



## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

**Název:** Historie VUT (iOS mobilní aplikace)  
**Student:** Vojtěch Pájer  
**Vedoucí:** Ing. Martin Pípl  
**Studijní program:** Informatika  
**Studijní obor:** Softwarové inženýrství  
**Katedra:** Katedra softwarového inženýrství  
**Platnost zadání:** Do konce zimního semestru 2016/17

### Pokyny pro vypracování

Cílem práce je navrhnout a implementovat mobilní aplikaci pro operační systém iOS. Aplikace bude navržena a implementována jak pro mobilní telefony, tak pro tablety. Aplikace zmapuje historii VUT. Důležité momenty, významné v dějové osobnosti, vznik fakult, chronologii rektorů. Aplikace bude mít českou a anglickou lokalizaci. Proveďte analýzu, prozkoumejte konkurenční řešení, navrhnete wireframy. Obzvláště si nechte záležet na návrhu uživatelského rozhraní. Prozkoumejte aplikace, které získaly ocenění na Apple design awards. Inspiraci můžete hledat například z aplikace Boeing Milestones in Innovation nebo Yahoo News Digest. Aplikace bude využívat webový back-end a komunikovat přes aplikační rozhraní (API), které navrhne další diplomant. Pokud by se spolupráce nedařila, vložte data přímo do aplikace. Na historickém obsahu spolupracujte s diplomantem, který bude řešit verzi aplikace pro OS Android.

### Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

prof. Ing. Pavel Tvrdlík, CSc.  
ředitel katedry

V Praze dne 25. února 2015



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



Bakalářská práce

## **Historie ČVUT - mobilní iOS aplikace**

*Vojtěch Pajer*

Vedoucí práce: Ing. Martin Půlpitel

16. května 2016



---

## Poděkování

V první řadě bych rád poděkoval Ing. Martinu Půlpitlovi za vedení této bakalářské práce.

Dále děkuji Bc. Tomáši Krabačovi, za metodické vedení v rámci této práce a cenné rady během její tvorby.

Nakonec bych chtěl poděkovat své rodině a přátelům za podporu po celou dobu studia.



---

## 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 16. května 2016

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2016 Vojtěch Pajer. 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**

Pajer, Vojtěch. *Historie ČVUT - mobilní iOS aplikace*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.



---

## Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací mobilní aplikace pro operační systém iOS prezentující informace o univerzitě České vysoké učení technické v Praze. Součástí práce je sběr dat o univerzitě, návrh designu aplikace a spolupráce s dvěma diplomanty, kteří v rámci svých diplomových prací vytvářeli webový backend a verzi této aplikace pro operační systém Android.

**Klíčová slova** Mobilní aplikace, iOS, ČVUT, historie, UX design, Swift

---

## Abstract

This bachelor thesis deals with design and implementation of mobile application for operating system iOS that displays information about Czech Technical University in Prague. The work includes collecting of data about university, creating a design draft and cooperation with two master students who have been creating backend part and an Android version of this application.

**Keywords** Mobile application, iOS, ČVUT, history, UX design, Swift



---

# Obsah

<b>Úvod</b>	<b>1</b>
Motivace . . . . .	1
Struktura práce . . . . .	2
<b>1 Specifikace zadání a cíl projektu</b>	<b>3</b>
1.1 Cíl projektu . . . . .	3
1.2 Nefunkční požadavky . . . . .	4
1.3 Nezrealizované funkcionality . . . . .	5
<b>2 iOS</b>	<b>7</b>
2.1 Architektura . . . . .	7
2.2 XCode . . . . .	9
2.3 UI komponenty . . . . .	10
2.4 Controllery . . . . .	11
2.5 Gesta . . . . .	11
<b>3 Analýza a sběr dat</b>	<b>13</b>
3.1 Sběr dat . . . . .	13
3.2 Příklad . . . . .	14
3.3 Inspirace . . . . .	14
<b>4 Návrh</b>	<b>19</b>
4.1 Wireframy . . . . .	19
4.2 Architektura . . . . .	23
<b>5 Realizace</b>	<b>25</b>
5.1 Způsob implementace . . . . .	25
5.2 Boeing efekt . . . . .	26
5.3 Page View Controller a Collection View . . . . .	29
5.4 Použité knihovny . . . . .	30

<b>6 Testování</b>	<b>33</b>
6.1 Uživatelské testy . . . . .	33
<b>Závěr</b>	<b>35</b>
Rozšiřitelnost . . . . .	35
<b>Literatura</b>	<b>37</b>
<b>A Seznam použitých zkratk</b>	<b>41</b>
<b>B Obsah příloženého CD</b>	<b>43</b>

---

## Seznam obrázků

3.1	Hlavní menu aplikace Boeing Milestones . . . . .	15
3.2	Detail článku aplikace Yahoo News Digest . . . . .	16
3.3	Virtuální prohlídka aplikace Mississippi State University . . . . .	17
3.4	Aplikace Timeline - Art Museum . . . . .	18
4.1	Návrh kategorie Historie ČVUT . . . . .	20
4.2	Návrh rozcestníku osobností . . . . .	21
4.3	Návrh kategorie Osobnosti . . . . .	21
4.4	Návrh kategorie Přehled . . . . .	22
4.5	Návrh hlavního menu . . . . .	23
5.1	Použití Boeing efektu . . . . .	26
5.2	Rozevírání buňky detailu . . . . .	29
5.3	Detail historické události . . . . .	29
5.4	Obrazovka rektorů . . . . .	30
6.1	Rozdělení testerů podle zkušenosti s iOS . . . . .	33



---

## Seznam tabulek

3.1	Rozdělení práce při sběru dat . . . . .	13
3.2	Rozdělení překladů dat . . . . .	14





---

# Úvod

Mobilní aplikace patří k nejmladším odvětvím v informatice a v dnešní době zažívají stále větší růst na popularitě. Jsou vytvářeny pro operační systémy iOS, Android, Windows Phone a další. Jejich uživatelé mají odlišnou představu o tom, jak by měla aplikace vypadat, většinou však vyžadují intuitivní a zajímavé uživatelské rozhraní.

České vysoké učení technické (dále jen ČVUT) je nejstarší nevojenská technická univerzita v Evropě, jejíž kořeny sahají až do počátku 18. století. Během více než čtyř století se v jejím čele vystříдалo nespočet rektorů a navštívilo ji mnoho významných a zajímavých osobností. Na půdě ČVUT se od jeho založení odehrálo velké množství zásadní událostí, které mimo jiné zapříčinily vznik osmi rozsáhlých fakult.

Tématem této bakalářské práce je dohledat a sjednotit historické informace, týkající se ČVUT a vytvořit mobilní aplikaci pro operační systém iOS, která je bude zobrazovat. Důležitým úkolem bude navrhnout originální a intuitivní uživatelské rozhraní, které zaujme a udrží uživatelovu pozornost.

## Motivace

Důvodem výběru tohoto tématu byla především motivace naučit se vyvíjet mobilní iOS aplikace. Vývoj aplikace s nestandardním uživatelským rozhraním, na které si tato aplikace kladla důraz, se jevil jako velmi dobrým prostředkem k pochopení programovacího jazyka, neboť bylo zřejmé, že pro její tvorbu nebude stačit pochopení základů jazyka, ale naopak bude zapotřebí do něj proniknout hlouběji.

## Struktura práce

Bakalářská práce je rozdělena do osmi kapitol:

1. *Úvod*: Důvod vzniku tohoto projektu a odhalení motivace na jeho výběr a vypracování.
2. *Specifikace zadání a cíl projektu*: Bližší specifikace požadavků na vypracování projektu a objasnění jeho celkového přínosu. Určení cíle projektu.
3. *iOS*: Uvedení čtenáře do základů týkajících se operačního systému iOS.
4. *Analýza a sběr dat*: Prozkoumání aplikací, kterými by bylo možné se inspirovat. Dále také popis průběhu shromažďování dat, které jsou aplikací prezentovány.
5. *Návrh*: Návrh uživatelského rozhraní aplikace. Popisuje funkce aplikace a pomocí wireframů přibližuje návrh každé z obrazovek aplikace.
6. *Realizace*: Ukazuje některé použité knihovny a popisuje problematické části implementace aplikace.
7. *Testování*: Testování aplikace.
8. *Závěr*: Shrnutí bakalářské práce, zhodnocení dosaženého řešení, možnosti rozšíření.

---

# Specifikace zadání a cíl projektu

Projekt tvorby aplikace Historie ČVUT byl rozdělen do tří zadání.

Prvním zadáním bylo navrhnout a implementovat mobilní aplikaci pro operační systém iOS, sesbírat data, která bude prezentovat a prozkoumat aplikace se zajímavým designem, kterými by se návrh aplikace mohl inspirovat. Toto zadání bylo vypracováno autorem této práce.

