



Zadání bakalářské práce

Název:	Návrh mobilní aplikace letového deníku pro paragliding
Student:	Jindřich Doležal
Vedoucí:	Ing. Petra Pavlíčková, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Informační systémy a management
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2024/2025

Pokyny pro vypracování

Cílem bakalářské práce je navrhnout a naprogramovat mobilní aplikaci umožňující správu, resp. vedení leteckého deníku pro paragliding.

- 1) Analyzujte současné webové řešení.
- 2) Proveďte sběr požadavků pro novou mobilní aplikaci.
- 3) Navrhněte letový deník pro paragliding.
- 4) Na základě návrhu implementujte mobilní aplikaci.
- 5) Letový deník pro paragliding otestujte a proveďte uživatelské testy.
- 6) Vyhodnoťte a doporučte další rozvoj mobilní aplikace.
- 7) Zhodnoťte výhody a nevýhody, popřípadě ekonomický a business dopad vyvinuté mobilní aplikace.

Bakalářská práce

NÁVRH MOBILNÍ APLIKACE LETOVÉHO DENÍKU PRO PARAGLIDING

Jindřich Doležal

Fakulta informačních technologií
Katedra softwarového inženýrství
Vedoucí: Ing. Petra Pavlíčková, Ph.D.
16. května 2024

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2024 Jindřich Doležal. 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, je nezbytný souhlas autora.

Odkaz na tuto práci: Doležal Jindřich. *Návrh mobilní aplikace letového deníku pro paragliding*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2024.

Obsah

Poděkování	vi
Prohlášení	vii
Abstrakt	viii
Seznam zkratk	ix
Úvod	1
1 Cíl práce	2
2 Teoretická část	3
2.1 Architektury	3
2.1.1 Clean architektura	3
2.1.2 MVVM	4
2.1.3 MVI	5
2.2 Dependency injection	6
2.2.1 Factory	6
2.3 Frameworky pro vývoj multiplatformních aplikací	7
2.3.1 Flutter	7
2.3.2 React Native	7
2.3.3 Kotlin Multiplatform	8
2.4 Firebase	9
2.4.1 Firebase Firestore	10
2.4.2 Firebase Storage	10
2.4.3 Firebase Authentication	10
2.5 Jetpack Compose	10
2.6 Multiplatform Settings	11
2.7 SQLDelight a SQLite	11
2.8 Swift a Swift UI	11
2.9 Twine	12
2.10 XCTest	12
3 Analýza	13
3.1 Analýza současného webového řešení	13
3.1.1 Představení současného webového řešení	13
3.1.2 Vzhled a ovládání	13
3.1.3 Fungování webového řešení	15
3.1.4 Shrnutí analýzy současného webového řešení	19
3.2 Alternativní fungující mobilní aplikace	20
3.2.1 Flyskyhy	20
3.2.2 Gaggle - Flight Recorder	21
3.2.3 Rogallo BlackBox	21

3.2.4	Další aplikace a shrnutí	22
3.3	Požadavky	23
3.3.1	Funkční	23
3.3.2	Nefunkční	24
3.4	Model případů užití	24
3.4.1	Aktéři	24
3.4.2	Případy užití	24
3.4.3	Mapování požadavků na případy užití	29
3.5	Doménový model	30
4	Návrh mobilní aplikace	32
4.1	Název navrhované aplikace	32
4.2	Platforma aplikace	32
4.2.1	Vývoj multiplatformní aplikace	33
4.3	Databáze a napojení na webové řešení	34
4.4	Architektura aplikace	34
4.4.1	Sdílená část	35
4.4.2	Část specifická pro platformu	35
4.5	Návrh rozhraní aplikace	36
4.5.1	Cíl návrhu rozhraní aplikace	36
4.5.2	Barvy a ikona navrhované aplikace	36
4.5.3	Prototyp aplikace	37
5	Implementace	40
5.1	Vývojové nástroje	40
5.2	Začátek vývoje aplikace	40
5.3	Databáze a napojení na webové řešení	42
5.4	Autorizace a autentizace	43
5.5	Ukázka implementace	43
5.5.1	Sdílená část	43
5.5.2	Část specifická pro platformu	45
5.6	Zhodnocení implementace	49
6	Testování	50
6.1	Testování aplikace	50
6.2	Uživatelské testování	50
6.2.1	Vstupní dotazník	51
6.2.2	Testovací scénář	51
6.2.3	Výstupní dotazník	52
6.2.4	Testování s uživateli	53
6.2.5	Vyhodnocení uživatelského testování	61
7	Zhodnocení aplikace a doporučení dalšího vývoje	62
7.1	Doporučení dalšího vývoje aplikace	63
7.2	Přínosy aplikace	63
8	Závěr	65
A	Ukázka uživatelského rozhraní aplikace	66
	Obsah příloh	72

Seznam obrázků

2.1	Diagram reprezentující myšlenku Clean architekturu	4
2.2	Grafické znázornění architektury MVVM	5
2.3	Grafické znázornění architektury MVI [9]	6
2.4	Kotlin Multiplatform [23]	9
3.1	Snímek obrazovky webu XContest na mobilním zařízení [43]	14
3.2	Snímek obrazovky webu XContest na počítači [43]	14
3.3	Ukázka závodu UP Krkonoše Open 2024 [44]	15
3.4	Ukázka zobrazení detailu přeletu na XContest [45]	16
3.5	Ukázka sekce Můj profil na XContest [46]	18
3.6	Ukázka aplikace Flyskyhy [47]	20
3.7	Ukázka aplikace Gaggle - Flight Recorder [49] [50] [51] [52]	21
3.8	Ukázka aplikace Rogalo BlackBox [54]	22
3.9	Model případů užití	25
3.11	Doménový model navrhované mobilní aplikace	30
4.1	Průzkum společnosti JetBrains týkající se sdílení kódu napříč platformami [59]	33
4.2	Barevná paleta používaná při návrhu aplikace XCAApp	36
4.3	Ikona aplikace XCAApp	37
4.4	Low-Fi prototyp aplikace vytvořený v desktopové aplikaci Balsamiq	37
4.5	Hi-fi prototyp aplikace vytvořený v aplikaci Sketch po opravení chyb objevené během testování	38
5.1	Ukázka uživatelského rozhraní implementovaného prototypu aplikace XCAApp	49
A.1	Ikona aplikace na ploše a spouštěcí obrazovka prototypu aplikace XCAApp	66
A.2	Ukázka profilu pilota a detailu letu	67
A.3	Ukázka obrazovky změny údajů profilu, nastavení, obrazovky profilu a deníku pro nepřihlášeného uživatele	67

Seznam tabulek

3.10	Mapování požadavků na případy užití	29
------	-----------------------------------------------	----

Seznam výpisů kódu

5.1	Přidání knihoven z Firebise Kotlin SDK do Gradle	41
5.2	Konfigurace závislostí ve sdíleném modulu	42
5.3	Změna v souboru KMMConfig.kt	42
5.4	Konfigurace Firebase na platformě iOS	42
5.5	Ukázka implementace třídy FlightSource a její funkce getFlights	44
5.6	Ukázka implementace třídy FlightSourceImpl	44
5.7	Ukázka implementace třídy FlightRepositoryImpl	45
5.8	Ukázka implementace rozhraní GetAllFlightsUseCase	45
5.9	Ukázka lokalizace řetězců pro nástroj Twine	46
5.10	Ukázka použití lokalizovaných řetězců ve SwiftUI	46
5.11	Ukázka Viewmodelu pro zobrazení příspěvků	47
5.12	Ukázka View pro zobrazení příspěvků	48

Chtěl bych poděkovat především Ing. Petře Pavlíčkové, Ph.D. jakožto vedoucímu mé bakalářské práce, za její ochotu, vstřícnost a cenné rady, které mi pomohly během vytváření bakalářské práce. Dále bych chtěl poděkovat Tedeáši Sosínovi a jeho kolegům v Matee za poskytnuté konzultace. V neposlední řadě bych chtěl poděkovat rodině a přátelům za podporu při tvorbě této práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při 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. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, 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.

V Praze dne 16. května 2024

Abstrakt

Tato bakalářská práce se zabývá analýzou a návrhem designu mobilní aplikace, která je napojená na webovou službu umožňující vést letový deník pro paragliding. Hlavním cílem je vytvoření prototypu mobilní aplikace, která svým uživatelům slouží jako letový deník a přinese funkcionality, které uživatelé znají z webového rozhraní služby. Pro jeho naplnění je proveden rozbor webového rozhraní a analýza současných řešení na trhu. Dále je navržen a vytvořen prototyp mobilní aplikace. Závěrem je na základě testování s potenciálními uživateli provedena analýza a navrženy úpravy pro další rozvoj aplikace.

Výsledek práce může sloužit jako základ pro vytvoření multiplatformní mobilní aplikace, která bude používána paraglidisty po celé planetě a přinese lepší uživatelský zážitek a snazší přístup k vedení letového deníku.

Klíčová slova letový deník, paragliding, multiplatformní mobilní aplikace, Kotlin Multiplatform, uživatelské testování, iOS, Android

Abstract

This bachelor thesis deals with the analysis and design of a mobile application that is connected to a web service that allows to keep a pilot logbook for paragliding. The main goal is to create a prototype of a mobile application that serves as a flight diary for its users and brings the functionalities that users are familiar with from the web interface of the service. In order to achieve this goal, an analysis of the web interface and an analysis of current solutions on the market will be carried out. In addition, a prototype of the mobile application is designed and developed. Finally, based on tests with potential users, an analysis will be made and modifications will be proposed for the further development of the application.

The result of the work can be used as a basis for the creation of a multiplatform mobile application that will be used by paragliders all over the world, bringing a better user experience and easier access to keeping a pilot logbook.

Keywords pilot logbook, paragliding, multiplatform mobile application, Kotlin Multiplatform, user testing, iOS, Android

Seznam zkratek

API	Application programming interface
ČPP	Český pohár paraglidingu
ČVUT	České vysoké učení technické
DTO	Data transfer object
FA	Fakulta architektury
FIT	Fakulta informačních technologií
GPS	Global Positioning System
iOS	iPhone operating system
IGC	Flight data format
JS	JavaScript
KMP	Kotlin Multiplatform
MVC	Model-View-Controller
MVI	Model-View-Intent
MVVM	Model-ViewModel-Model
NASA	National Aeronautics and Space Administration
SDK	Software development kit
SQL	Structured Query Language
UC	Use Case
UI	User Interface
UX	User Experience

Úvod

Člověk se poprvé vznesl k nebi a pohlédl k zemi v roce 1783 díky balónu bratří Montgolfiérů. O dvě stě let později Jean-Cloude Bétemps a André Bohn vzlétli k nebi pomocí padákové konstrukce, kterou původně vyvíjela NASA pro návrat kapsle s astronauty z vesmíru. Jejich let vzbudil velkou pozornost médií, získal si své příznivce a Bétemps je dodnes považován za zakladatele paraglidingu. Tento sport se stal velice rychle populární a v roce 1989 se uskutečnil první světový pohár v paraglidingu v Rakousku. [1]

Paragliding se řadí mezi letové sporty a stejně jako ostatní není jednoduché se ho naučit, hrozí možnost pádu a vážného zranění. Proto je komunita velice důležitá. Letci si navzájem pomáhají, radí si a sdílejí mezi sebou své zkušenosti z letů, které absolvovali. Stejně jako piloti v letadle si i paraglidisté vedou letový deník, do kterého si zaznamenávají své přelety. To jim může pomoci ke zdokonalení techniky letu, sledování pokroku a sdílení zážitků s ostatními.

