05-12]. Dostupné z: <https://docs.gitlab.com/ee/ci/yaml/>
- [22] Facebook Inc.: Testing React Apps [online]. *Jest Documentation*, 2019, [cit. 2019-04-24]. Dostupné z: <https://jestjs.io/docs/en/tutorial-react>

- [23] Cypress.io: JavaScript End to End Testing Framework | Cypress.io [online]. 2019, [cit. 2019-03-03]. Dostupné z: <https://www.cypress.io>
- [24] SVOBODA, V.: Nielsen's Heuristic Evaluation [online]. *Vojta Svoboda Blog*, 2011, [cit. 2019-04-19]. Dostupné z: <http://blog.vojtasvoboda.cz/nielsens-heuristic-evaluation>
- [25] NIELSEN, J.: 10 Usability Heuristics for User Interface Design [online]. *Nielsen Norman Group*, 1994, [cit. 2019-04-21]. Dostupné z: <https://www.nngroup.com/articles/ten-usability-heuristics>
- [26] ŠTOURACOVÁ, M.: Heuristická analýza v kontextu HCI se zaměřením na webové stránky akademické knihovny [online]. Bakalářská práce, 2013, [cit. 2019-05-03]. Dostupné z: [https://is.muni.cz/th/mdqia/Stouracova\\_BP.pdf](https://is.muni.cz/th/mdqia/Stouracova_BP.pdf)



## Seznam použitých zkratk

**API** Application Programming Interface

**BREAD** Browse, Read, Edit, Add, Delete

**CI** Continuous Integration

**CSS** Cascading Style Sheets

**DOM** Document Object Model

**GDPR** General Data Protection Regulation

**HTTPS** Hypertext Transfer Protocol Secure

**JSON** JavaScript Object Notation

**JSX** JavaScript XML

**NPM** Node Package Manager

**REST** Representational State Transfer

**SQL** Structured Query Language

**UI** User Interface

**XML** Extensible Markup Language

**XSS** Cross-site scripting

**YAML** YAML Ain't Markup Language





---

## Obsah přiložené SD karty

Součástí přílohy této práce je paměťová karta typu microSDHC. Samotná karta je vložena do adaptéru umístěného na zadních deskách této práce. Níže na této stránce je uveden obsah SD karty.

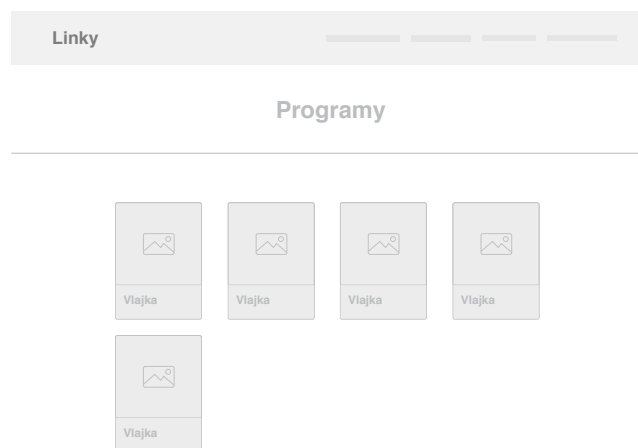
README.md	.....	stručný popis obsahu SD karty
doc		
├── index.html	.....	offline verze dokumentace API
src		
├── URL.md	.....	odkaz do Git repozitáře ke zdrojovým souborům
├── code	.....	obsah Git repozitáře
├── thesis	.....	zdrojová forma práce ve formátu $\LaTeX$
└── thesis.pdf	.....	text práce ve formátu PDF