Druhé zadání bylo vypsáno jako diplomová práce a bylo stejné s prvním zadáním s jediným rozdílem - cílovým operačním systémem byl Android. Toto zadání si zvolil Bc. Tomáš Krabač, se kterým autor této práce spolupracoval na sběru dat a návrhu designu aplikace.

Třetí zadání bylo také vypsáno jako diplomová práce a mělo za úkol vytvořit webový backend, ze kterého budou výše zmíněné aplikace získávat data. Toto zadání vypracoval Ing. Vojtěch Bartoš.

Na projektu se podíleli i další lidé. Prof. PhDr. Marcela Efmertová, CSc. provedla korekturu historických informací o ČVUT, Ing. Roman Gordienko vytvořil na základě wireframe<sup>1</sup> grafiku aplikace a Bc. Tomáš Hána pomáhal s překladem historických textů. Společnost Ackee s.r.o [1] poskytla technické zázemí a konzultace po celou dobu projektu.

## 1.1 Cíl projektu

Cílem práce je vytvoření funkční aplikace, která poutavou formou prezentuje informace o ČVUT. Mezi tyto informace patří chronologie rektorů, důležité historické události, vznik fakult a významné vědecké osobnosti univerzity. Požadavky na aplikaci byly rozděleny na funkční požadavky a nefunkční požadavky.

---

<sup>1</sup>Wireframe je návrh obrazovky, znázorňující rozložení prvků a použité funkce.

### 1.1.1 Funkční požadavky

Funční požadavky specifikují, co by aplikace měla dělat. Následující seznam funkční požadavků uvádí ty, které jsou nutné pro úspěšné splnění zadání.

- Aplikace zobrazuje historické události ČVUT
- Aplikace zobrazuje rektory ČVUT
- Aplikace zobrazuje významné vědecké osobnosti ČVUT
- Aplikace zobrazuje vznik fakult ČVUT

Během analýzy vznikl také seznam funkčních požadavků, které nejsou pro splnění zadání nutné, ale hodí se k charakteru aplikace a byly do ní zakomponovány.

- Aplikace zobrazuje úspěšné sportovce ČVUT
- Aplikace zobrazuje základní informace o ČVUT
- Aplikace zobrazuje základní informace o fakultách ČVUT

## 1.2 Nefunkční požadavky

Nefunkční požadavky určují, jak by se aplikace měla chovat. Narozdíl od funkčních požadavků budou jednotlivé nefunkční požadavky blíže popsány.

### 1.2.1 Aplikace funguje na mobilních telefonech a tabletech

Na základě zvyklostí uživatelů, aplikace fungující jak na mobilních telefonech tak na tabletech mají často odlišný vzhled uživatelského rozhraní na obou z těchto zařízení. Tablety disponují větším displejem než mobilní telefony a umožňují tak efektivnější využití místa na obrazovce, vyšší kvalitu obrázků, apod. Je tedy nutné vytvořit odlišný návrh pro mobilní telefony a pro tablety.

### 1.2.2 Aplikace je dostupná v češtině a angličtině

Vytvoření aplikace podporující pouze češtinu by výrazně zúžilo cílovou skupinu uživatelů. Webové API bylo tedy navrženo tak, aby poskytovalo data jak v češtině, tak v angličtině. Aplikace rozhodne, která z těchto dat zobrazí na základě nastavení jazyka zařízení.

### 1.2.3 Aplikace stahuje data z webového API

Vytvoření webového API<sup>2</sup> měl na starost Ing. Vojtěch Bartoš. Dokumentace k API je dostupná na [2].

Stahování dat do aplikace z webového API je v dnešní době běžné a přináší s sebou řadu výhod. Přidávání dat do aplikace lze provést aktualizováním API a není nutné zasahovat do kódu aplikace. Snižuje se také celková velikost aplikace, neboť veškerá její data jsou uložena na internetu.

Hlavní nevýhodou používání webového API je nutnost internetového připojení při používání aplikace.

## 1.3 Nezrealizované funkcionality

V průběhu analýzy bylo zvaženo hodně prvků, které by mohly být součástí aplikace. Některé z nich byly zamítnuty kvůli nevhodnosti. V následujících odstavcích jsou tyto prvky popsány.

### 1.3.1 Virtuální prohlídka

Zakomponování virtuální prohlídky do aplikace byl nápad, který byl zvažován nejdříve. Byly vytvořeny i návrhy obrazovek s virtuální prohlídkou, ale nakonec byl návrh zamítnut kvůli nedostatku času.

### 1.3.2 Aktuality

Informace o aktuálním děním na ČVUT byly zamýšleny jako úvodní obrazovka celé aplikace. Bylo však odsouhlaseno, že aktuality neodpovídají charakteru celé aplikace.

### 1.3.3 Podpora více jazyků

Aplikace podporuje angličtinu a češtinu. Přidáním dalších jazyků by se ještě více rozšířila její cílová skupina.

---

<sup>2</sup>Webové API (Web Application Programming Interface) je rozhraní pro komunikaci s webovými aplikacemi.



---

# iOS

V této kapitole bude popsán operační systém iOS, jeho architektura a budou vysvětleny základní principy a postupy při vývoji iOS aplikací.

iOS je mobilní operační systém UNIXového typu vytvořený společností Apple, který je nyní používán v zařízeních iPad<sup>3</sup>, iPhone<sup>4</sup>, iPod Touch<sup>5</sup> a Apple TV<sup>6</sup>. Dle tvrzení společnosti Apple se jedná o nejpokročilejší operační systém na světě [3].

Aplikace pro iOS je možné psát ve dvou programovacích jazycích. Prvním z nich je jazyk Objective-C. Jedná se o jazyk, který je nadstavbou jazyka C, doplněnou o možnost objektově orientovaného programování a dynamického runtime<sup>7</sup>. Objective-C tedy rozšiřuje syntaxi jazyka C možností použití tříd a metod [4]. Je v něm zároveň implementován koncept zasílání zpráv podobný tomu, který využívá jazyk Smalltalk [5].

Druhým jazykem je Swift. Swift je nový programovací jazyk vytvořený společností Apple v roce 2014. Je zamýšlen jako náhrada za jazyky postavené na základě jazyka C. Hlavním důvodem vytvoření Swiftu byla myšlenka jazyka, jehož rozsah využití bude sahát od systémového programování přes programování mobilních a desktopových aplikací až po cloudové služby [6]. Je navržen tak, aby vývojářům ulehčil vytváření a zpravování kódu a zabránil tvorbě zbytečných chyb při vývoji.

## 2.1 Architektura

Operační systém je prostředník mezi hardwarem a aplikacemi, které jsou na zařízení spuštěny. Architektura operačního systému iOS je čtyřvrstvá [7]. V následujících odstavcích jsou vrstvy popsány.

---

<sup>3</sup>iPad (<http://www.apple.com/ipad/>)

<sup>4</sup>iPhone (<http://www.apple.com/iphone/>)

<sup>5</sup>iPod Touch (<http://www.apple.com/ipod-touch/>)

<sup>6</sup>Apple TV (<http://www.apple.com/tv/>)

<sup>7</sup>Dynamický runtime je schopnost jazyka rozhodovat o typech objektů za běhu programu.

### 2.1.1 Vrstva Cocoa Touch

Vrstva **Cocoa Touch** obsahuje nejdůležitější frameworky<sup>8</sup> pro tvorbu iOS aplikací. Pomocí těchto frameworků je vytvářen finální vzhled aplikace. Poskytují také množství důležitých technologií, bez kterých by tvorba aplikací nebyla možná [8]. Patří mezi ně například:

- *Gesture recognition* - Zaznamenávání interakce uživatele s dotykovou obrazovkou zařízení.
- *Multitasking* - Umožnění aktivity aplikací i v okamžiku, kdy nejsou používány a jsou spuštěny na pozadí.
- *MapKit Framework* - Snadná práce s mapami.

### 2.1.2 Vrstva Media

Vrstva **Media** obsahuje grafické, zvukové a video technologie, které jsou používány k tvorbě multimediálních prvků aplikace [9]. K těm nejzajímavějším patří:

- *Core Animation* - Možnost tvorby animací, například přesunů prvků a změn tvarů objektů.
- *Media Player Framework* - Umožnění snadného přístupu k písním a seznamům písní v knihovně iTunes.
- *UIImagePickerController* - Možnost vyfocení fotografie, resp. nahrání videa a okamžité použití v aplikaci.

### 2.1.3 Vrstva Core Services

Vrstva **Core Services** obsahuje základní systémové služby pro aplikace. Nejdůležitější z těchto služeb jsou frameworky **Core Foundation** a **Foundation**, které definují základní datové typy používané ve všech aplikacích [10]. Z těchto dalších nutno zmínit:

- *Grand Central Dispatch* - Práce s vlákny.
- *In-App Purchases* - Možnost nákupu specifických výhod, vlastností nebo bonusů přímo z aplikace.
- *Ad Support Framework* - Podpora reklam.
- *Core Data Framework* - Poskytnutí databáze, do které mohou aplikace ukládat používaná data.