Možností, jak si vést letový deník, je v dnešní době spousta. Platforma XContest přináší možnost vedení letového deníku, být součástí komunity a účastnit se soutěží v paraglidingu, čímž se odlišuje od ostatních platforem a patří mezi jedny z největších na světě. Přestože je tak populární, nabízí pouze zastaralé webové rozhraní, které již dnešním požadavkům uživatelů nedostačuje.

Toto téma je zvoleno, protože je možnost přijít s řešením, které by přineslo pilotům využívací službu XContest snazší přístup, lepší ovládání a přehlednější prostředí pro vedení jejich letového deníku. Dané řešení by mohlo být použito pro další vývoj a jako základ pro vytvoření plnohodnotné mobilní aplikace, kterou by mohli piloti po celém světě používat.

Bakalářská práce je členěna do pěti hlavních částí – analýza, návrh aplikace, implementace, testování a dopad aplikace. Těmto částem předchází kapitola s cílem práce a kapitola popisující použité technologie a postupy.

První část bakalářské práce se zabývá analýzou současného webového řešení a to jak z pohledu uživatelů, kteří službu používají, tak z hlediska datové struktury a funkčnosti. Dále porovná existující mobilní aplikace, které umožňují vést letový deník pro paragliding. Na základě těchto informací jsou vytvořeny funkční a nefunkční požadavky a sestaven model případů užití.

Ve druhé části práce je navržena architektura aplikace letového deníku pro paragliding na základě konkrétních požadavků a navrhnutá struktura aplikace i její design.

Ve třetí části práce je implementována mobilní aplikace, která byla navržena. Jsou popsány použité vývojové nástroje a ukázány části implementace aplikace.

Ve čtvrté části práce je provedeno testování vytvořené aplikace a uživatelské testování aplikace, u kterého je sbírána zpětná vazba.

V poslední části jsou zhodnoceny výhody a nevýhody mobilní aplikace. Na závěr jsou navrženy další úpravy aplikace a zhodnocen výsledek práce.



Kapitola 1

Cíl práce

Cílem bakalářské práce je navrhnout a implementovat aplikaci letového deníku pro paragliding. Aplikace bude uživatelům poskytovat funkcionality pro správu letového deníku, prohlížení informací o soutěžích a správu uživatelského profilu. Tímto způsobem doplní stávající webové řešení této problematiky a zpřístupní uživatelům pohodlnější přístup k jeho funkcím na mobilních zařízeních. Aplikace bude vyvíjena multiplatformně, aby byla dostupná pro více mobilních operačních systémů.

Cílem analytické části je provést analýzu stávajícího webového řešení pro správu letového deníku pro paragliding. Analýza poskytne vhled do fungování a funkcionalit webové aplikace. Na základě získaných poznatků budou stanoveny požadavky, které by měla nově navrhovaná mobilní aplikace splňovat.

Díky poznatkům z analytické části bude navrženo vhodné řešení, které bude reflektovat stanovené požadavky a důraz bude kladen na multiplatformní povahu řešení. Toto řešení bude následně dle návrhu implementováno.

Vytvořená aplikace se otestuje, aby vše správně fungovalo. Pro maximální spolehlivost aplikace bude aplikace podrobena technickému i uživatelskému testování.

Po dokončení implementace bude mobilní aplikace podrobena řadě testů za účelem ověření její správné funkčnosti a kvality. Nejprve proběhne technické testování, jehož cílem bude otestovat jednotlivé moduly aplikace, následně projde aplikace uživatelským testováním, jehož prostřednictvím budou otestovány jednotlivé uživatelské scénáře a uživatelské rozhraní.

Na závěr bude doporučen další rozvoj aplikace a zhodnocen ekonomický a business dopad aplikace. Výsledná práce by měla přinést užitek jak pro koncové uživatele, tak i pro společnost spravující současné webové řešení. Ta by získala návrh možného řešení pro vytvoření vlastní mobilní aplikace, která by sloužila jako alternativa k webovému rozhraní. Tato mobilní aplikace by uživatelům poskytla lepší zážitek při používání na mobilních zařízeních a mohla by potenciálně rozšířit uživatelskou základnu.

Kapitola 2

Teoretická část

V této kapitole bakalářské práce jsou popsány technologie a metody, které se používají při návrhu mobilní aplikace letového deníku pro paragliding.

2.1 Architektury

V následující podkapitole bakalářské práce jsou podrobně popsány architektury, které jsou v bakalářské práci zmíněny a použity při tvorbě mobilní aplikace letového deníku pro paragliding.

2.1.1 Clean architektura

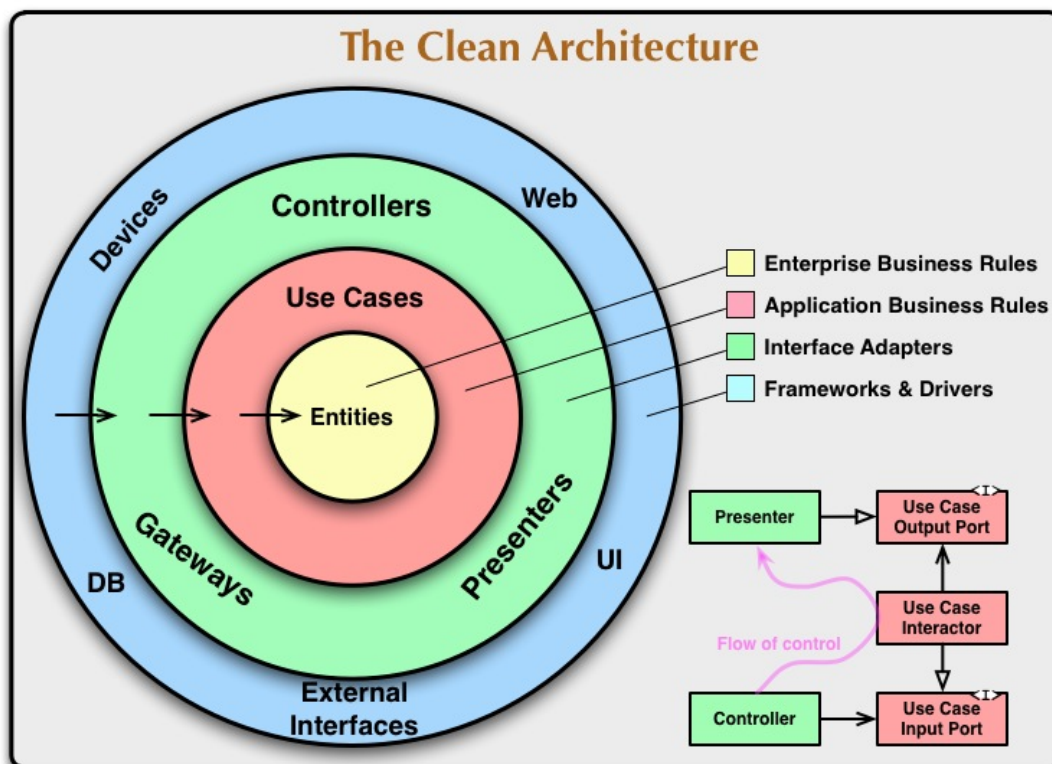
„Cílem softwarové architektury je minimalizovat lidské zdroje potřebné k vybudování a údržbě požadovaného systému.“- Robert C. Martin, Clean architektura [2]

Clean architektura je navržena Robertem C. Martinem (Uncle Bob), který ji popsal ve své knize s názvem *„Clean Architecture: A Craftsman’s Guide to Software Structure and Design“* v roce 2017. Cílem architektury je oddělení jednotlivých zájmů, kterého dosahuje rozdělením softwaru do vrstev. To přináší své výhody.

- Architektura není závislá na existujících knihovnách. Díky tomu ji lze využívat jako frameworky a nebýt limitován jejich omezeními.
- Snazší testování byznys pravidel a scénářů.
- Nezávislost na uživatelském rozhraní, které lze jednoduše měnit bez změn byznys pravidel.
- Databázi lze změnit bez nutnosti měnit byznys pravidla.

Clean architektura je uspořádána do vrstev, přičemž mechanismy (vnější kruhy) jsou konkrétnější a zásady (vnitřní kruhy) abstraktnější, viz. diagram 2.1. Pravidlo závislosti je hlavním pravidlem, které zajišťuje fungování této architektury. Toto pravidlo říká, že závislosti zdrojového kódu mohou směřovat pouze dovnitř. To znamená, že vnitřní kruhy nemohou odkazovat na názvy, funkce nebo datové formáty vytvořené ve vnějších kruzích, což zajišťuje čisté oddělení a flexibilitu.

Entita může být objekt s metodami nebo soubor datových struktur a funkcí. Žádná změna ve vrstvách aplikace by neměla ovlivnit tuto vrstvu, která zapouzdřuje ta nejobecnější a nejvýše postavená pravidla.



■ **Obrázek 2.1** Diagram reprezentující myšlenku Clean architekturu

Use Cases (česky případy použití) je vrstva, ve které jsou všechny případy užití implementovány. Tyto případy užití organizují tok dat do a z entit. Změny v této vrstvě by neměly ovlivnit vrstvu entit a změny vnějších vrstev by neměly ovlivnit Use Cases. Pokud se ovšem změny detaily případu užití, určitě to ovlivní i některý kód v této vrstvě.

The Interface Adapters vrstva převádí data mezi jádrem aplikace (entitami a případy použití) a externími systémy. Tato vrstva zachovává nezávislost logiky jádra aplikace tím, že zpracovává převod dat pro externí systémy, jako jsou databáze a externí služby, a naopak. Zároveň by žádná vnitřní vrstva neměla mít znalosti o těchto externích systémech nebo jejich specifických datových formátech.

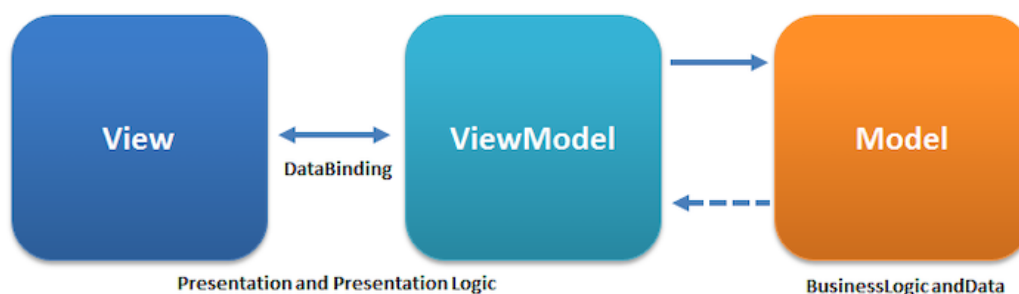
V té nejvíce vnější vrstvě se nachází Frameworky a Drivery. Obsahuje kód, který řeší detaily Web Frameworku nebo databáze, a komunikuje s vnitřní vrstvou. Právě řešení těchto detailů necháváme na vnější straně, kde mohou jen málo škodit a mohou se je kdykoliv změnit.