---

# Wireframy

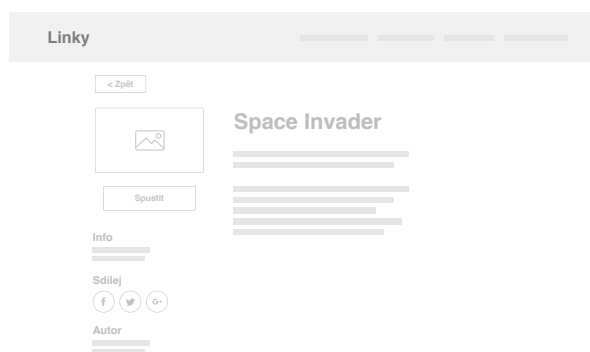
## Program



Obrázek C.1: Wireframe seznamu programů

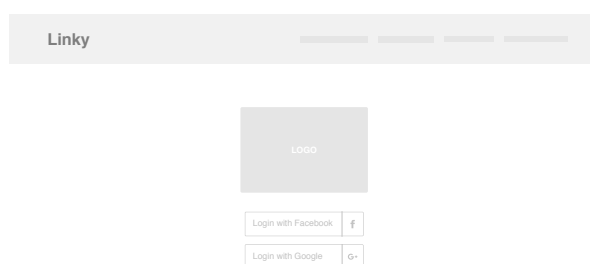
## C. WIREFRAMY

---



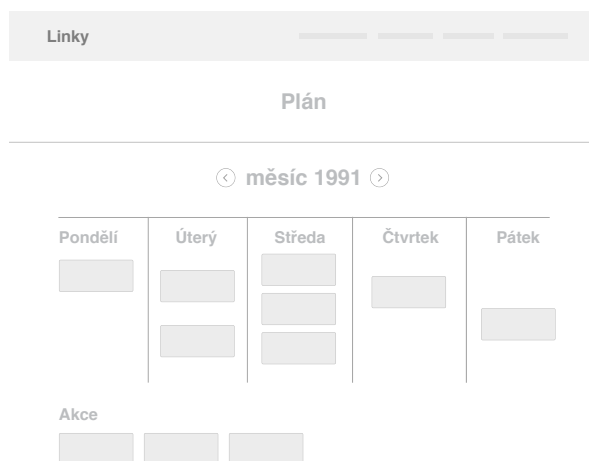
Obrázek C.2: Wireframe detailu programu

## Přihlášení



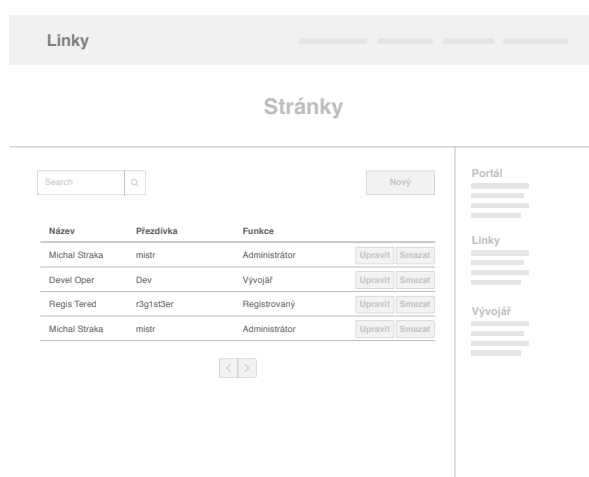
Obrázek C.3: Wireframe přihlašovací stránky

## Plán

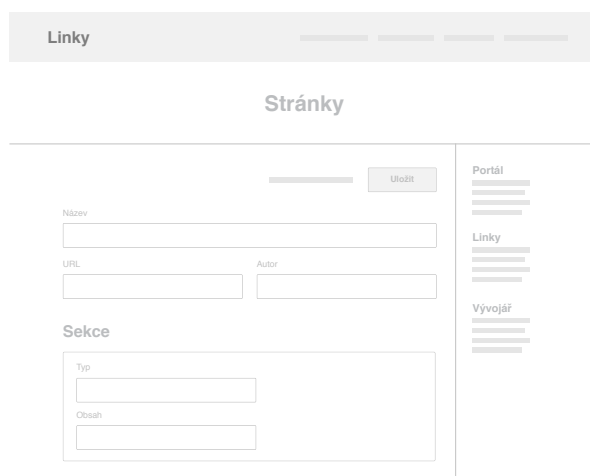


Obrázek C.4: Wireframe plánu akcí

## Správa záznamů

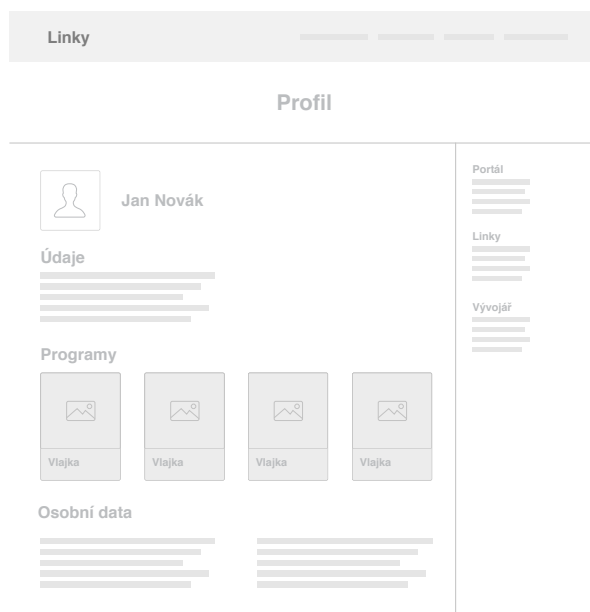


Obrázek C.5: Wireframe seznamu stránek



Obrázek C.6: Wireframe formuláře pro úpravu statické stránky

## Uživatelský profil



Obrázek C.7: Wireframe uživatelského profilu

---

## Nahrání programu

The wireframe shows a page with a header 'Linky' and a main title 'Nahrát nový program'. The form is divided into two columns. The left column contains a 'Nahrát nový program' section with input fields for 'Název', 'Kód', and 'URL', a dashed box for file uploads labeled 'Drag Files Here' with a plus sign, and an 'Odeslat' button. The right column contains a sidebar with three sections: 'Portál', 'Linky', and 'Vyvořář', each with three horizontal lines representing content.

Obrázek C.8: Wireframe formuláře pro nahrání programu

## Nastavení

The wireframe shows a page with a header 'Linky' and a main title 'Nastavení'. The form is divided into two columns. The left column contains an 'Instalace' section with a placeholder image and a slider for 'Jaz', and a 'Portál' section with input fields for 'Hlavní stránka' and 'Facebook ID'. The right column contains a sidebar with three sections: 'Portál', 'Linky', and 'Vyvořář', each with three horizontal lines representing content.

Obrázek C.9: Wireframe formuláře pro nastavení instalace