---

<sup>8</sup>Framework je softwarová struktura, která obsahuje knihovny, základní komponenty a další prvky podporující vývoj.



### 2.1.4 Vrstva Core OS

Vrstva Core OS obsahuje prvky, na kterých je postavena většina ostatních technologií [11]. Za zmínku stojí například:

- *External Accessory Framework* - Komunikace s externími hardwarovým doplňky.
- *Security Framework* - Možnost zvýšit zabezpečení dat v aplikaci nad rámec základních poskytovaných bezpečnostních funkcí.
- *Local Authentication Framework* - Použití Touch ID<sup>9</sup>.

## 2.2 XCode

XCode [12] je vývojové prostředí společnosti Apple, které obsahuje nástroje k vývoji aplikací pro operační systémy iOS, OS X<sup>10</sup>, tvOS<sup>11</sup> a watchOS<sup>12</sup>. Mezi podporované jazyky patří C, C++, Objective-C, Java, Python, Swift a mnoho dalších. XCode umožňuje správu celého vývoje - vytvoření aplikace, testování, optimalizaci a nahrání na App Store<sup>13</sup>.

V následující části budou popsány nástroje, které byly během vývoje aplikace Historie ČVUT využity.

### 2.2.1 iOS Simulator

iOS Simulator [13] je nástroj, který je nainstalována spolu s XCode softwarem. Nástroj umožňuje vývojářům testovat běh aplikace během vývoje simulováním iPhone, iPad, Apple Watch<sup>14</sup> nebo Apple TV prostředí. Jedná se o vynikající nástroj, který výrazně urychluje proces testování. Jednou z velkých výhod je možnost otestovat aplikaci na zařízeních, které vývojář nemá k dispozici. Lze tak objevit chyby v rozložení prvků, případně použití funkcí, které nejsou daným zařízením podporované apod.

### 2.2.2 Playgrounds

Playground [14] je interaktivní vývojové prostředí podporující jazyk Swift, které okamžitě zobrazuje hodnoty všech proměnných. Tento nástroj výrazně urychluje učení a chápání jazyka Swift, protože umožňuje experimentovat s jazykem a poskytuje okamžité výsledky.

<sup>9</sup>Touch ID je identifikace uživatele pomocí otisku prstu.

<sup>10</sup>OS X je operační systém pro počítače společnosti Apple.

<sup>11</sup>tvOS je operační systém pro televize společnosti Apple.

<sup>12</sup>watchOS je operační systém pro hodinky společnosti Apple.

<sup>13</sup>App Store je obchod s aplikacemi společnosti Apple.

<sup>14</sup>Apple Watch (<http://www.apple.com/watch/>)

### 2.2.3 Interface Builder

**Interface Builder** [15] je velmi mocný nástroj, který umožňuje vytvoření kompletního uživatelského rozhraní. Nástroj funguje na principu drag and drop<sup>15</sup> a zpřístupňuje přidávání oken, obrázků, tlačítek a mnoho dalších objektů. Pro jednotlivé komponenty uživatelského rozhraní se pak v dalším kroku vytvoří funkce, které zpracovávají interakce s nimi, například kliknutí na tlačítko, nebo provedení gesta.

## 2.3 UI komponenty

UI komponenty (dále jen **komponenty**) jsou základní ovládací prvky, které uživatel vidí, a které používá. Všechny **komponenty** jsou obsažené v **Cocoa Touch** frameworku zvaném **UIKit**. V následující sekci budou popsány pouze některé z těch, které byly při vývoji aplikace **Historie ČVUT** použity.

### 2.3.1 View

**View** je základní komponenta, od které dědí všechny ostatní komponenty. Obsahuje základní sadu metod pro nastavení tvaru, rozměrů, pozadí, apod.

### 2.3.2 Table View

**Table View** je jednosloupcová tabulka, kterou lze procházet gestem **swipe** (viz 2.5.2).

### 2.3.3 Collection View

**Collection View** je nadstavba tabulky a umožňuje definovat vlastní **Layout**<sup>16</sup> položek. Je tedy možné vytvořit tabulku s libovolným počtem sloupců a řádků.

### 2.3.4 Text View

**Text View** je komponenta pro zobrazování většího množství textu. Textem se dá procházet a je možné ho editovat.

### 2.3.5 Image View

**Image View** je komponenta pro zobrazování obrázků.

---

<sup>15</sup>Drag and drop (neboli *táhnout a pustit*) v tomto případě znamená táhnout prvek na požadované místo v obrazovce a pustit ho tam.

<sup>16</sup>Layout určuje rozložení položek na obrazovce. Lze definovat jejich velikost, tvar, story mezi nimi, apod.

### 2.3.6 Button

Button neboli tlačítko je komponenta, která reaguje na kliknutí a spouští určitou akci.

## 2.4 Controllery

Controllery reprezentují jednotlivé obrazovky aplikace a jsou základem celé její struktury. Každá aplikace obsahuje minimálně jeden, většinou však více controllerů. Do controllerů se vkládají jednotlivé komponenty a tvoří tak celkový vzhled obrazovky. Controllery jsou pojmenované podle jejich počátečního obsahu - View Controller je prázdná obrazovka, Collection View Controller je vyplněn Collection View, atp.

Speciálním případem je Page View Controller, který simuluje chování otáčení stránek v knize. Stránky jsou reprezentovány obrazovkami, kterými lze pomocí gesta swipe procházet.

## 2.5 Gesta

Jakákoliv interakce uživatele s displejem zařízení se dá považovat za gesto. Některé komponenty mají definované reakce na určitá gesta, například tabulka reaguje na gesto swipe (viz 2.5.2) posunutím obsahu ve směru gesta.

Třídy starající se o gesta se nacházejí ve frameworku UIKit. Ten je schopný rozlišit sedm druhů gest předdefinovaných systémem a je možné v něm vytvářet gesta vlastní.

V této sekci budou vysvětlena pouze ta, která byla použita při vývoji aplikace Historie ČVUT.

### 2.5.1 Tap

Tap je jednoduchý dotek obrazovky, který slouží nejčastěji k potvrzení dialogů nebo výběru položky.

### 2.5.2 Swipe

Swipe gesta lze docílit přiložením jednoho prstu na obrazovku a přejetím s ním libovolným směrem. Používá se například k procházení tabulkou.

### 2.5.3 Pinch open

Pinch open se provádí pomocí dvou prstů. Spojené prsty se přiloží na libovolné místo obrazovky a roztáhnou se od sebe. Gesta se využívá například při přibližování v mapách nebo zvětšení obrázků.

### 2.5.4 Pinch close

`Pinch close` je přesný opak gesta `pinch open`. Provádí se přiložením dvou prstů a jejich následným přiblížením. Gesto se využívá například při oddalování v mapách nebo zmenšení obrázků.

---

# Analýza a sběr dat

## 3.1 Sběr dat

Přestože je aplikace postavena především na poutavém grafickém návrhu, bez dat, která jsou jí prezentována, nemá její existence smysl. Sběr dat byl tedy jednou z klíčových aktivit prováděných v rámci tohoto projektu. Práci si autor této práce rozdělil s Bc. Tomášem Krabačem tak, jak je to vyobrazeno v tabulce 3.1. Tato sekce se zmiňuje pouze o kategoriích, na kterých se podílel autor této práce.

	<b>Tomáš Krabač</b>	<b>Vojtěch Pajer</b>
Milníky ČVUT	x	
Přehled ČVUT	x	x
Přehled fakult ČVUT	x	x
Rektoři		x
Významné osobnosti	x	
Sportovci		x

Tabulka 3.1: Rozdělení práce při sběru dat

### 3.1.1 Přehled ČVUT

Informace o ČVUT byly získány z knihy [16] a z webu ČVUT [17].

### 3.1.2 Přehled fakult ČVUT

Informace o fakultách ČVUT byly také získány z knihy [16] a z webu [18]. Obrázky u jednotlivých fakult byly vytvořeny v rámci projektu *Sedm statečných* [19].

#### 3.1.3 Rektoři

Seznam rektorů ČVUT je již zpracován na webu ČVUT [20], odkud byla data převzána. Popis většiny rektorů obsahuje pouze informaci o tom, v jakém oboru byli profesory a je zde rozhodně prostor k budoucímu rozšíření.

#### 3.1.4 Sportovci

Hlavním zdrojem informací o úspěšných sportovcích ČVUT byla kniha *Sport na ČVUT kdysi a dnes* [21]. Kniha shrnuje dění na sportovní půdě ČVUT od začátku 20. století do současnosti.

