

Sem vložte zadání Vaší práce.



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



Diplomová práce

## **Letecké záznamy pro iOS pomocí moderních architektur a FRP**

***Bc. Martin Žid***

Vedoucí práce: Ing. Dominik Veselý

29. srpna 2017



---

## Poděkování

Doplňte, máte-li komu a za co děkovat. V opačném případě úplně odstraňte tento příkaz.



---

## 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 29. srpna 2017

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2017 Martin Žid. 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**

Žid, Martin. *Letecké záznamy pro iOS pomocí moderních architektur a FRP*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.



---

# Abstrakt

Tato práce realizuje iOS aplikaci pro evidenci letů. Práce analyzuje obdobné aplikace a předpisy pro piloty České republiky, podle nichž probíhá návrh funkcionality vytvářené aplikace. Podle návrhu je následně zvolena vhodná architektura a vytvořeno uživatelského rozhraní v podobě wireframů.

Aplikace je implementována s použitím zvolené architektury a pomocí principů FRP. V průběhu implementace aplikace jsou realizovány jednotkové testy a na konci jsou provedeny uživatelské testy. Na základě výsledků testů je aplikace upravena do finální podoby.

V práci jsem vytvořil funkční iOS aplikaci s využitím moderní architektury a principů FRP. Aplikace bude sloužit pilotům České republiky pro elektronickou evidenci letů a bude jim také ulehčovat administrativu s evidencí spojenou.

V příloze této bakalářské práce je možné nalézt všechny zdrojové kódy jak aplikace, tak i testů společně s vytvořenými wireframy.

**Klíčová slova** mobilní aplikace pro evidenci letů, iOS, Swift, FRP, ReactiveCocoa, MVVM architektura

# Abstract

Sem doplňte ekvivalent abstraktu Vaší práce v angličtině.

**Keywords** flight records mobile application, iOS, Swift, FRP, ReactiveCocoa, MVVM architecture

---

# Obsah

Úvod	1
1 Cíl práce	3
2 Analýza a návrh	5
2.1 Architektury při tvorbě iOS aplikací . . . . .	5
3 Realizace	7
Závěr	9
Literatura	11
A Seznam použitých zkratek	13
B Obsah přiloženého CD	15



---

## Seznam obrázků

2.1	Model-View-Controller diagram . . . . .	6
2.2	Model-View-Controller při vývoji iOS aplikace . . . . .	6



---

# Úvod





## **Cíl práce**



## Analýza a návrh

### 2.1 Architektury při tvorbě iOS aplikací

Při tvorbě iOS aplikace je možné si vybrat z několika architektur. V této kapitole budu rozebírat pouze MVC, MVVM a VIPER.

#### 2.1.1 MVC

Architektura MVC je zkratka pro „Model View Controller“ neboli tři komponenty, ze kterých se architektura skládá. Jedná se o softwarovou architekturu, které se velice často používá při tvorbě aplikací s uživatelským rozhraním. [1]

- *Model* definuje jaká data aplikace obsahuje a pokud dojde k jakékoliv změně, tak informuje buď *Controller* nebo *View* (tzv. své observery). [2]
- *View* vrstva je prezentována samotnému uživateli. Tedy jsou zde zobrazena aplikační data a je zachycována uživatelova práce s aplikací. [1]
- *Controller* je vrstva mezi *View* a *Model* zabezpečující logiku aplikace. Stará se o promítnutí změn do *View* pokud se změní *Model*. Zároveň provádí úpravy v *Model* při uživatelově manipulaci s *View*. [2]

Však co se týče iOS vývoje, vrstvy *View* a *Controller* jsou téměř spojeny, protože *Controller* je příliš úzce zapojený do životního cyklu *View*. [3]

Základní myšlenku MVC a MVC při vývoji iOS aplikace ukazují obrázky 2.1 a 2.2.

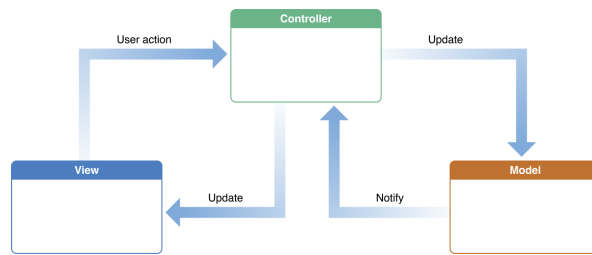
MVC je základní architekturou pro tvorbu iOS aplikací. Není však jedinou možností.