Žádné pravidlo neříká, že se při použití Clean architektury musí použít právě tyto čtyři vrstvy. Důležité je použití principu závislosti a že kód vnitřních vrstev nesmí záviset na kódu vnějších vrstev, závislosti směřují pouze dovnitř. Čím hlouběji se nacházíme uvnitř kruhů, tím vyšší je úroveň abstrakce. Tento přístup usnadňuje paralelní vývoj, zvyšuje modulárnost a testovatelnost systému. Clean architektura tak představuje užitečný koncept pro strukturování složitějších softwarových projektů a oddělení jejich jednotlivých částí. [3]

2.1.2 MVVM

Architektura MVVM (Model-View-ViewModel) je softwarový návrhový vzor, který ve společnosti Microsoft vytvořil Ken Cooper a Ted Peters. Tento návrhový vzor odděluje grafické uživatelské

rozhraní(View) od obchodní logiky aplikace(Model) a snaží se je mít nezávislé na sobě. [4]



■ **Obrázek 2.2** Grafické znázornění architektury MVVM [5]

V diagramu 2.2 je graficky znázorněna tato architektura. Model obsahuje datovou vrstvu aplikace a logiku pro práci s daty. View představuje grafickou část uživatelského rozhraní, která zobrazuje data a předává uživatelské vstupy. ViewModel je prostřední vrstva mezi nimi. Na jedné straně přebírá data z Modelu a na druhé straně je poskytuje View. Sleduje změny v Modelu a aktualizuje View. Naopak při uživatelských vstupech informuje Model a aktualizuje jeho stav.

Tento Model se často zaměňuje se starším návrhovým vzorem MVC (Model-View-Controller). MVVM odděluje prezentační vrstvu od logiky aplikace prostřednictvím ViewModelu, zatímco v MVC tuto roli plní Controller. ViewModel umožňuje dvoucestnou vazbu mezi View a Modelem, zatímco v MVC je komunikace jednosměrná přes Controller. MVVM se snaží minimalizovat logiku ve View a přesunout ji do ViewModelu, zatímco v MVC může View obsahovat vlastní logiku.

Architektura MVVM s sebou přináší výhody i nevýhody. Mezi výhody patří možnost paralelního vývoje prezentační a logické vrstvy oddělenými týmy, jednodušší testovatelnost a oddělení jednotlivých komponent usnadňuje čitelnost a rozšiřování funkcionalit. Mezi nevýhody architektury MVVM patří, že není vhodná pro jednoduchá uživatelská rozhraní, neboť s sebou přináší navíc komplexnost, zároveň u rozsáhlejších projektů může být náročné náležitě navrhnout ViewModel. Data binding v MVVM také komplikuje ladění ve srovnání s imperativním programováním a nelze ji kvůli tomu použít v Clean architektuře. [6]

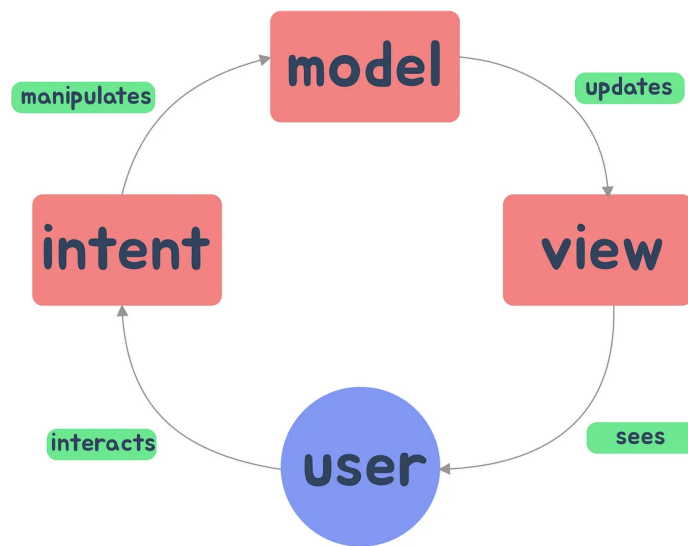
2.1.3 MVI

Architekturu MVI (Model-View-Intent) představil Hannes Dorfmann [7] a snaží se vyřešit problémy, které vývojáři mají během vytváření komplexních a reaktivních rozhraní. Cílem MVI je zajistit jednosměrný tok dat a důsledné oddělení jednotlivých komponent. [8]

Model obsahuje neměnná data a podnikovou logiku aplikace. View slouží k zobrazení aktuálního stavu, je však pasivní a neobsahuje logiku narozdíl od View v architektuře MVVM. Intent (od interakce) reprezentuje uživatelské akce, které spouštějí změny stavů v modelu.

MVI přináší několik výhod: Zajišťuje předvídatelné řízení stavu, neboť změny jsou vyvolány pouze uživatelskými akcemi. Umožňuje jednodušší testování jednotlivých komponent díky neměnnému modelu a spouštění změn přes intenty. Podporuje reaktivní programování, což usnadňuje zpracování asynchronních operací. Důsledné oddělení funkcí jednotlivých částí pak vede k lepší udržovatelnosti aplikace a nižšímu riziku chyb při úpravách. Mezi nevýhody architektury MVI patří vyšší komplexita implementace vyplývající z jednosměrného toku dat a důsledného oddělení jednotlivých komponent. Stejně tak vyšší náročnost pochopení konceptu pro nové vývojáře. [8]

Tento návrhový vzor se díky svým vlastnostem často používá právě při vytváření mobilních aplikací pro platformu Android nebo iOS. Důvody využití popsal také Apple v jedné ze svých



■ **Obrázek 2.3** Grafické znázornění architektury MVI [9]

prezentací. Data mají jediný zdroj pravdy, tok dat je jednosměrný od uživatelských akcí. Komponenty nemají vedlejší účinky a jejich výstup je pro stejný vstup stále stejný. [10]

2.2 Dependency injection

Dependency injection (česky vkládání závislostí) je softwarový design pattern, který se používá k řešení problému závislostí mezi objekty v objektově orientovaném programování. Princip spočívá v tom, že objekt neprovádí iniciaci svých závislostí sám, ale místo toho jsou mu tyto závislosti předány, například prostřednictvím konstruktoru, metody nebo rozhraní (interface). Tímto způsobem se zvyšuje modularita, testovatelnost a flexibilita kódu, protože jednotlivé komponenty jsou na sobě méně závislé. [11]

Používání závislostí je jedním z principů objektově orientovaného návrhu od Roberta C. Martina známý jako SOLID. Právě poslední písmeno je zkratka pro Dependency injection princip. Jeho podstatou je, že entity by měly záviset na abstraktních rozhraních, nikoliv na konkrétních implementacích. To znamená, že vysokoúrovňové moduly by neměly záviset na nízkoúrovňových modulech, ale místo toho by měly záviset na abstraktních rozhraních. [12]

2.2.1 Factory

Factory je bezplatný open-source nástroj, který je vhodný pro použití na platformě iOS. Jedná se o adaptabilní, výkonný a bezpečný nástroj pro správu závislostí, řídí proces vkládání závislostí pro určitý objekt nebo službu a v případě potřeby vytváří objekt požadovaného typu. Podporuje širokou škálu funkcionalit, jako jsou kontejnery, parametry, unit testy nebo náhled ve SwiftUI.

Definice závislostí vyžadují pouze minimum kódu, což přispívá k celkové stručnosti a flexibilitě řešení. Factory je kompatibilní s různými frameworky, od UIKit po SwiftUI, a je vhodný pro použití v projektech s různými přístupy, jako jsou MVVM, MVI nebo Clean architektura.

Factory má rozsáhlou dokumentaci a příklady, které pokrývají všechny třídy, metody a případy použití nástroje. Navzdory bohaté funkcionalitě má Factory méně než 800 řádků spustitelného kódu. [13] [14]

2.3 Frameworky pro vývoj multiplatformních aplikací

Výběr správného nástroje pro vývoj multiplatformní aplikace je důležitým krokem před započtím návrhu a implementace. Proto se tato podkapitola zaměřuje na popsání některých z nejpopulárnějších frameworků pro vývoj multiplatformních aplikací.

2.3.1 Flutter

Flutter je open-source framework pro vytváření multiplatformních aplikací, který společnost Google vydala v roce 2017. Umožňuje vytvářet nativní aplikace pro Android i iOS z jednoho kódového základu pomocí programovacího jazyka Dart.

Jedním z hlavních benefitů je funkce Hot Reload, která pomáhá při zobrazování UI bez nutnosti kompilace při každé změně. Dále Flutter usnadňuje vývoj aplikací díky sadě připravených widgetů, které zastupují běžné funkce jako mapy, posuvné seznamy a navigace. Jedná se o komplexní widgety, díky kterým není nutné psát kód pro základní prvky uživatelského rozhraní, což ušetří čas na implementaci jiných částí v kódu. [15]

Velmi zajímavým nástrojem je Flutter Flow, který umožňuje vizuální návrh uživatelského rozhraní aplikací pomocí přetažením a pokládáním prvků, aniž by bylo nutné psát kód. Poskytuje grafické rozhraní pro design aplikace, které je automaticky převedeno do kódu Flutter. Flutter využívá grafický engine Skia, který zajišťuje hardwarově akcelerované vykreslování všech pixelů na obrazovce, díky čemuž umožňuje plynulé animace 60 snímků za sekundu i na zařízeních s nižším výkonem. K frameworku Flutter rovněž patří rozsáhlá komunita vývojářů, kteří přispívají pluginy a knihovnamí. [16]

Mezi nevýhody Flutteru patří omezený počet knihoven třetích stran v porovnání s frameworkem React Native, strmá křivka učení se nového programovacího jazyku Dart a také zatím nižší využití ve velkých korporacích než u zavedenějších technologií pro vývoj multiplatformních aplikací. [17]

2.3.1.1 Dart

Dart je klientsky orientovaný programovací jazyk vyvinutý firmou Google pro vývoj multiplatformních aplikací. Cílem Dartu je nabídnout nejproduktivnější jazyk pro multiplatformní vývoj.

Hlavním principem Dartu je, že podporuje nulovou bezpečnost nulových hodnot, takže proměnné nemohou nabývat hodnoty null, pokud to není explicitně povoleno. Kód je velmi čitelný díky podobnosti se strukturovanými jazyky jako Java nebo C#. Dart podporuje funkcionální programování, ale jeho hlavní paradigma je objektové orientovaná.

Dart je multiplatformní, tzn. že kód lze překládat do nativního kódu pro Android, iOS, web i desktop. To je jeden z důvodů, proč je Dart používán ve flutter frameworku pro vývoj multiplatformních aplikací. [18]

2.3.2 React Native

React Native je open-source UI framework, který byl představen v roce 2015 společností Meta a je založen na JavaScript knihovně React. Poskytuje základní strukturu aplikace a obsahuje připravené komponenty, knihovny a referenční materiály pro rychlejší implementaci.

Aplikace vytvořené v React Native působí jako klasické nativní aplikace. Přestože není kód psán přímo pro konkrétní platformu, je renderován do nativních komponent daného systému. To umožňuje lepší výkon a zážitek uživatele, neboť aplikace jsou optimalizovány pro dané zařízení. [19]

React Native nabízí podobně jako Flutter rychlé aktualizace kódu (Fast Refresh), díky kterým vidí vývojáři změny v komponentách okamžitě. Další výhodou je zaměření na uživatelské

rozhraní, kdy React primitiva renderují do nativních UI komponent dané platformy, což umožňuje vytvářet přizpůsobivé uživatelské rozhraní. Od verze 0.62 je integrován debugger Flipper pro ladění mobilních aplikací včetně React Native, poskytující nástroje jako prohlížeč logů, inspektor rozložení a síťový inspektor. [20]

Díky svým schopnostem se využívá pro vytváření multiplatformních aplikací. Umožňuje vytvářet specifické verze komponent pro jednotlivé platformy, a tak lze kód vyvíjet jednotně pro Android i iOS. Díky tomu snižuje náklady a šetří čas na vývoj.