### 3.2 Překlad

Dle nefunkčního požadavku 1.2.2 má být aplikace dostupná nejen v češtině, ale i v angličtině. Webové API bylo navrženo tak, aby bylo možné ke každé položce nahrávat informace v obou jazycích. Na překladu všech textů se podílel autor této práce a Bc. Tomáš Hána, a rozdělení jejich práce je znázorněné v tabulce 3.2.

	Tomáš Hána	Vojtěch Pajer
Milníky ČVUT	x	
Přehled ČVUT		x
Přehled fakult ČVUT		x
Rektoři		x
Významné osobnosti		x
Sportovci		x

Tabulka 3.2: Rozdělení překladů dat

### 3.3 Inspirace

Pro návrh uživatelského rozhraní bylo nutné prozkoumat již existující aplikace se zajímavým designem a inspirovat se jimi. Největší pozornost měly aplikace, které získaly ocenění *Apple Design Awards*<sup>17</sup>.

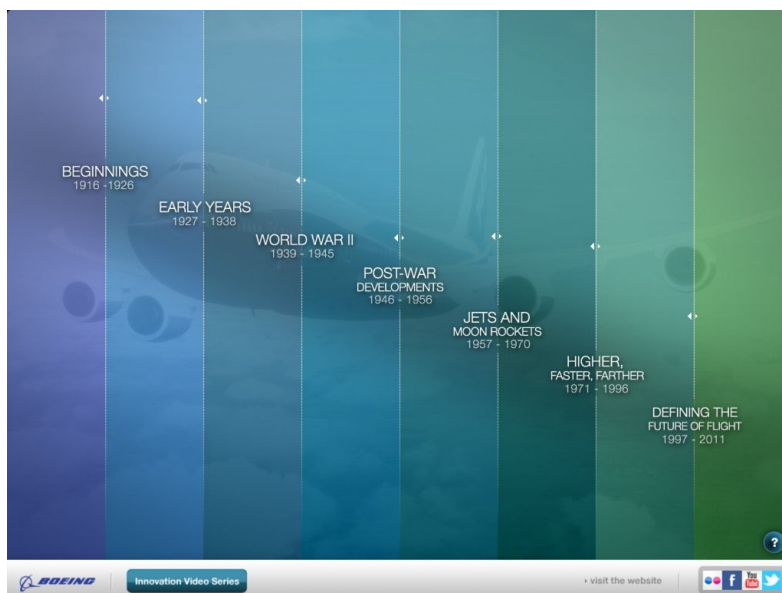
#### 3.3.1 Boeing Milestones

Aplikace *Boeing Milestones* je výborně popsána na její úvodní stránce na *App Storu* - "S ohromujícími archivními snímky a podmanivými videi, "Mi-

---

<sup>17</sup>Ocenění, které společnost Apple každým rokem uděluje aplikacím s nejzajímavějším designem

*stones in Innovation*"oživuje dlouhý řetězec průlomových inovací Boeingu od roku 1916 do současnosti"[22].



Obrázek 3.1: Hlavní menu aplikace Boeing Milestones

Aplikace je celá postavena na efektu, který je později nazýván Boeing efekt. Důležité události v historii Boeingu jsou rozděleny do jednotlivých období, která jsou prezentována jako horizontálně se posouvající seznam položek. Přístupu k důležitým rokům daného období lze docílit provedením gesta *pinch open* (viz. podsekcce 2.5.3). Po provedení tohoto gesta se obrazovka v dané položce rozdělí na dvě části, horizontálně se rozevře a zobrazí důležité roky z tohoto období. Roky jsou prezentovány stejným způsobem, jako období a zobrazení jejich detailu lze docílit stejně.

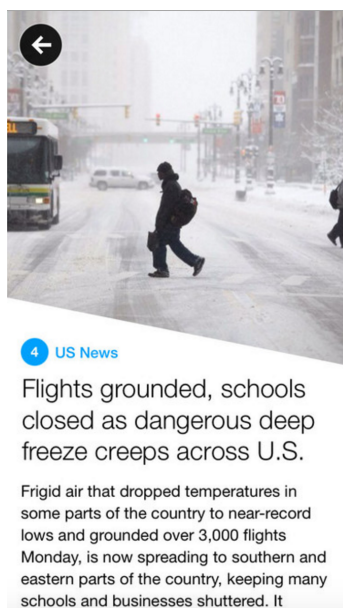
Téměř všudypřítomné tlačítko *zpět* v levém horním rohu obrazovky je v případě této aplikace nahrazeno gestem *pinch close* (viz. podsekcce 2.5.4), které detail elegantně uzavře a vrátí uživatele na seznam roků.

Nápady použité v této aplikaci jsou velmi zajímavé a posloužily jako hlavní inspirace pro návrh aplikace *Historie ČVUT*. Přesto lze v aplikaci nalézt několik nedostatků.

Při zobrazení detailu libovolného roku je možné pomocí gesta *swipe* (viz. podsekcce 2.5.2) přecházet na roky následující. Posune-li se uživatel na následující rok a z tohoto roku se vrátí zpět na časovou osu, časová osa se uzavře na roku, na kterém byla otevřena, což neodpovídá roku, ze kterého se na ní uživatel vrací.

#### 3.3.2 Yahoo News Digest

Aplikace **Yahoo News Digest** [23] poskytuje souhrn těch nejdůležitějších novinek z dění ve světě. V roce 2014 získala za svůj design ocenění **Apple Design Awards**.



Obrázek 3.2: Detail článku aplikace Yahoo News Digest

Obrazovky jednotlivých článků mají v horní části obrázek vazující se k dané události a ve spodní části text k této události. Aplikace disponuje velmi použitavou funkcionalitou, tzv. **Parallax scrolling efektem**.

**Parallax scrolling efekt** je grafický efekt, při kterém se objekty na pozadí obrazovky pohybují pomaleji než objekty v jejím popředí, čímž vytváří iluzi hloubky obrazu. Tento efekt byl a stále je velmi často používán ve 2D počítačových hrách a v dnešní době se stal velmi populární ve vývoji webových stránek [24].

Při vysouvání části s textem z dolní části obrazovky se obrázek v její horní části pohybuje pomaleji, než vysouvaná část. Ve výsledku se zdá, že část s textem překrývá obrázek.

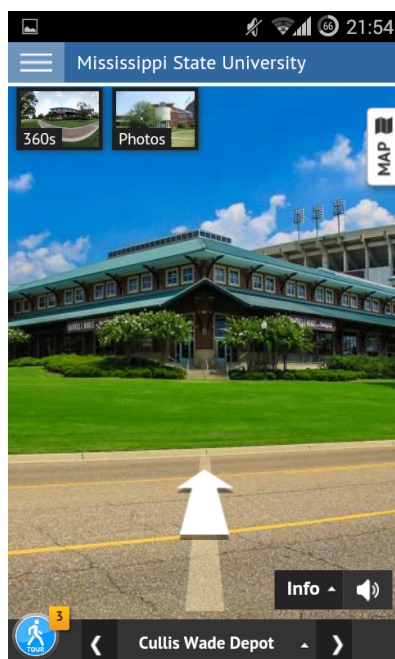
Aplikace působí velmi elegantně a intuitivně a funkcionalita **Parallax scrolling efektu** byla zakomponována do návrhu aplikace **Historie ČVUT**.

#### 3.3.3 Mississippi State University

**Mississippi State University** [25] je aplikace, která provádí uživatele virtuální prohlídkou po areálu kampusu univerzity. Místa virtuální prohlídka jsou tvořena fotografiemi v určitém pořadí. Některá místa obsahují textový popis a



fotky, u jiných je možnost zobrazení 360° prostoru. Přítomná je také možnost zobrazení všech bodů prohlídky a posunutí se na libovolný z nich.



Obrázek 3.3: Virtuální prohlídka aplikace Mississippi State University

Aplikace inspirovala virtuální prohlídku v aplikaci *Historie ČVUT*. Jednoduchost prohlídky vedené po jedné, předem určené trase, byla velkým pozitivem. Zabránila tak pohybu v kruzích a zajistila, že všechna místa prohlídky budou navštívena.

#### 3.3.4 Timeline - Art Museum

Poslední aplikací, která stojí za zmínku je *Timeline - Art Museum* [26]. Tato aplikace zachycuje obrazy známých malířů a informace o jejich životě. Každý obraz obsahuje informaci o tom, ve které galerii ho lze nalézt a jakou technikou byl namalován. Obrazovka malířů má velmi zajímavý návrh (viz obrázek 3.4). Ve spodní části obrazovky se nachází navigační lišta, zobrazující horizontálně posuvný seznam všech malířů. S navigační lištou se dá manipulovat pomocí gesta *swipe* (viz. podsekcce 2.5.2). Při vybrání některého z umělců se zobrazí jeho detail.

Aplikace je dostupná pouze na OS Android.