#### 2.1.2 MVVM

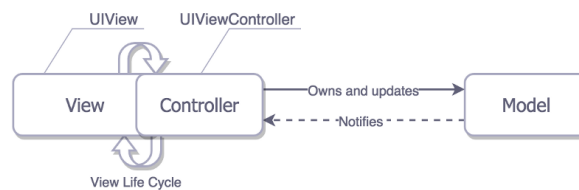
Architektura MVVM má obdobné koncepce jako MVC. Jedná se také o zkratku, tentokrát „Model-View-ViewModel“. [5]

## 2. ANALÝZA A NÁVRH

---



Obrázek 2.1: Model-View-Controller diagram [4]



Obrázek 2.2: Model-View-Controller při vývoji iOS aplikace [3]

- *Model* je totožný s *Model* vrstvou architektury MVC, jedná se tedy o datovou část aplikace.
- *View* prezentuje aplikační data uživateli a monitoruje jeho akce. Však, jak již bylo zmíněno dříve, u iOS aplikací se jedná spíše o vrstvu *View/Viewcontroller*. Tato vrstva obsahuje pouze minimum logiky aplikace a reaguje hlavně na *ViewModel*. [6]
- *ViewModel* spojuje *View* a *Model* a zajišťuje hlavní logiku aplikace. *ViewModel* tedy komunikuje s *Model* a jeho metodami a následně připravuje data pro *View*. Obsahuje také implementaci funkcí, které reagují a zpracovávají akce uživatele, např.: kliknutí na tlačítko. [5]

Tedy pro shrnutí rozdílů MVC a MVVM u iOS bych zmínil to, že iOS MVC má ve výsledku téměř jen dvě vrstvy *View/Viewcontroller* a *Model*. Když potom uvažujeme architekturu MVVM *View/Viewcontroller* je opravdu pouze jednou vrstvou a mezi ní a *Model* je vložena nová vrstva *ViewModel*, která je spojuje a do které je přesunuta i většina aplikační logiky.

Mezi výhody architektury MVVM oproti MVC patří např.:

- poskytuje návrhový princip tzv. separation of concerns, neboli oddělení zájmů;
- zlepšuje možnost testovatelnosti aplikace.

## **Realizace**



---

## **Závěr**





---

## Literatura

- [1] MVC Architecture. *MDN web docs* [online]. Mountain View, California, United States: Mozilla and individual contributors, © 2005-2017, [cit. 2017-08-29]. Dostupné z: <https://developer.mozilla.org/cs/>
- [2] MVC Architecture. *Developer Chrome* [online]. Silicon Valley: Google, © 2017, [cit. 2017-08-29]. Dostupné z: [https://developer.mozilla.org/en-US/Apps/Fundamentals/Modern\\_web\\_app\\_architecture/MVC\\_architecture](https://developer.mozilla.org/en-US/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture)
- [3] Orlov, B.: IOS Architecture Patterns. *Medium* [online], 2015, [cit. 2017-08-29]. Dostupné z: <https://medium.com/ios-os-x-development/ios-architecture-patterns-ecba4c38de52>
- [4] Model-View-Controller. *Apple Developer* [online]. California, U.S.: Apple Inc., © 2015, [cit. 2017-08-29]. Dostupné z: <https://developer.apple.com/library/content/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>
- [5] The MVVM Pattern. *Microsoft Developer Network* [online]. Washington, U.S.: Microsoft, © 2017, [cit. 2017-08-29]. Dostupné z: <https://msdn.microsoft.com/en-us/library/hh848246.aspx>
- [6] Morrison, J.; Schmidt, M.: IOS Design Patterns: MVC and MVVM. *CapTech*, 2014, [cit. 2017-08-29]. Dostupné z: <https://www.captchiconsulting.com/blogs/ios-design-patterns-mvc-and-mvvm>



## Seznam použitých zkratk

**GUI** Graphical user interface

**XML** Extensible markup language



## Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	exe .....	adresář se spustitelnou formou implementace
	src	
	impl.....	zdrojové kódy implementace
	thesis .....	zdrojová forma práce ve formátu L <sup>A</sup> T <sub>E</sub> X
	text .....	text práce
	thesis.pdf .....	text práce ve formátu PDF
	thesis.ps .....	text práce ve formátu PS