Mezi nevýhody použití React Native patří slabší výkon aplikací než u plně nativních aplikací, protože nemůže využít veškeré možnosti konkrétní platformy. Dále není efektivní pro složitá uživatelská rozhraní s pokročilými interakcemi, kde by nativní vývoj přinesl lepší výsledky. Aktualizace na novější verze React Native bývají komplikované. Nicméně vývojáři se snaží zvyšovat výkon, například novým JS enginem Hermes pro starší Android zařízení. [21]

2.3.2.1 ReactJS

React nebo také ReactJS je open-source front-end JavaScript knihovna pro vytváření komponent pro uživatelské rozhraní.

React je jednoduchý na použití a s mírnými znalostmi JavaScriptu si ho snadno osvojí i začínající vývojáři. Zjednodušuje práci díky modulární struktuře, která urychluje vývoj a údržbu aplikací a umožňuje opakované využití kódu. [16]

Pokud nemá vývojář základní znalosti JavaScriptu, může být pro něj používání ReactJS náročnější. React dále není tolik vybavený framework, který doplňují knihovny. Na starších zařízeních může být výkon pomalejší z důsledku klientského renderování a náročného zpracování JavaScriptu. [22]

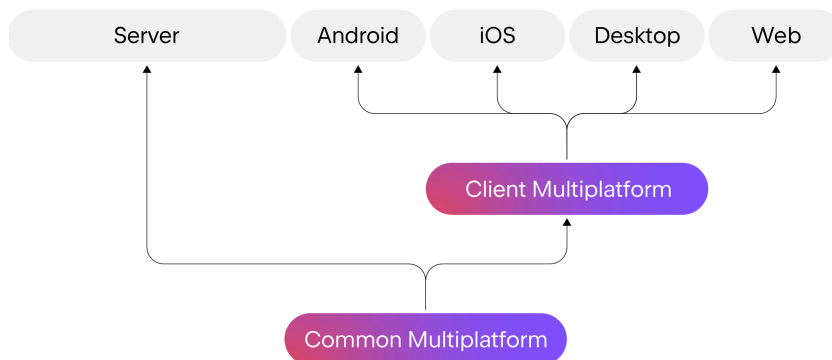
2.3.3 Kotlin Multiplatform

Kotlin Multiplatform(KMP) je open-source framework pro vytváření multiplatformních aplikací vytvořený společností JetBrains. Přináší možnost sdílení kódu mezi platformami na takové úrovni, jak si vývojáři přejí, a jednoduché integrace do projektů. Používá programovací jazyk Kotlin. [20] Jedná se o jeden z nejmladších frameworků používaný pro multiplatformní vývoj.

KMP umožňuje opakované využití kódu mezi platformami Android, iOS, webem, desktopem i serverovou částí při zachování nativního kódu tam, kde je potřeba. Jednou z klíčových výhod KMP je použití programovacího jazyka Kotlin, který představuje moderní a expresivní jazyk, jenž si získal značnou popularitu zejména v komunitě vývojářů pro Android. Týmům, které již jazyk Kotlin ovládají, umožňuje KMP hladký přechod a vývojářům efektivně využít jejich stávající dovednosti a znalosti získané z programování v Kotlinu.

Spolu s KMP lze využít i UI framework Compose Multiplatform, který poskytuje plnou sdílenost kódu pro logiku i uživatelské rozhraní. Nevyžaduje přidání nového programovacího jazyka do projektů, které již používají Kotlin pro Android. [24] Uživatelské prostředí lze vytvořit jednou a použít ho na platformách Android, iOS, web a desktop. Dále se vyznačuje bohatým ekosystémem ověřených knihoven a frameworků pro rychlý start a snazší vývoj. Nabízí hladkou integraci se specifickými funkcemi jednotlivých platform stejně jako u nativních aplikací. V neposlední řadě umožňuje opakované využití komponent napříč platformami, včetně připravených motivů a nástrojů pro vlastní vizuální stylizaci. [25]

Výhod, které KMP přináší, je celá řada. Klíčovou předností je efektivní sdílení kódu mezi platformami Android a iOS, díky čemuž dochází k omezení duplicity kódu a zajištění jeho konzistence. Následný překlad do nativních reprezentací jednotlivých systémů umožňuje optimalizaci výkonu aplikací. Pro programovací jazyk Kotlin se nabízí široká škála knihoven pro usnadnění vývoje. Možnost využít multiplatformní framework pro tvorbu UI Compose Multiplatform nebo využít nativní řešení pro dané platformy je výhodou, která dělá z KMP flexibilní a snadno škálovatelný nástroj.



■ Obrázek 2.4 Kotlin Multiplatform [23]

Nevýhodou KMP je samotné použití programovacího jazyka Kotlin. Přestože je to velká výhoda pro ty, kteří se s ním setkali, pro vývojáře, kteří se s Kotlinem nikdy nepotkali, to může být velkou nevýhodou a vyžaduje časovou investici do seznámení se s jeho syntaxí a prvky. Protože je KMP novým frameworkem, může být dostupnost komplexní dokumentace a nástrojů řešení nižší než u zavedenějších nástrojů. [26]

2.3.3.1 Kotlin

Kotlin je open-source, staticky typovaný programovací jazyk, který podporuje zároveň funkcionální i objektově-orientované programování. Spravuje jej Kotlin Foundation, skupina založená JetBrainsem a Googlem, jejímž úkolem je rozvoj a udržování jazyka. Společnost Google oficiálně Kotlin podporuje pro vývoj aplikací pro platformu Android.

Jazyk Kotlin přejímá syntaxi a koncepty z jazyků jako C#, Java nebo Scala. Neusiluje o unikátnost, ale čerpá inspiraci z dlouhodobého vývoje jazyků. Existuje ve variantách pro frameworky JVM(Kotlin/JVM), JavaScript(Kotlin/JS) i nativní kód(Kotlin/Native). Mezijazyková kompatibilita s Javou je klíčová pro rozšíření Kotlinu, umožňuje volání kódu z jednoho jazyka do druhého a využívání stávajících Java knihoven. Mezijazyková kompatibilita také znamená, že není nutné přecházet na Kotlin naráz. Projekty mohou obsahovat kód v Kotlinu i Javě. [27]

2.4 Firestore

Firestore je vývojářská platforma od společnosti Google poskytující řadu služeb pro vývoj mobilních a webových aplikací. Nabízená řešení pokrývají celý proces vývoje aplikace. Mezi známé služby spadající do této platformy patří Firestore, Firestore Authentication, Firestore Hosting nebo Firestore Storage. Dále Firestore nabízí služby pro analýzu a testování, jako je například Firestore Analytics, Firestore Crashlytics nebo Firestore Performance Monitoring.

Pro vývoj multiplatformních aplikací nabízí Firestore snadnou integraci na platformy iOS, Android, Flutter, Unity nebo C++. Proto je Firestore oblíbená platforma při vytváření mobilních aplikací, protože umožňuje vývojářům soustředit se na budování funkcionalit a uživatelského rozhraní, zatímco základní infrastrukturní prvky, jako jsou databáze, autentizace nebo hosting, jsou zajištěny platformou Firestore. [28]

2.4.1 Firebase Firestore

Firebase Firestore je NoSQL cloud databáze, která je součástí platformy Firebase a Google Cloud. Tato služba udržuje data synchronizovaná napříč klientskými aplikacemi a nabízí offline podporu pro mobilní zařízení bez ohledu na latenci sítě nebo připojení k internetu. Toho dosahuje, neboť ukládá do mezipaměti data, která aplikace aktivně používá. Aplikace může zapisovat, číst a dotazovat se na data, i když je zařízení offline. Když se zařízení vrátí do režimu online, synchronizují se veškeré místní změny zpět do úložiště Firebase Firestore.

Aplikace mohou k databázi přistupovat pomocí nativní SDK. Databáze podporuje mnoho datových typů a umožňuje vytvářet hierarchické datové struktury. Umožňuje provádět dílčí dotazy na úrovni dokumentů, jako například řadit nebo filtrovat příchozí data.

Firebase Firestore umožňuje propojení s dalšími službami Firebase, jako například s Firebase Storage. [29]

2.4.2 Firebase Storage

Cloudové úložiště Firebase Storage je podobně jako Firebase Firestore postaveno na infrastruktuře Google Cloud. Umožňuje ukládání souborů, jako jsou například obrázky a videa. Umožňuje vytvářet odkazy na nahrané soubory pro snadné stahování a používat API služby Firestore Storage pro přístup k souborům. [30]

Při kombinaci Firebase Firestore a Firebase Storage se ukládají reference na soubory fyzicky umístěné v úložišti Firebase Storage do databáze Firestore.

2.4.3 Firebase Authentication

Firebase Authentication poskytuje backendové služby pro ověřování uživatelů aplikace. Nabízí snadno použitelná SDK a připravené knihovny uživatelského rozhraní. Podporuje ověřování pomocí hesel, telefonních čísel a populárních poskytovatelů, jako jsou Google, Facebook a Twitter. Po vyplnění přihlašovacích údajů je Firebase Authentication ověří a vrátí klientovi odpověď. Po úspěšném přihlášení má aplikace přístup k základním informacím o uživatelském profilu a může řídit přístup k datům uloženým v dalších službách Firebase. [31]

Díky přihlášení ke svému profilu mohou uživatelé získat personalizovaný obsah a bezpečně ukládat a spravovat osobní uživatelská data, ke kterým nemá nikdo jiný přístup.

2.5 Jetpack Compose

Jetpack Compose je oficiálně doporučeným nástrojem pro vývoj nativního uživatelského rozhraní v systému Android. Umožňuje deklarativní popis uživatelského rozhraní, kdy jeho jednotlivé komponenty lze definovat voláním funkcí bez nutnosti psát tradiční imperativní kód. Tento přístup usnadňuje a urychluje vývoj díky menšímu množství psaného kódu a intuitivnímu rozhraní v jazyce Kotlin. Jetpack Compose nabízí výkonné nástroje pro tvorbu uživatelského rozhraní a alternativu k dosavadnímu imperativnímu způsobu programování uživatelského rozhraní v systému Android. [32]

Jetpack Compose umožňuje vytvářet uživatelské rozhraní s menším množstvím kódu ve srovnání s tradičním přístupem v systému Android. To ovlivňuje všechny fáze vývoje - vývojář se může soustředit na řešení problému, testování a ladění vyžadují méně času a klesá pravděpodobnost chyb. Zároveň Jetpack Compose nabízí jednodušší způsob vytváření běžných prvků uživatelského rozhraní, jako jsou tlačítka, seznamy či animace.