### 3. ANALÝZA A SBĚR DAT

---



Obrázek 3.4: Aplikace Timeline - Art Museum

---

# Návrh

V této kapitole je popsán průběh návrhu uživatelského rozhraní, podle kterého jsou později implementovány jednotlivé obrazovky aplikace. Na konci kapitoly je popsána zvolená architektura aplikace.

## 4.1 Wireframy

Wireframe je návrh obrazovky, znázorňující rozložení prvků a použité funkce. Po prozkoumání dostatečného množství aplikací, kterými bylo možné se inspirovat byly ve spolupráci s Bc. Tomášem Krabačem navrženy wireframy pro všechny obrazovky aplikace.

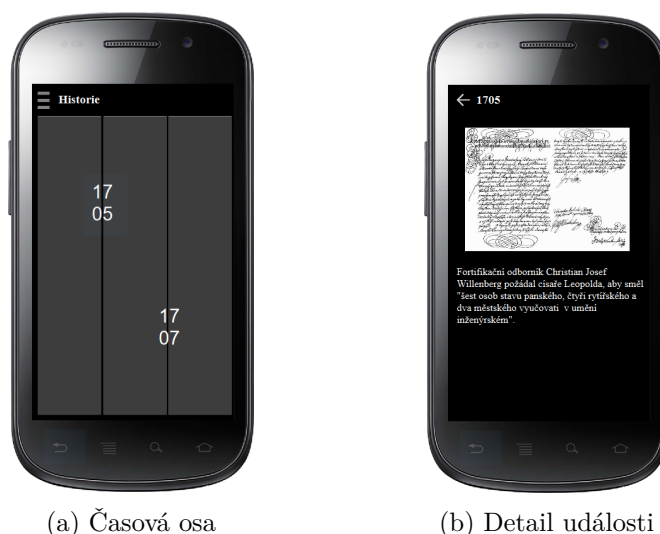
Wireframy byly vytvořeny v aplikaci *Pencil*, která je volně dostupná na [27] a obsahuje šablony pro mobilní aplikace. Pro jednotnost návrhu byly všechny wireframy navrženy pro mobilní telefony s operačním systémem Android.

Navržené wireframy byly předány grafikovi firmy Ackee s.r.o Ing. Romanu Gordienkovi, který na jejich základě vytvořil grafický návrh aplikace.

### 4.1.1 Historie ČVUT

Prvním úkolem bylo rozhodnout, jak navrhnout kategorii zobrazující historii ČVUT. Návrh této obrazovky byl velmi důležitý, neboť měla být první obrazovkou, která se uživateli zobrazí při spuštění aplikace. Kategorie je rozdělena na dvě obrazovky, první zobrazující časovou osu s roky důležitých událostí a druhou zobrazující detail jednotlivých událostí.

Nejzajímavějším přístupem k zobrazování dat na časové ose se ukázalo řešení použité v aplikaci *Boeing Milestones*, popsané v podsekcí 3.3.1. Gesto *pinch open* (viz 2.5.3), které slouží ve zmiňované aplikaci k přístupu k detailům daného roku, bude podporováno. Bude k němu přidáno také gesto *tap* (viz 2.5.1), které usnadní manipulaci na zařízeních s menším displejem, na kterých je provádění gesta *pinch open* často neobratné. Návrh časové osy je



(a) Časová osa

(b) Detail události

Obrázek 4.1: Návrh kategorie Historie ČVUT

zobrazen na obrázku 4.1a.

Obrazovka s detailem roku bude rozdělena na dvě části. Horní část obrazovky bude vyplněna obrázkem vztahujícím se k dané události a její spodní část bude obsahovat text popisující danou událost (viz obrázek 4.1b). V této obrazovce bude použit **Parallax scrolling** efekt přítomný v aplikaci **Yahoo News Digest** popsané v podsekcí 3.3.2.

### 4.1.2 Osobnosti

Kategorie osobnosti bude rozdělena na tři podkategorie - rektori, významné osobnosti a sportovci. Rozcestník mezi těmito třemi podkategoriemi je stejně jako časová osa kategorie Historie ČVUT inspirován aplikací **Boeing Milestones** (viz obrázek 4.2).

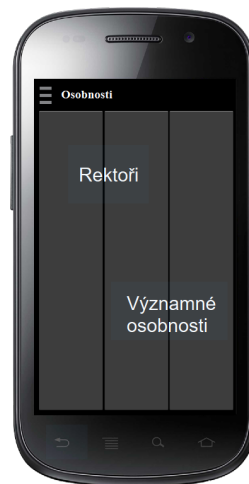
#### 4.1.2.1 Rektoři

Návrh obrazovky rektorů je založen na aplikaci **Timeline - Art Museum** zmíněné v podsekcí 3.3.4 a je zobrazen na obrázku 4.3a.

Ve spodní části obrazovky je umístěna navigační lišta s chronologickým seznamem všech rektorů seřazených podle let, ve kterých byli ve vedení univerzity. Zbytek obrazovky vyplňuje detail zobrazovaného rektora.

#### 4.1.2.2 Významné osobnosti a sportovci

Obrazovky kategorie Významných osobností a kategorie sportovců mají stejný návrh a jsou jedny z těch, u kterých se návrh pro mobilní telefony a tablety liší. Důvody odlišného návrhu jsou vysvětleny v sekci 1.2.1.



Obrázek 4.2: Návrh rozcestníku osobností



Obrázek 4.3: Návrh kategorie Osobnosti

Návrh pro mobilní telefony rozděluje tuto kategorii do dvou obrazovek - seznam a detail. Seznam osobností jednoduše zobrazuje vertikálně posouvateľný list všech osobností v abecedním pořádku, tak jako je tomu na obrázku 4.3b. Při zvolení osobnosti se zobrazí její detail, který je vidět na obrázku 4.3c.

Návrh pro tablety využívá prostoru na displeji a sjednocuje obrazovku seznamu a detailu do jedné obrazovky. V levé části se nachází list osobností a v části pravé se zobrazuje jejich detail.

### 4.1.3 Přehled

Kategorie přehledu zobrazuje základní informace o univerzitě a jejích fakultách. Rozcestník byl navržen stejně jako rozcestníky předchozích dvou kategorií (viz obrázek 4.4a).

Přehled obsahuje dva druhy detailních obrazovek - obrazovku univerzity a fakult. Jejich návrh je téměř identický, jediný rozdíl je v zobrazovaných datech. Detail univerzity zobrazuje informace o univerzitě jako celku (viz obrázek 4.4b) a detail fakult informace o jednotlivých fakultách (viz obrázek 4.4c).



(a) Rozcestník přehledu

(b) Seznam univerzity

(c) Detail fakulty

Obrázek 4.4: Návrh kategorie Přehled

### 4.1.4 Hlavní menu

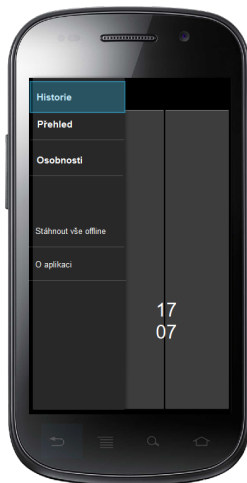
Aplikace prezentuje informace na obrazovkách, které spolu nijak nesouvisí, a proto bylo nutné navrhnout způsob, jak mezi nimi přecházet.

Návrhů bylo hned několik. Prvním z nich bylo použití tabů. Taby rozdělí jednotlivé kategorie do záložek a umístí je do lišty ve spodní části obrazovky. Mezi záložkami se dá v libovolném okamžiku přepínat. Nevýhodou tohoto přístupu byla skutečnost, že všudypřítomné taby zmenší celkovou využitelnou plochu obrazovky a v kombinaci s efektem použitým v aplikaci *Boeing Milestones* působí nevzhledně.

Další možností bylo vytvoření úvodní obrazovky sloužící jako rozcestník k jednotlivým kategoriím. Nevýhodou tohoto způsobu je prohloubení aplikace.

Posledním a nakonec také zvoleným řešením bylo využití postranního menu s položkami pro jednotlivé kategorie. Menu je schované v levé části obrazovky

a je možné ho zobrazit kliknutím na ikonu v **Navigation Baru**<sup>18</sup>. Tento přístup neomezuje celkovou plochu obrazovky ani nezvyšuje celkovou hloubku aplikace, proto byla zvolen jako vyhovující.



Obrázek 4.5: Návrh hlavního menu

## 4.2 Architektura

Vhodná volba architektury ovlivňuje schopnost aplikace reagovat na pozdější změny. Je-li architektura navržena chybně, můžou se budoucí změny stát jak časově, tak finančně velmi nákladné.

V aplikaci byla použita dvouvrstvá architektura obsahující **prezentační** a **aplikační vrstvu**.

**Aplikační vrstva** se stará o data, se kterými aplikace pracuje. V případě této aplikace se jedná o data stažená z webového backendu.

**Prezentační vrstva** zajišťuje zobrazení dat, stará se o interakci s uživatelem a komunikuje s **aplikační vrstvou**.

---

<sup>18</sup>Navigation Bar je lišta v horní části obrazovky přítomná u všech obrazovek aplikace.





---

## Realizace

V této kapitole bude rozebrána implementace aplikace provedená na základě návrhu z předchozí kapitoly. Podrobný popis programování každé obrazovky aplikace by byl zdlouhavý, proto se tato kapitola bude věnovat pouze implementaci klíčových prvků aplikace.

Práce na implementační části aplikace začala v listopadu roku 2014. Jediné, co bylo v tu dobu jasné, bylo použití efektu použitým v aplikaci *Boeing Milestones* (viz 3.3.1) jako klíčového prvku aplikace. První dva měsíce byla veškerá implementační činnost věnována jeho zprovoznění. Funční prototyp aplikace byl připraven v květnu roku 2015. Prototyp byl celý napsán v jazyce Objective-C, neboť v tu dobu dostupná verze Swift ještě nebyla použitelná a Objective-C byl hlavním programovacím jazykem pro iOS.

V červnu roku 2015 uspořádala v Kalifornii společnost Apple *Worldwide Developers Conference* (dále jen *WWDC*) [28]. Na této konferenci Apple každoročně vývojářům představuje nejnovější software a technologie z předchozího roku. Jednou z novinek bylo představení nové verze jazyka Swift, který měl nahradit Objective-C jako hlavní programovací jazyk pro všechny operační systémy společnosti Apple.

Většina vývojářů začala po *WWDC* 2015 vyvíjet aplikace v jazyce Swift. Z tohoto důvodu v listopadu 2015 začala být aplikace převáděna do jazyka Swift.

### 5.1 Způsob implementace

Pro vývoj aplikací v programu *XCode* (viz 2.2) existují dva způsoby - návrh uživatelského rozhraní v *Interface Builderu*, popsaného v sekci 2.2.3, nebo napsání celého kódu ručně.

Použití *Interface Builderu* se stává čím dál víc populární, především z důvodu, že společnost Apple do něj stále přidává nové funkcionality, které vývojářům ulehčují práci. Čím nestandardnější však návrh uživatelského rozhraní je, tím obtížnější se stává jeho vytvoření v *Interface Builderu* a ně-

## 5. REALIZACE

---

kterých funkcionalit ani docílit nelze. Z tohoto důvodu byla aplikace **Historie ČVUT** napsána bez použití **Interface Builderu**.

### 5.2 Boeing efekt

Implementace efektu použitého v aplikaci **Boeing Milestones** byla rozhodně nejnáročnějším úkolem z celé aplikace. Přinesla s sebou hned několik problémů zmíněných v následujících kapitolách.



(a) Přehled

(b) Osobnosti

Obrázek 5.1: Použití Boeing efektu

#### 5.2.1 Zobrazení

V první řadě bylo nutné rozhodnout, jak zobrazovat horizontální seznam položek. Jasným kandidátem byla třída **UIKit Frameworku Collection View** popsaná v podsekcí 2.3.3. Layout třídy **Collection View** byl nastaven tak, aby jednotlivé buňky měly výšku celé obrazovky, čímž bylo docíleno vytvoření horizontálně posuvného seznamu. Umístění štítků s názvy jednotlivých buňek se

pravidelně mění mezi umístěním v horní části, ve středu a ve spodní části buňky, tak jak je znázorněno na obrázku 5.1. Obrázek na pozadí musel být z důvodu neaktuálnosti API vložen napevno do aplikace.

### 5.2.2 Animace efektu

Po provedení gest `tap` nebo `pinch open` (viz. 2.5), začíná animace **Boeing efektu**. Animace je rozdělena na více částí.

Nejprve je vytvořen `screenshot`<sup>19</sup> obrazovky. Do obrazovky je vloženo jedno `View` (viz. 2.3.1) a dvě `Image View` (viz. 2.3.5), která ho překrývají. Jejich velikost je shodná s velikostí celé obrazovky. `View` na pozadí je upraveno tak, aby mělo shodný vzhled s pozadím detailní obrazovky. Při otevírání lze vidět, jak bude přibližně detailní obrazovka vypadat a po otevření se v připravené šabloně pouze objeví obrázek a text. Proces je znázorněn na obrázcích 5.2 a 5.3. `Screenshot` obrazovky je nastaven jako obrázek obou přidávaných `Image View`.

Následně je obrazovka rozdělena na dvě části. Středem každé buňky prochází tenká linka. Na x-ové souřadnici obrazovky, na které se linka nachází dochází k rozdělení obrazovky. Toho je docíleno tak, že pro obě `Image View` je vytvořena maska. Ta má rozměry od středu buňky k levému okraji obrazovky pro první `Image View` a od středu buňky k pravému okraji obrazovky pro druhé `Image View`. Barva obou masek je nastavena na černou, čímž se zneviditelnění veškerý obsah, který překrývají, a vytvoří se tak dvě nezávislé části obrazovky.

V tomto okamžiku se průběh efektu liší podle gesta, které jej vyvolalo. V případě gesta `pinch open` se obě `Image View` pohybují podle pohybu prstů na obrazovce. Je tedy možné obě části obrazovky kontrolovat. Při ukončení kontaktu prstů s obrazovkou je zjištěno, jestli byly prsty před odebráním spojovány nebo rozpojovány. V prvním případě jsou části spojeny a je možné znovu pracovat se seznamem položek, v opačném případě se obě části rozevrou do stran a zobrazí se detail položky.

V případě gesta `tap` je přeskočen proces s ovládáním části obrazovky a rovnou dojde k rozevření částí do stran a zobrazení detailu položky.

### 5.2.3 Návrat z detailu

Pro návrat z detailu položky na seznam položek můžou nastat dva případy, jejichž zpracování je velmi odlišné.

Řešení návratu z položky, která byla otevřena, je velmi jednoduché. Po ukončení **Boeing efektu** je obrazovce s detailem předána podoba obou `Image View`. Ty jsou vloženy za okraje obrazovky. V okamžiku, kdy je vyvolána akce `zpět`, je animován jejich pohyb směrem ke středu obrazovky, ve kterém se spojí.

<sup>19</sup>Screenshot je snímek aktuálního stavu obrazovky.

Na obrazovce s detailem je možnost se pomocí gesta `swipe` (viz 2.5.2) přesunout na detail další položky v seznamu. V tuto chvíli se stávají `ImageView` za okraji obrazovky neaktuální. Pro pochopení způsobu řešení je nejprve nutné vysvětlit práci s protokoly.

### 5.2.3.1 Protokol

Protokol [29] definuje proměnné, rozhraní metod, a další požadavky, které jsou nutné pro splnění určité funkcionality. Poté může být protokol přijat libovolnou třídou, která splňuje tyto požadavky.

### 5.2.3.2 Vyhovění protokolu

O třídě, která splňuje všechny požadavky daného protokolu se říká, že vyhovuje tomuto protokolu. Pro všechny protokolem definované proměnné třída obsahuje stejně pojmenované proměnné shodného datového typu. Pro každou protokolem definovanou metodu poskytuje její implementaci.

### 5.2.3.3 Delegace

Delegace je postup, který umožňuje třídám přenechat část svých odpovědností na třídách jiného typu. Toho lze dosáhnout vytvořením protokolu, který zapouzdří tyto odpovědnosti. Tím zaručí, že třída, která vyhovuje tomuto protokolu, obsahuje zapouzdřenou funkcionalitu. Tato třída se nazývá delegát. Třída, jejíž odpovědnosti jsou delegovány, nepotřebuje žádné informace o struktuře svého delegáta.

### 5.2.3.4 Použití v aplikaci

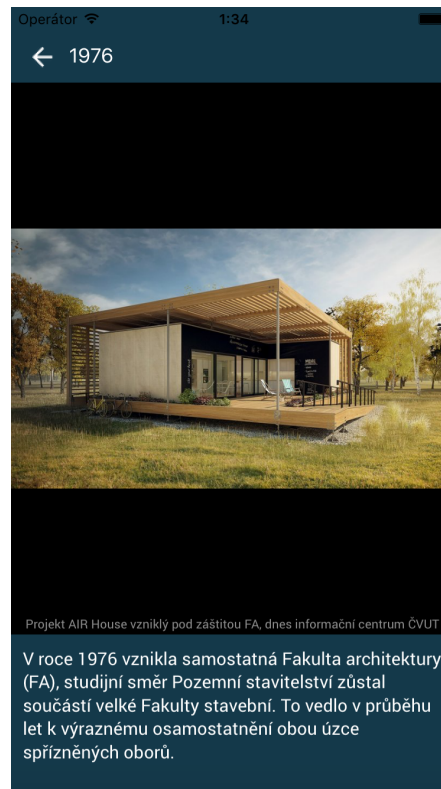
Pro vyřešení neaktuálních `ImageView` v obrazovce s detailem byl vytvořen protokol obsahující rozhraní dvou metod. Třída zobrazující seznam položek implementuje metody tohoto protokolu a po zobrazení třídy detailu se přihlásí jako delegát této třídy.