Jetpack Compose je kompatibilní s existujícím kódem v Android aplikacích. Compose lze volat z Views a naopak, tedy že mezi těmito dvěma technologiemi je zajištěna vzájemná provázanost. Většina běžně používaných knihoven jako je navigace, view model nebo Kotlin coroutines fungují

s Compose bez problémů. To umožňuje postupně přecházet na knihovnu Compose tam, kde je to vhodné. [33]

2.6 Multiplatform Settings

Multiplatform Settings je knihovna programovacího jazyka Kotlin, která umožňuje v multiplatformních aplikacích uchovávat data, která se skládají z hodnoty a klíče. Knihovna nabízí jednoduchou konfiguraci ve sdílené části kódu a snadné používání. Poskytuje sdílené API na podporových platformách, jako je například iOS, Android, macOS, JS, JVM, nebo Windows. [34]

Díky podpoře široké škály platform, snadné konfiguraci a intuitivnímu ovládní se jedná o cenný nástroj, který se hojně využívá pro vývoj multiplatformních aplikací s jednotnou správou lokálních dat.

2.7 SQLDelight a SQLite

SQLDelight je nástroj, který z SQL příkazů generuje bezpečné rozhraní API pro programovací jazyk Kotlin. Toto řešení ověřuje správnost vašeho databázového schématu, SQL příkazů a migračních skriptů již v době kompilace. Zároveň poskytuje vývojářům funkce integrovaného vývojového prostředí, jako je automatické doplňování kódu a refaktoring, které usnadňují psaní a údržbu SQL dotazů. [35]

V případě multiplatformní aplikace podporuje relační databázi SQLite. SQLite je databázový engine, který čte a zapisuje data přímo do jednoho souboru. Proto se s oblibou používá při vývoji aplikací. Má poměrně nízké paměťové nároky a proto je poměrně rychlá, je pečlivě testována a má pověst spolehlivého řešení. [36]

2.8 Swift a Swift UI

Swift je výkonný a intuitivní programovací jazyk pro všechny platformy společnosti Apple. Byl představen na konferenci WWDC v roce 2014 a nahradil používaný programovací jazyk Objective-C s cílem usnadnit vývoj aplikací a snížit výskyt chyb.[37]

Swift je založen na programovacím jazyku C, Objective-C a C++. Snaží se využívat výhody těchto jazyků, jako je například rychlost a efektivnost. Protože se jedná o relativně nový jazyk, snaží se přinést moderní přístup k programování a vyvíjení aplikací, aby byl kód čistý a snadno čitelný. Swift také používá moderní techniky práce s pamětí, řetězci a souběžností, takže je výkonný a flexibilní pro různé aplikace. Snaží se i o bezpečnost kódu tím, že vynucuje inicializaci, kontroluje přetečení dat v paměti, automaticky spravuje paměť a omezuje přístup do paměti. [38]

SwiftUI je framework, pomocí kterého mohou vývojáři vyvíjet rozhraní pro zařízení, které patří do ekosystému Apple. Tento framework je postaven na programovacím jazyku Swift. Hlavní výhodou je funkce Live Preview, která vývojářům umožňuje zobrazit rozhraní aplikace, aniž by museli aplikaci kompilovat a sestavit. Kód je při psaní okamžitě viditelný jako náhled a uživatelské rozhraní můžete dokonce zobrazit ve více konfiguracích, například ve světlém a tmavém vzhledu.

SwiftUI je dostupné ve vývojovém nástroji XCode a může se použít s původním nástrojem na vytváření uživatelského prostředí UIKit a AppKit. Tím umožňuje postupný přechod u střeších aplikací, které používají UIKit nebo AppKit, na SwiftUI. SwiftUI je nový nástroj, který se s každou novou verzí zlepšuje, nemusí ovšem nabízet prvek rozhraní, který vývojář požaduje. Proto je vhodné kombinovat SwiftUI s UIKitem a AppKitem a využít tak to nejlepší ze všech světů. [39]

2.9 Twine

Twine je nástroj příkazového řádku určený pro správu řetězců a jejich překladů. Veškeré řetězce jsou uchovávány v jediném textovém souboru, který Twine následně využívá k importu a exportu lokalizačních souborů v různých formátech. Mezi podporované formáty patří soubory `.strings` pro iOS a macOS, soubory `.xml` pro Android a soubory `jquery-localize.json`.

Toto řešení umožňuje snadno vytvářet a sdílet překlady v rámci více projektů a exportovat lokalizační soubory v libovolném formátu dle potřeb. Díky tomu je možné efektivně spravovat a distribuovat překlady napříč různými platformami a vývojovými prostředími. [40]

2.10 XCTest

XCTest je robustní testovací framework integrovaný přímo do vývojového prostředí Xcode, které je oficiálním nástrojem pro testování aplikací na platformách Apple, jako jsou iOS, macOS, watchOS a tvOS. Tento framework umožňuje vývojářům vytvářet a spouštět různé typy testů, které ověřují funkčnost, stabilitu a výkonnost jejich aplikací.

Mezi hlavní typy testů podporované XTestem patří jednotkové testy pro izolované testování komponent kódu, UI testy pro ověřování uživatelského rozhraní a performance testy pro měření výkonnosti aplikace. XCTest nabízí širokou škálu pokročilých funkcí, jako je asynchronní testování, mocking pro simulaci závislostí, robustní aserce pro kontrolu očekávaných výsledků a generování podrobných testovacích reportů. Díky této komplexnosti a integraci do Xcode je XCTest nezbytným nástrojem pro vývojáře, kteří chtějí zajistit vysokou kvalitu a spolehlivost svých aplikací na platformách Apple. [41]

Kapitola 3

Analýza

V této kapitole je popsána analytická část bakalářské práce. Obsahuje analýzu současného webového řešení a porovnává současná řešení na poli mobilních aplikací. Díky těmto informacím jsou identifikovány důležité funkcionality aplikace a vytvořeny funkční a nefunkční požadavky, případy užití a scénáře případů užití.

3.1 Analýza současného webového řešení

V této části bakalářské práce je provedena analýza webové platformy XContest. Cílem analýzy je získání přehledu o fungování webu, co přináší pro uživatele a nalezení silných a slabých stránek.

3.1.1 Představení současného webového řešení

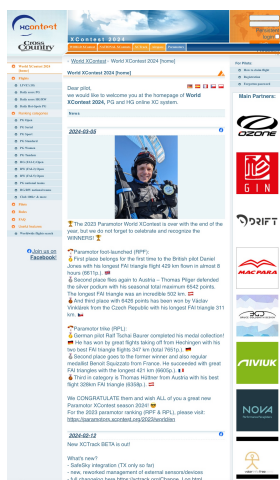
XContest je letový deník pro piloty paraglidingu po celém světě, kteří chtějí porovnávat své výkony s ostatními. Umožňuje zaznamenávat přelety pilotů, prohlížet si přelety ostatních letců a porovnávat je s těmi svými. Je to encyklopedie letových tras, do které mají všichni přístup a mohou si ji prohlížet. Spoléhá se na tvůrce třetích stran, kteří se snaží integrovat XContest do svých vlastních aplikací, aby měli možnost snadného odeslání svých zaznamenaných přeletů na servery XContest. Nejvíce známou aplikací pro záznam přeletu, která je platformou XContest propagována, je aplikace XCTrack. Ta přemění mobilní telefon s operačním systémem Android na přenosný letový počítač, který umožňuje zaznamenaný let nahrát na XContest. [42]

3.1.2 Vzhled a ovládání

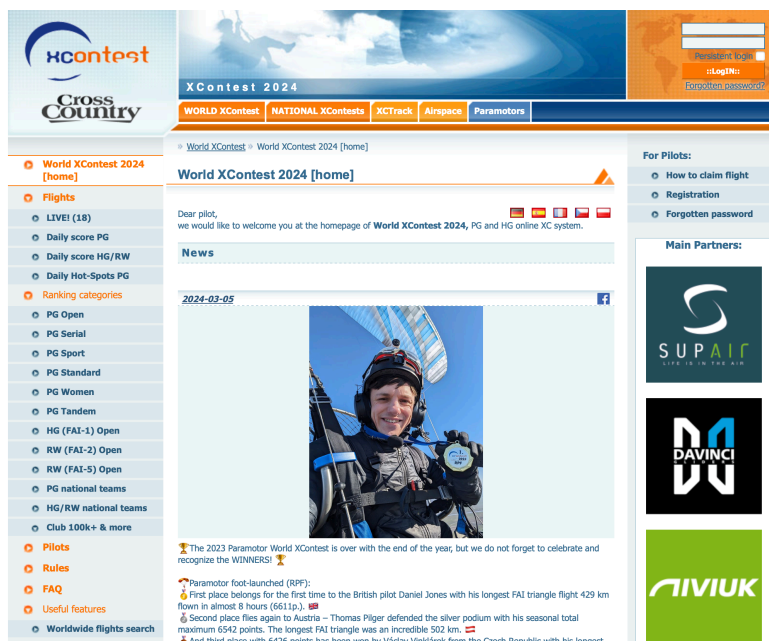
XContest používá jednoduchý design a uživatelské rozhraní, které cílí na maximální komfort a efektivitu při interakci uživatelů. V současné době může působit zastarale a nepřívětivě. Dominantním prvkem stránky je boční menu na levé straně stránky a horní panel, kde je logo XContest, lišta s několika záložkami a formulář pro přihlašování. Barevná paleta webu není pestrá a reflektuje barvy loga XContest. Střídají se zde tóny modré a pro zvýraznění se využívá oranžová barva.

Struktura webu je složitá a může nově přichozím uživatelům působit potíže. Horní lišta se záložkami prepíná uživatele, mezi jednotlivými soutěžemi, které na XContest probíhají. Menu, které je na levé straně stránky, umožňuje přístup k různým sekcím, kategoriím a informacím, které se týkají soutěže. Na pravé straně je menší menu, které přináší rychlý přístup k sekcím, které se týkají profilu uživatele a letového deníku.

Obsah webu je srozumitelný a přímočarý. Texty jsou psány s ohledem na cílovou skupinu a doplněny o relevantní obrázky, tabulky nebo interaktivní mapy. Díky pevné struktuře a přehlednému uspořádání informací usnadňuje web uživatelům orientaci a umožňuje jim najít vše, co potřebují.



■ Obrázek 3.1 Snímek obrazovky webu XContest na mobilním zařízení [43]



■ Obrázek 3.2 Snímek obrazovky webu XContest na počítači [43]

Web bohužel není responzivní, a proto se nemusí správně zobrazovat na mobilních zařízeních a tabletech. Uživatelé si musejí stránku přibližovat a posouvat, aby se k požadovanému obsahu dostali. To může vést k frustraci a zhoršení uživatelského zážitku. Jak se zobrazuje XContest na mobilních zařízeních lze vidět na obrázku 3.1. Na obrázku 3.2 se zobrazuje stejná stránka na počítači.

3.1.3 Fungování webového řešení

Platforma XContest nabízí řadu možností pro své uživatele. Jednotlivé funkcionality lze rozdělit na ty, které jsou dostupné nepřihlášeným i přihlášeným uživatelům a na ty, které jsou dostupné pouze přihlášeným uživatelům.

3.1.3.1 Nepřihlášení uživatelé

Služba XContest přináší řadu funkcionalit pro nepřihlášené uživatele. Funkcionality, které budou v této části popsány, jsou:

- Výběr soutěží
- Zobrazení novinek
- Živé sledování přeletů
- Prohlížení soutěží, pilotů a jejich přeletů

3.1.3.1.1 Výběr soutěží Jako první funkcionalita, která je potřeba zmínit, je výběr soutěží. Na platformě XContest probíhá světová soutěž **WORLD XContest**, národní soutěže (např. **Český pohár paraglidingu - ČPP**), regionální soutěže a klubové soutěže. Dále se zde konají závody, do kterých se mohou piloti přihlásit a účastnit se jich v daný okamžik konání. Ukázkou jednoho z plánovaných závodů lze vidět na tomto obrázku 3.3.