První metoda je volána při provedení gesta `swipe` a žádá svého delegáta o poskytnutí informací o položce, která má být po provedení gesta `swipe` zobrazena.

Druhá metoda je volána po úspěšném dokončení gesta `swipe` a žádá svého delegáta o zaslání obrázku, který je nastaven jako obrázek obou `ImageView` za okraji obrazovky. Delegát tuto metodu implementuje posunutím seznamu položek tak, aby buňka nové položky byla ve středu obrazovky a sejme `screenshot` obrazovky, který odešle jako návratovou hodnotu. Vše se odehrává na pozadí obrazovky, aby nedocházelo k rozptylování uživatele.



Obrázek 5.2: Rozevírání buňky detailu



Obrázek 5.3: Detail historické události

### 5.3 Page View Controller a Collection View

Na obrázku 5.4 lze vidět vzhled obrazovky rektorů. Výběr rektora lze provést dvěma způsoby - gestem `swipe` a zvolením specifického rektora na navigační liště. Pro dosažení tohoto chování bylo v jedné obrazovce zkombinováno chování tříd `Page View Controller` [30] (viz 2.4) a `Collection View` [31] (viz 2.3.3), které jsou častěji v obrazovkách osamocené.

Obě tyto třídy poskytují metody volané při různých typech událostí, ke kterým v nich dochází. Stačilo tedy vybrat metody monitorující relevantní události a implementovat je tak, aby byl při interakci s jedním prvkem vždy upraven i prvek druhý a nedocházelo tak k nekonzistencím (rektor, jehož detail je zobrazován se liší od rektora zvoleného na navigační liště).

U `Page View Controlleru` se jedná o metodu volanou ve chvíli, kdy je dokončeno gesto `swipe` (viz 2.5.2). Metoda obsahuje referenci na předchozí obrazovku a informaci o tom, jestli bylo gesto dokončeno úspěšně. Porovnáním předchozí a nové obrazovky lze určit, v jakém směru bylo gesto provedeno a upravit podle toho navigační lištu.

Pro `Collection View` je to metoda volaná při výběru některé z položek v



Obrázek 5.4: Obrazovka rektorů

navigační liště. V tomto okamžiku je nutné aktualizovat stránku zobrazující detail osobnosti. Pro zachování standardního chování `Page View Controlleru`, který při gestu `swipe` provede specifickou animaci změny stránky, je nutné zjistit, zda-li byl nově vybraný rektor rektorem dříve nebo později, než ten předchozí. Podle toho se poté provede animace daným směrem.

### 5.4 Použité knihovny

Při vývoji aplikace byly použity následující knihovny třetích stran:

- *Alamofire*: Swiftová knihovna [32] pro síťovou komunikaci protokolu HTTP<sup>20</sup> poskytující šikovné rozhraní postavené nad síťovací Apple knihovnou, které usnadňuje řadu síťovacích úkolů. V aplikaci je využívána ke stahování dat z webového back-endu.

---

<sup>20</sup>HTTP (Hypertext Transfer Protocol) je internetový protokol, který je základem komunikace na webu.

- *Argo*: Knihovna [33] starající se o mapování JSON<sup>21</sup> dat do objektů.
- *Curry*: Knihovna [34] přinášející do jazyka možnost tzv. *curryingu*. *Currying* lze jednoduše vysvětlit jako rozdělení funkce, která přijímá více argumentů do série funkcí, které přijímají část (většinou jeden) z těchto argumentů. V aplikaci je používána v kombinaci s knihovnou *Argo* na inicializaci objektů, do kterých jsou mapovány JSON data.
- *AlamofireImage*: Podpůrná knihovna [35] ke knihovně *Alamofire* starající se o stahování a cachování<sup>22</sup> obrázků.
- *AEXML*: Jednoduchá knihovna [36] pro práci s XML<sup>23</sup> soubory.
- *EZLoadingActivity*: Knihovna [37] zmrazující obrazovku v okamžiku načítání dat.

---

<sup>21</sup>JSON (JavaScript Object Notation) je datový formát určený pro přenos dat.

<sup>22</sup>Cachování je dočasné ukládání dat do paměti aplikace.

<sup>23</sup>XML (Extensible Markup Language) - značkovací jazyk sloužící k ukládání a přenosu.





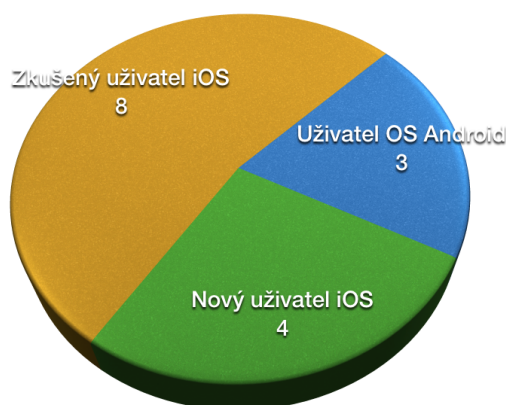
## Testování

Jelikož se jedná o informační aplikace, ve které nedochází k žádným složitým výpočtům, krajním případům, apod. nýbrž je celá založena na uživatelské interakci, bylo rozhodnuto, že jediné testování, které bude provedeno je uživatelské testování.

Distribuce Apple aplikací je poměrně složitá. Chce-li vývojář dopravit aplikaci na zařízení uživatelů, potřebuje k tomu být zapsán v **Apple Developer programu** [38], za který je nutné každoročně platit. Jedinou možností tedy bylo poskytování testerům mobilní telefon, na kterém byla aplikace spuštěna a pozorovat jejich počínání.

### 6.1 Uživatelské testy

Aplikace byla otestována patnácti uživateli. Jejich zkušenost s operačním systémem iOS je vyobrazena na obrázku 6.1.



Obrázek 6.1: Rozdělení testerů podle zkušenosti s iOS

Poznatky z testování jsou následující:

## 6. TESTOVÁNÍ

---

- Všichni testéři pochopili, že seznamem položek historických událostí a fakult se dá horizontálně projíždět.
- Většina testerů měla problém s otevřením detailu položek v seznamu položek. Každých z nich se pokoušel položku rozevřít jedním poklepáním a až po nějaké době vyzkoušeli dvojitě poklepání.
- Pouze jeden tester objevil možnost otevření položky gestem `pinch open`.
- Někteří s testerů se snažili otevřít postranní menu gestem `swipe` po `Navigation Baru`.
- Dvěma testerům zabralo dlouhou dobu než zjistili, jak přejít v obrazovce rektorů na dalšího rektora.
- Pět testerů neobjevilo možnost posouvání spodní lišty s chronologickým seznamem rektorů v obrazovce rektorů.
- Tři testéři neobjevili možnost přechodu na další historickou událost pomocí gesta `swipe`.
- Tři testéři neobjevili možnost přechodu na další fakultu pomocí gesta `swipe`.

Na základě poznatků z testování byly v aplikaci provedeny následující změny:

- Gesto pro otevření detailu položek bylo zaměněno z dvojitého poklepání na jednoduché poklepání.
- Byla přidána možnost otevření potranního menu provedením gesta `swipe` po `Navigation Baru`.

---

# Závěr

Úkolem této bakalářské práce bylo vytvořit mobilní aplikaci pro operační systém iOS, která bude prezentovat důležité historické události ČVUT, chronologii rektorů univerzity, významné vědecké osobnosti a vznik jednotlivých fakult. Na základě těchto požadavků byla vytvořena aplikace Historie ČVUT, která kromě výše zmíněných položek obsahuje ještě informace o ČVUT, informace o fakultách ČVUT a informace o úspěšných sportovcích.

Součástí práce bylo shromáždění výše zmiňovaných dat, na kterém jsem spolupracoval s diplomantem vyvíjejícím verzi aplikace pro operační systém Android. Po průzkumu podobných aplikací byly vytvořeny wireframy, podle kterých navrhl designer grafiku aplikace.

Aplikace je funkční jak na mobilních telefonech, tak na tabletech, podporuje český a anglický jazyk a je napojená na webové API. Toto API bylo vytvořené dalším diplomantem, který na projektu spolupracoval.

Implementace aplikace přinesla mnohé problémy. Při jejich řešení jsem pochopil některé pokročilejší postupy programování v jazyce Swift, jako například práci s protokoly a animacemi. Dále jsem se dozvěděl zajímavé informace o světě univerzitě.

## Rozšiřitelnost

Dalším přidaným prvkem aplikace by mohla být virtuální prohlídka po areálu univerzity, která již byla součástí návrhu aplikace. Dále implementace obrázků, která by se zobrazila pouze při prvním spuštění aplikace a objasnila uživatelům její ovládání. Počet uživatelů, kteří si nevědí rady s ovládáním aplikace by se tím rapidně snížil.



---

# Literatura

- [1] Ackee s.r.o: Vývoj mobilních a webových aplikací | Ackee. [online], [cit. 14.5. 2016]. Dostupné z: <https://www.ackee.cz/>
- [2] Vojtěch Bartoš: HisČVUT API. [online], [cit. 14.5. 2016]. Dostupné z: <http://docs.hiscvut.apiary.io/>
- [3] Apple Inc.: What is iOS? [online], [cit. 14.5. 2016]. Dostupné z: <http://www.apple.com/ios/>
- [4] Apple Inc.: About Objective-C. [online], [cit. 14.5. 2016]. Dostupné z: <https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>
- [5] Peter William Lount: Smalltalk.org. [cit. 14.5. 2016]. Dostupné z: <http://www.smalltalk.org/>
- [6] Apple Inc.: About Swift. [online], [cit. 14.5. 2016]. Dostupné z: <https://swift.org/about/>
- [7] Apple Inc.: About the iOS Technologies. [online], [cit. 14.5. 2016]. Dostupné z: <https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>
- [8] Apple Inc.: Cocoa Touch Layer. [online], [cit. 14.5. 2016]. Dostupné z: [https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechnologies/iPhoneOSTechnologies.html#//apple\\_ref/doc/uid/TP40007898-CH3-SW1](https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechnologies/iPhoneOSTechnologies.html#//apple_ref/doc/uid/TP40007898-CH3-SW1)
- [9] Apple Inc.: Media Layer. [online], [cit. 14.5. 2016]. Dostupné z: <https://developer.apple.com/library/ios/documentation/>

Miscellaneous/Conceptual/iPhoneOSTechOverview/MediaLayer/  
MediaLayer.html#/apple\_ref/doc/uid/TP40007898-CH9-SW4

- [10] Apple Inc.: Core Services Layer. [online], [cit. 14.5. 2016]. Dostupné z: [https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/CoreServicesLayer/CoreServicesLayer.html#/apple\\_ref/doc/uid/TP40007898-CH10-SW5](https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/CoreServicesLayer/CoreServicesLayer.html#/apple_ref/doc/uid/TP40007898-CH10-SW5)
- [11] Apple Inc.: Core OS Layer. [online], [cit. 14.5. 2016]. Dostupné z: [https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/CoreOSLayer/CoreOSLayer.html#/apple\\_ref/doc/uid/TP40007898-CH11-SW1](https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/CoreOSLayer/CoreOSLayer.html#/apple_ref/doc/uid/TP40007898-CH11-SW1)
- [12] Apple Inc.: XCode - At a Glance. [online], [cit. 14.5. 2016]. Dostupné z: [https://developer.apple.com/library/ios/documentation/ToolsLanguages/Conceptual/Xcode\\_Overview/](https://developer.apple.com/library/ios/documentation/ToolsLanguages/Conceptual/Xcode_Overview/)
- [13] Apple Inc.: About Simulator. [online], [cit. 14.5. 2016]. Dostupné z: [https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/iOS\\_Simulator\\_Guide/Introduction/Introduction.html](https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/iOS_Simulator_Guide/Introduction/Introduction.html)
- [14] Apple Inc.: About Playgrounds. [online], [cit. 14.5. 2016]. Dostupné z: [https://developer.apple.com/library/ios/recipes/Playground\\_Help/Chapters/AboutPlaygrounds.html](https://developer.apple.com/library/ios/recipes/Playground_Help/Chapters/AboutPlaygrounds.html)
- [15] Apple Inc.: Interface Builder Built-In. [online], [cit. 14.5. 2016]. Dostupné z: <https://developer.apple.com/xcode/interface-builder/>
- [16] Tayerlová, M.: *Česká technika*. Praha: České vysoké učení technické v Praze, druhé vydání, 2004, ISBN 8001031659.
- [17] Dubnová, D.: Informace pro uchazeče o studium. [online], [cit. 14.5. 2016]. Dostupné z: <https://www.cvut.cz/informace-pro-uchazece-o-studium>
- [18] Jarmila Linderová: Fakulty a pracoviště. [online], [cit. 14.5. 2016]. Dostupné z: <https://www.cvut.cz/fakulty-a-pracoviste>
- [19] České vysoké učení technické v Praze: Sedm statečných ČVUT. [online], [cit. 14.5. 2016]. Dostupné z: <http://www.sedmstatecnych.cz/>
- [20] České vysoké učení technické v Praze: Rektoři ČVUT. [online], [cit. 14.5. 2016]. Dostupné z: <http://intranet.cvut.cz/fotobanka/osobnosti/rektori>

- 
- [21] Valjent, Z.: Sport na ČVUT kdysi a dnes. [online], [cit. 14.5. 2016]. Dostupné z: [http://www.utvs.cvut.cz/lectors/zv\\_sport\\_CVUT\\_kdysi\\_dnes.pdf](http://www.utvs.cvut.cz/lectors/zv_sport_CVUT_kdysi_dnes.pdf)
- [22] The Boeing Company: Boeing Milestones. [software], [cit. 14.5. 2016]. Dostupné z: <https://itunes.apple.com/us/app/boeing-milestones/id510294916?mt=8>
- [23] Yahoo! Inc.: Yahoo News Digest. [software], [cit. 14.5. 2016]. Dostupné z: <https://itunes.apple.com/us/app/yahoo-news-digest/id784982356?mt=8>
- [24] Web Designer: Parallax Scrolling Tutorial: How To Make a Parallax Website. [online], [cit. 14.5. 2016]. Dostupné z: <http://1stwebdesigner.com/parallax-scrolling-tutorial/>
- [25] You Visit: Mississippi State University. [software], [cit. 14.5. 2016]. Dostupné z: <https://play.google.com/store/apps/details?id=com.yc360.college.msstate>
- [26] Netsco: Timeline - Art Museum. [software], [cit. 14.5. 2016]. Dostupné z: <https://play.google.com/store/apps/details?id=kr.netsco.artist>
- [27] Evolus: Pencil Project. [software], [cit. 14.5. 2016]. Dostupné z: <http://pencil.evolus.vn/>
- [28] Apple Inc.: Worldwide Developers Conference. [online], [cit. 14.5. 2016]. Dostupné z: <https://developer.apple.com/videos/wwdc2015/>
- [29] Apple Inc.: Protocols. [online], [cit. 14.5. 2016]. Dostupné z: [https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift\\_Programming\\_Language/Protocols.html](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Protocols.html)
- [30] Apple Inc.: UINavigationController. [online], [cit. 14.5. 2016]. Dostupné z: <https://developer.apple.com/library/ios/documentation/UIKit/Reference/UINavigationControllerClassReferenceClassRef/>
- [31] Apple Inc.: UICollectionView. [online], [cit. 14.5. 2016]. Dostupné z: [https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView\\_class/](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/)
- [32] Alamofire Software Foundation: Alamofire. [software], [cit. 14.5. 2016]. Dostupné z: <https://github.com/Alamofire/Alamofire>
- [33] thoughtbot, inc.: Argo. [software], [cit. 14.5. 2016]. Dostupné z: <https://github.com/thoughtbot/Argo>
- [34] thoughtbot, inc.: Curry. [software], [cit. 14.5. 2016]. Dostupné z: <https://github.com/thoughtbot/Curry>

## LITERATURA

---

- [35] Alamofire Software Foundation: AlamofireImage. [software], [cit. 14.5. 2016]. Dostupné z: <https://github.com/Alamofire/AlamofireImage>
- [36] Marko Tadic: AEXML. [software], [cit. 14.5. 2016]. Dostupné z: <https://github.com/tadija/AEXML>
- [37] Goktug Yilmaz: EZLoadingActivity. [software], duben 2015, [cit. 14.5. 2016]. Dostupné z: <https://github.com/goktugyil/EZLoadingActivity>
- [38] Apple Inc.: Apple Developer Program. [online], [cit. 14.5. 2016]. Dostupné z: <https://developer.apple.com/programs/>



## Seznam použitých zkratk

**API** Application Programming Interface

**ČVUT** České vysoké učení technické

**HTTP** Hypertext Transfer Protocol

**JSON** JavaScript Object Notation

**OS** Operační systém

**XML** Extensible Markup Language



---

## Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
application	
├── screenshots.....	screenshoty obrazovek aplikace
├── videos.....	videa průchodu aplikací
src	
├── thesis.....	složka s $\text{\LaTeX}$ zdrojovými kódy práce
├── impl.....	zdrojové kódy implementace
text.....	text práce
├── thesis.pdf.....	text práce ve formátu PDF