■ **Obrázek 3.3** Ukáзка závodu UP Krkonoše Open 2024 [44]

Ačkoliv se XContest zaměřuje na soutěže (anglicky contests), je důležité je vnímat spíše jako samostatné weby pod hlavičkou XContestu. Kromě světové a národních soutěží si každá další soutěž může vytvořit vlastní pravidla bodování letů. Na webu dané soutěže pak můžete sledovat pouze piloty a jejich přelety, kteří se do dané soutěže přihlásili. Každá soutěž může mít několik kategorií pro zobrazení bodování a může vyhlášovat vlastní závody. Pod národní soutěže se mohou přidávat i přidružené soutěže.

The screenshot displays the 'Detail přeletu' page for Moritz Kampelmühler on 20.3.2024. The flight details are as follows:

pilot:	Moritz Kampelmühler	AT
datum:	20.03.2024 11:36	success
startovníte:	Schockl	AT
trati:	62.22 km	87.10 b.
kluzák:	SKYWALK X-Alps 5	
trvání:	3:31 h	18.23 km/h

The map shows the flight path starting from Schockl, passing through various points like Frohalden, Pöchlarn, and ending at Mitterdorf. A profile graph below the map shows altitude changes over time. The 'Spoluletci' table lists other pilots who started from the same location:

Rank	Date	Pilot	Country	Distance	Score
1	20.03.24 12:31	Sandro Traussnig	AT	25.89 km	13.13
2	20.03.24 12:03	Willi Schrottnig	AT	23.19 km	10.07
3	20.03.24 12:32	Ronald Walli	AT	22.12 km	13.70

■ Obrázek 3.4 Ukázka zobrazení detailu přeletu na XContest [45]

3.1.3.1.2 Zobrazení novinek Domovská stránka představuje pro každou soutěž klíčový komunikační kanál. Na domovských stránkách Světové soutěže a národních soutěží jsou zveřejňovány novinky, které byly dříve prezentovány na facebookových stránkách dané soutěže. Tyto novinky informují účastníky o aktuálním dění v soutěži, o důležitých termínech a jak se do soutěže přihlásit. Na obrázku 3.2 je ukázána úvodní stránka XContest, na které se zobrazují novinky ve světové soutěži World XContest.

3.1.3.1.3 Živé sledování přeletů XContest umožňuje sledovat piloty paraglidingu během jejich letu. Tato funkcionality má široké spektrum využití. Může být nápomocná při hledání pilotů, kteří se nacházejí v oblasti kolem vás, nebo to usnadňuje sledování pilotů během závodu. Dále umožňuje uživatelům sledovat, kde se nacházejí jejich blízcí, kteří se věnují paraglidingu. Pro využití této funkce je nutné, aby piloti používali aplikace, které tuto službu podporují. XContest sám o sobě nenabízí monitoring letu, ale propojuje se s kompatibilními aplikacemi, které tuto funkci zajišťují.

3.1.3.1.4 Prohlížení soutěží, pilotů a jejich přeletů Kdokoliv po celém světě má možnost si prohlédnout žebříčky hodnocení soutěží. Nabízí se možnost přelety filtrovat podle několika různých parametrů, jako je datum letu, země přeletu, kategorie a další. Kliknutím na jméno pilota se zobrazí jeho profil, včetně jeho statistik a přeletů, které přihlásil do soutěže. Detail přeletu můžete vidět na obrázku 3.4, který zobrazí detailnější informace o jeho průběhu, interaktivní mapu i například spoluletce, kteří ze stejného stanoviště a v podobný čas odstartovali.

3.1.3.2 Registrace a přihlašování

Aby se mohl nepřihlášený uživatel stát přihlášeným, musí vyplnit přihlašovací formulář a mít zaregistrovaný účet na XContest. Přihlašování je velice jednoduché. V pravém horním rohu se nachází přihlašovací formulář, kde uživatel vyplní uživatelské jméno a heslo a stiskne tlačítko *Přihlásit se*.

Registraci zahájí uživatel tím, že stiskne v menu na pravé straně stránky položku *Registrace*. Webová stránka zobrazí registrační formulář. Ten obsahuje několik sekcí. V sekci *Identita* se vytvoří uživatelské jméno a heslo. V sekci *Další údaje* se vyplňují údaje, které se budou zobrazovat ostatním pilotům, jako je jméno a příjmení, datum narození, a nebo číslo pilotního průkazu. Číslo pilotního průkazu není u světové soutěže povinné, ale u některých soutěží (např. ČPP) je vyžadováno. Další sekci je *Nastavení*, kde uživatel nastavuje posílání oznámení na e-mail nebo svoji fotografii. V neposlední řadě v sekci *Přihlášení do soutěže* uživatel potvrzuje, které soutěže se chce zúčastnit. Po úspěšné registraci XContest odešle potvrzovací e-mail a uživatel se může přihlásit.

3.1.3.3 Přihlášení uživatele

Přihlášení uživatele mají přístup ke všem funkcionalitám, které jsou popsány výše pro nepřihlášené uživatele. Vzhled XContest se pro přihlášené uživatele výrazně nemění. Na pravé straně webové stránky naleznou **Moje menu** a pod ním položky *Můj profil*, *Přihlásit přelet*, *Moje přelety* a *Soukromé zprávy*. Tyto položky otvírají pro přihlášené uživatele nové možnosti, jak XContest používat, jako například:

- Správa profilu pilota
- Správa letového deníku
- Vytvářet nebo se přidávat do týmů
- Komentování přeletů ostatních pilotů
- Posílání soukromých zpráv

3.1.3.3.1 Správa profilu pilota Přihlášený uživatel má na XContest vytvořený profil, který si může kdokoli prohlédnout. Profil obsahuje základní informace, jako je jméno, příjmení, jaký kluzák používá, statistiky z přeletů, profilovou fotku nebo i pořadí v kategoriích soutěže. Přihlášený uživatel může své profilové informace upravovat, když otevře sekci *Můj profil* v menu. Jaké konkrétní informace může uživatel měnit ukazuje obrázek 3.5. Jsou to stejné informace, které uživatel vyplňuje během registrace.

3.1.3.3.2 Správa letového deníku Jednou z nejdůležitějších funkcí XContest je správa letového deníku. Následující odstavce se věnují přidávání nových přeletů a editaci přeletů v letovém deníku.

V pravém menu naleznete položku *Přihlásit přelet*. Přihlašování přeletu se skládá ze dvou kroků. Prvním krokem je zpracování GPS tracklogu ve formátu IGC. Ten pilot získá ze svých vario-přístrojů, nebo mobilních aplikací, které umí ze zaznamenaného letu vytvořit GPS tracklog ve formátu IGC. Po odeslání na server se soubor zkontroluje a projde validací G-recordu. Pokud je soubor v pořádku, je zařazen do fronty k optimalizaci trati a zobrazí se formulář pro zadání zbývajících údajů (to bude druhý krok). V případě odmítnutí IGC souboru se zobrazí chybové hlášení s důvodem odmítnutí. V druhém kroku se zobrazí formulář, který přihlášený uživatel vyplní a potvrdí. Formulář obsahuje základní údaje, jako je stanoviště, použitý kluzák a stát. Dále lze upravit čas, kdy pilot vzlétl a přistál, přidat fotky a okomentovat let. V neposlední řadě lze vybrat soutěže, do kterých chce pilot přelet přidat.

Vedle toho lze přihlásit přelet například pomocí mobilní aplikace XTrack, která umožňuje propojení s účtem XContest. Protože aplikace nabízí zaznamenávání letu a funguje jako letový počítač, výrazně zjednodušuje přihlašování přeletu.

Další položkou v pravém menu je položka *Moje přelety*. Ta zobrazí seznam všech přeletů, které uživatel přihlásil k soutěži. Uživatel má následující možnosti práce s přelety:

■ Obrázek 3.5 Ukázka sekce Můj profil na XContest [46]

- zobrazit detail přeletu,
- upravit informace o přeletu,
- stáhnout přelet ze soutěže,
- odstranit přelet ze všech soutěží.

Informace, které může pilot po přihlášení přeletu měnit, jsou stejné, které vyplňoval do formuláře během přihlašování přeletu. Přelet, který byl stažen ze soutěže, se zobrazuje s poloviční průhledností a nezapočítává se do bodového hodnocení. Uživatel ho však může znovu zveřejnit, pokud není starší 14 dní. Po uplynutí této lhůty se let zobrazí pouze soukromě, pokud byl přihlášen z aplikace XCTrack nebo jí podobné. Toto pravidlo je nastaveno z důvodu férovosti, aby se pořadí pilotů jednotlivých kategorií dramaticky neměnilo těsně před koncem ročníku. Odstranění přeletu ze všech soutěží je nevratné a povede k smazání z celého XContest.

3.1.3.3.3 Vytváření nebo přidávání se do týmů Piloti nejen soutěží na XContestu jako jednotlivci, ale mohou také vytvářet tříčlenné týmy a soutěžit v rámci týmových soutěží proti ostatním týmům. Body získané za přelety se členům v týmu sčítají. Každý přihlášený uživatel má možnost si vytvořit svůj vlastní tým v ČPP, přidat piloty do týmu, změnit vedoucího týmu, nebo zrušit tým a odstranit ho z XContest. Členové týmu se dají změnit dle potřeby.

3.1.3.3.4 Komentování přeletů a zaslání soukromých zpráv Přihlášení uživatelé mohou komentovat přelety ostatních letců. V detailu přeletu se nachází sekce *Diskuze přeletu*, kde se zobrazují komentáře k přeletu a je zde formulář pro napsání nového příspěvku. Každý příspěvek obsahuje titulky, přezdívku autora a samotný text příspěvku.

Uživatelé si mohou mezi sebou zasílat soukromé zprávy. Pokud chce přihlášený uživatel odeslat soukromou zprávu a ještě si s dotyčným uživatelem nikdy nepsal, musí najít profil pilota. Pod základními informacemi o pilotovi se nachází odkaz *Zaslat soukromou zprávu*. Po stisknutí se uživateli zobrazí formulář podobný tomu, který je u diskuze přeletu. Odeslaná soukromá zpráva je viditelná pouze adresátovi a uživateli, který zprávu poslal. Je také ihned zaslána adresátovi prostřednictvím e-mailu.

3.1.4 Shrnutí analýzy současného webového řešení

Webovou platformu XContest využívá po celém světě několik desítek tisíc pilotů. Cílem analýzy bylo získání přehledu o fungování webu, co přináší pro uživatele a nalezení silných a slabých stránek.

XContest je robustní platforma, která nabízí unikátní možnosti, které nemají na světě obdoby. Mezi její stěžejní funkce patří vedení letového deníku a účastnění se v leteckých soutěžích. Mohou ho používat i nepřihlášení uživatelé, kteří mají přístup k žebříčkům soutěží a vyhledávání pilotů po celém světě. Zároveň vytváří a podporuje komunitu pilotů. Sami piloti se podílí na jejím vývoji, organizují soutěže a udržují vzájemný kontakt. Tyto klíčové vlastnosti stojí za tím, proč je XContest tak populární.

Mezi slabé stránky se jistě řadí vzhled webu, který působí zastarale, a není responzivní. To způsobuje nevyužití plné šíře obrazovky na počítači, nebo špatně ovladatelné zobrazení na mobilním telefonu. XContest nabízí spoustu funkcí a možností použití, jeho fungování a ovládání nemusí být pro nově příchozí uživatele intuitivní. V neposlední řadě je zde absence vlastního řešení pro snadné nahrávání letů a spoléhání na aplikace do jiných vývojářů. Výjimkou je aplikace XCTrack, která je XContest doporučována, ale je dostupná pouze na mobilních zařízeních s operačním systémem android, nenabízí zobrazení soutěží z XContest a letový deník. Slouží primárně jako letový počítač pro paraglidisty.

3.2 Alternativní fungující mobilní aplikace

V obchodech s mobilními aplikacemi nalezneme mnoho aplikací, které umožňují vedení letového deníku pro paraglidisty. Každá z nich se liší v přístupu k řešení daného problému a má své výhody a nevýhody. V této části bakalářské práce jsou některé z nich ukázány a ve zkratce popsány. Aplikace jsou vybírány na základě doporučení paraglidistů, jejich funkčnosti a rozmanitosti v přístupu k řešení daného problému.

3.2.1 Flyskyhy

Mezi jednu z nejznámějších aplikací patří aplikace Flyskyhy, která je dostupná v obchodě App Store na platformě iOS za 249 Kč a přináší celou řadu zajímavých funkcí. Na obrázku 3.6 je ukázán vzhled aplikace a některé z jeho funkcí.



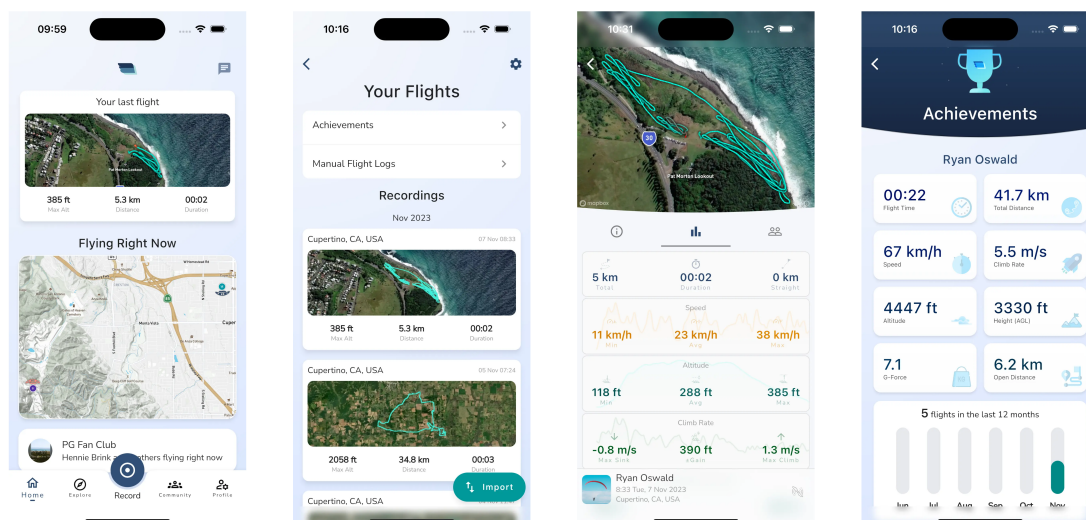
■ Obrázek 3.6 Ukázka aplikace Flyskyhy [47]

Letový deník, který nabízí aplikace Flyskyhy, je jednoduchý, funkční, nabízí zobrazení letů v určitém roce a možnost filtrování podle vzdálenosti, délky letu nebo způsobu vzletu. Zobrazení detailu jednotlivých letů nabízí nespočet informací o průběhu letu a celkových statistikách. Aplikace nabízí možnost exportu detailu letu s vybranými informacemi, které si uživatel může zvolit.

Hlavní předností aplikace je možnost přeměnit mobilní zařízení na profesionální variometr díky možnosti připojení vario zařízení přes Bluetooth a využití GPS, akcelerometru a gyroskopu, které jsou zabudovány v mobilním telefonu. Aplikace má širokou podporu na různé druhy těchto zařízení. Uživatel má možnost si rozložení prvků na obrazovce měnit a přizpůsobit svým potřebám. Pilot může svůj pohyb sledovat na mapě, která obsahuje informace o přistávacích plochách, stoupavých proudech i bezletových zónách. Pro úsporu dat si mapové podklady může uživatel stáhnout do svého zařízení. Aktuální pozici pilota během letu lze online sledovat na podporovaných serverech, jako například XContest Live, Livetrack24 nebo Loctome. Uživatel se musí k danému serveru přihlásit svými přihlašovacími údaji. [48]

3.2.2 Gaggle - Flight Recorder

Další aplikací pro paragliding je Gaggle, která je nejlépe hodnocená v obchodě App Store podle pilotů, kteří létají s motorovým paraglidingem. Je dostupná pro mobilní zařízení s operačním systémem iOS a Android, ale je k dispozici i webové rozhraní. Lze ji stáhnout zdarma a nabízí tři balíčky předplatného. Základní balíček je zdarma a umožňuje vést letový deník, nahrávat přelety a sledovat živě piloty v okolí. Nejdražší balíček je za 8 \$ měsíčně a nabízí pokročilé funkce, jako je předpověď počasí, chatbot a 3D sledování přeletu na mapě.



■ **Obrázek 3.7** Ukázka aplikace Gaggle - Flight Recorder [49] [50] [51] [52]

Aplikace není určena pouze pilotům paraglidingu, ale i pilotům lehkých letadel, vrtulníků a vzdušných balónů. Zaznamenávání letu je proto na velmi vysoké úrovni. Přináší možnost sledovat během letu aktuální polohu na mapě a poskytuje informace o rychlosti větru, výšce či rychlosti stoupání. Dále zobrazuje v reálném čase piloty kolem vás, pomáhá s hledáním teplých stoupavých proudů vzduchu pro paraglidisty a funkce SafeSky vás informuje o možných nebezpečích.

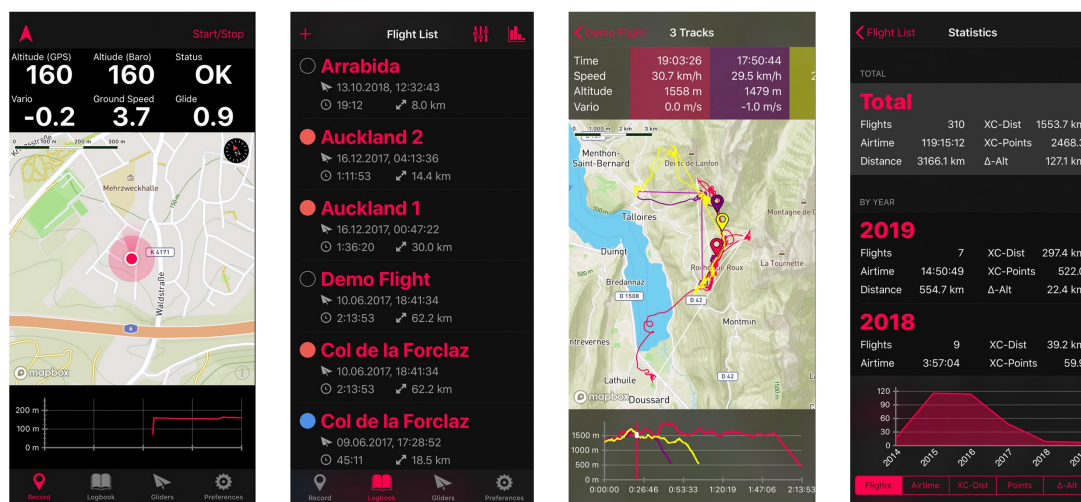
Aplikace přináší podrobnou analýzu přeletů, které pilot absolvoval. Krom statistik z letu, nabízí také přehrání přeletu, které zobrazí uživateli průlet trasou a informace o letu v jednotlivých úsecích.

V aplikaci je možné vytvářet komunity a skupiny, ve kterých si piloti mezi sebou sdílejí své přelety. Piloti si mohou mezi sebou posílat zprávy a sledovat navzájem svoje aktivity. [53]

3.2.3 Rogallo BlackBox

Za zmínku jistě stojí aplikace Rogallo BlackBox. Ta oproti předchozím aplikacím nepřináší nic nového, přesto stojí za zmínku, neboť nabízí funkcionalitu, které ji odlišují od již zmíněných aplikací. Je dostupná zdarma v obchodě App Store na platformě iOS a počítačích s operačním systémem MacOS.

Na obrázku 3.8 lze vidět, že aplikace umožňuje možnost záznamu letu, vedení letového deníku, analýzu přeletů a statistiky, které pilotovi ukáží jeho pokroky. Je zde navíc možnost porovnat let s dalšími přelety. Zaznamenané lety lze exportovat do IGC formátu a nahrát na server XContest. Aplikace také nabízí vést deník o vybavení a jeho údržbě. [55] Veškerá data se synchronizují přes iCloud a pilot si je může prohlížet, na svém počítači.



■ Obrázek 3.8 Ukázka aplikace Rogalo BlackBox [54]

3.2.4 Další aplikace a shrnutí

Výše uvedené aplikace ukazují možná řešení pro vedení letového deníku pro paraglidisty dostupná na trhu s mobilními aplikacemi. Nabídka aplikací je samozřejmě mnohem širší, avšak další aplikace obvykle nepřinášejí žádné inovativní funkce oproti již zmíněným.

Na trhu mobilních aplikací lze nalézt spoustu aplikací, které umožňují vedení letového deníku. Každá z nich se snaží přinést něco nového a odlišit se od ostatních. Základem všech aplikací je záznam letu a vedení letového deníku, který slouží jako cenný zdroj informací o letu.

Z dostupných aplikací pouze malá část umožňuje propojení se servery XContest, žádná z nich nenabízí přehled leteckých soutěží a jen některé se snaží o propojení pilotů. Některé z nich slouží primárně jako letové počítače s nepřehledným množstvím funkcionalit a správa letového deníku je až druhořadá. Aplikací, které by byli dostupné pro zařízení s operačním systémem Android a iOS, není mnoho. Mnoho aplikací je placených, a proto uživatelé, kteří přecházejí mezi telefony s různými operačními systémy, mohou platit různé aplikace pro danou platformu.

Na základě těchto zjištění a závěrů prezentovaných v analýze se tato bakalářská práce zabývá návrhem mobilní aplikace letového deníku pro paragliding. Tato aplikace bude řešit identifikované problémy a plnit specifické potřeby, které standardní řešení na trhu nenabízí.

3.3 Požadavky

Jedním z hlavních požadavků je správa přeletů, které piloti absolvovali. Uživatelé by měli být schopni přidávat nové přelety, editovat ty staré a zobrazit si svoji historii. Dalším důležitým požadavkem je přihlašování ke svému profilu.

Pro nepřihlášené uživatele je možnost zobrazení letů ostatních pilotů, jejich profilu a účast v soutěžích.

3.3.1 Funkční

F1 – Přihlašování uživatele

Aplikace bude umožňovat přihlášení uživatelů, kteří službu používali přes webové rozhraní. Po přihlášení bude uživatel mít možnost spravovat svůj profil a letový deník. Aplikace si uživatele zapamatuje, dokud se uživatel neodhlásí, aby se nemusel při každé nové interakci s aplikací přihlašovat.

F2 – Registrace uživatele

Neregistrovaní uživatelé budou mít možnost se registrovat. Uživatel zadá své údaje, vytvoří si jméno a heslo a zadá údaje ke svému novému profilu. Uživateli přijde potvrzovací e-mail a může se přihlásit ke svému účtu.

F3 – Správa přeletů

Přihlášený uživatel bude moci přidávat nové přelety, které absolvoval. Bude je moci následně editovat, přidávat obrázky a bude moci smazat přelety, které mu nevyhovují v letovém deníku v aktuálně probíhající soutěži.

F4 – Správa profilu pilota

Přihlášený uživatel bude moci upravovat údaje na svém profilu, měnit heslo k přihlášení a upravovat nastavení, které služba umožňuje.

F5 – Prohlížení profilů ostatních pilotů

Přihlášený i nepřihlášený uživatel bude mít možnost procházet žebříčky pilotů a zobrazit si detail jejich profilu.

F6 – Zasílání soukromých zpráv

Přihlášený uživatel bude mít možnost posílat ostatním uživatelům soukromé zprávy.

F7 – Prohlížení žebříčků soutěží

Přihlášení i nepřihlášení uživatelé budou moci prohlížet žebříčky soutěží a přelety, které k nim piloti přihlásili.

3.3.2 Nefunkční

N1 – Podporované platformy

Aplikace bude dostupná pro operační systémy Android a iOS v obchodech mobilních aplikací Google Play a App Store.

N2 – Lokalizace

Prostředí aplikace bude dostupné ve více jazycích. Základem bude čeština a angličtina.

N3 – Dostupnost

Aplikace bude data stahovat a posílat na cloud, čímž se minimalizuje riziko ztráty dat a zaručí se vysoká dostupnost, která bude alespoň 99%.

3.4 Model případů užití

Na základě funkčních požadavků aplikace identifikujeme uživatelské role(aktéry) a návaznost na případy užití, které si v této části popíšeme.

3.4.1 Aktéři

V mobilní aplikaci budou figurovat pouze dva aktéři — nepřihlášený uživatel a přihlášený uživatel.

3.4.1.1 Nepřihlášený uživatel

Prvním aktérem je nepřihlášený uživatel. Tím může být kdokoliv, kdo si stáhne mobilní aplikaci a začne ji používat. Má omezené možnosti, co může v aplikaci vykonávat.

3.4.1.2 Přihlášený uživatel

Druhým aktérem je přihlášený uživatel. Tím může být uživatel, který se na platformu registroval a přihlásil se v aplikaci ke svému účtu. Může využívat veškeré funkcionality, které aplikace nabízí.

3.4.2 Případy užití

Diagram 3.9 ukazuje aktéry a případy užití, které jsou v této kapitole podrobněji popsány.

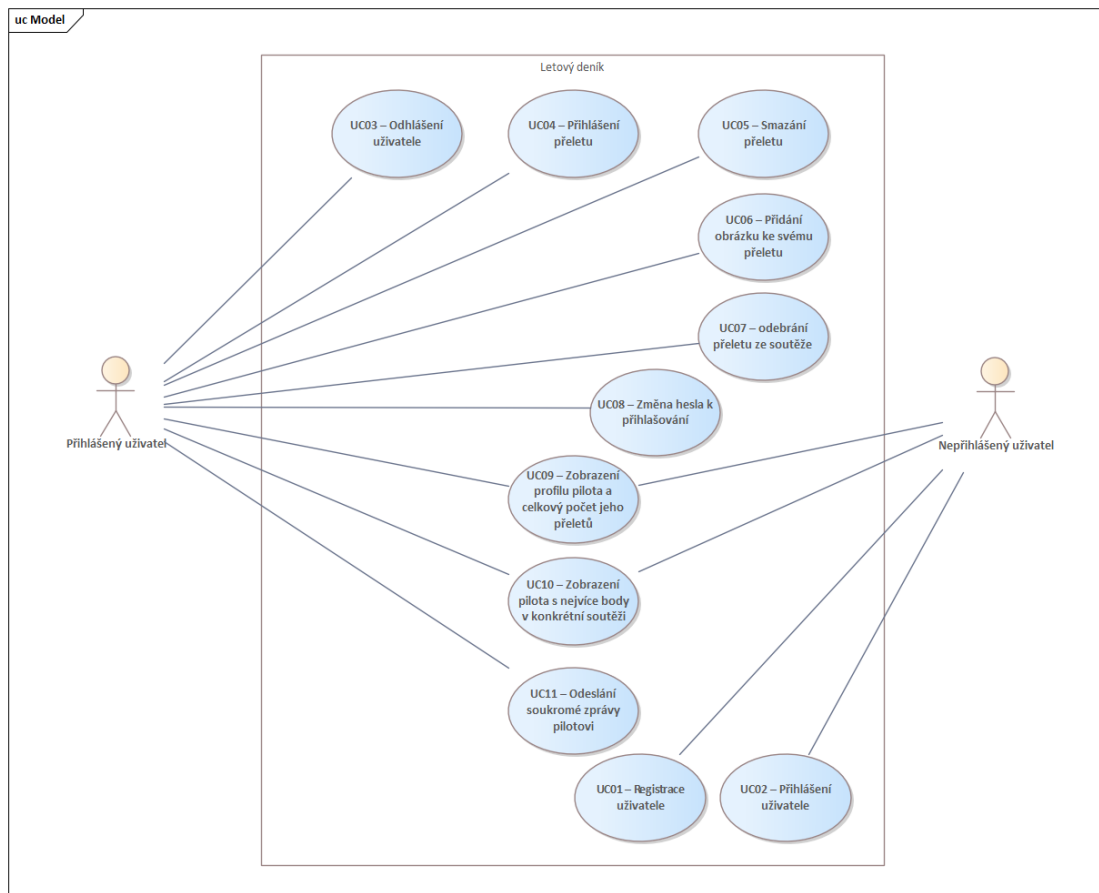
3.4.2.1 UC01 — Registrace uživatele

Neregistrovaný uživatel se může registrovat do aplikace.

Hlavní scénář případu užití:

1. Neregistrovaný uživatel přejde do sekce profilu uživatele a stiskne tlačítko *Registrace*.
2. Systém zobrazí formulář, který uživatel vyplní a stiskne tlačítko *Registrovat se*.
3. a) Uživatel vyplnil správně všechny údaje a scénář pokračuje krokem 4.
b) Uživatel nevyplnil všechna povinná pole nebo některý z údajů vyplnil špatně. Aplikace na to upozorní a uživatel se vrací do kroku 2.

- c) Uživatel stiskne tlačítko *Zrušit*. Skončí scénář případu užití a uživatel je navrácen na domovskou obrazovku.
4. Uživateli se zobrazí potvrzení o registraci a uživatel se může přihlašovat do aplikace (viz. UC02).



■ Obrázek 3.9 Model případů užití

3.4.2.2 UC02 — Přihlášení uživatele

Umožňuje **nepřihlášenému uživateli** použít své přihlašovací údaje a přihlásit se do aplikace. Tím se z něj stává **přihlášený uživatel** a může využívat veškeré funkcionality aplikace.

Hlavní scénář případu užití:

1. Nepřihlášený uživatel v aplikaci otevře sekci profilu a stiskne tlačítko *Přihlásit se*.
2. Aplikace zobrazí přihlašovací formulář s uživatelským jménem a heslem.
3. Uživatel údaje vyplní a stiskne tlačítko *Přihlásit se*.
4. a) Uživatel vyplnil správně všechny údaje a scénář pokračuje krokem 5.
b) Uživatel vyplnil některý z údajů špatně. Aplikace na to upozorní a vyzve uživatele ke kontrole údajů a uživatel se vrací do kroku 2.

- c) Uživatel stiskne tlačítko *Zpět*. Aplikace vrátí uživatele na domovskou obrazovku. Skončí scénář případu užití.

5. Přihlášení proběhlo úspěšně a uživateli se zobrazí informace o jeho profilu.

3.4.2.3 UC03 — Odhlášení uživatele

Umožňuje **přihlášenému uživateli** odhlásit se z aplikace. Přihlášený uživatel otevře sekci profilu a stiskne tlačítko *Odhlásit se*.

3.4.2.4 UC04 — Přihlášení přeletu

Případ užití umožňuje **přihlášenému uživateli** nahrát přelet do letového deníku.

Hlavní scénář případu užití:

1. Přihlášený uživatel otevře sekci letového deníku a stiskne na položku *Přihlásit přelet*.
2. Aplikace zobrazí uživateli obrazovku pro nahrávání souboru s GPS tracklogem ve formátu IGC.
3. Uživatel vybere ze souborů v mobilním zařízení konkrétní soubor a stiskne tlačítko *Odeslat*.
4. Aplikace odešle soubor ke zpracování a k validaci.
5. a) Pokud bude soubor v pořádku, bude zařazen do fronty k optimalizaci trati a zobrazí se formulář pro zadání zbývajících údajů.
b) V případě odmítnutí IGC souboru se zobrazí chybové hlášení s důvodem odmítnutí. V tom případě tento scénář případu užití končí a uživatel je vrácen na domovskou obrazovku.
6. Uživatel vyplní formulář a stiskne tlačítko *Odeslat*.
7. a) Uživatel vyplnil správně všechny údaje a scénář pokračuje krokem 8.
b) Uživatel nevyplnil všechna povinná pole. Aplikace na to upozorní a uživatel se vrací do kroku 6.
c) Uživatel stiskne tlačítko *Zpět*. Aplikace vrátí uživatele na obrazovku letového deníku. Skončí scénář případu užití.
8. Aplikace zobrazí jeho nově přihlášený přelet v leteckém deníku

3.4.2.5 UC05 — Smazání přeletu

Umožňuje **přihlášenému uživateli** smazání přeletu z jeho letového deníku.

Hlavní scénář případu užití:

1. Přihlášený uživatel otevře menu a stiskne položku *Deník*.
2. Aplikace zobrazí seznam přihlášených letů.
3. Uživatel nalezne přelet, který chce smazat.
4. Uživatel si zobrazí detail letu a stiskne tlačítko *Smazat let*.
5. V aplikaci vyskočí okno, které požádá o potvrzení nebo zrušení akce.
6. Uživatel akci potvrdí a let se smaže. Aplikace uživatele vrátí zpět do deníku.

3.4.2.6 UC06 — Přidání obrázku ke svému přeletu

Přihlášený uživatel má možnost ke svým přeletům přidávat obrázky, které pořídil.

Hlavní scénář případu užití:

1. Přihlášený uživatel otevře sekci letového deníku.
2. Aplikace zobrazí seznam přihlášených letů.
3. Uživatel nalezne přelet, ke kterému chce přidat obrázek.
4. Uživatel si zobrazí detail letu a stiskne tlačítko *Upravit let*.
5. Aplikace zobrazí formulář se všemi položkami, které může upravit.
6. Uživatel nalezne sekci *Fotky z přeletu* a stiskne tlačítko *Vybrat fotku*.
7. Uživatel ze své galerie fotek vybere fotografii, kterou chce nahrát k přeletu, a stiskne tlačítko *Vložit*.
8. Uživatel potvrdí provedené změny v přeletu stisknutím tlačítka *Odeslat*.
9. Aplikace zobrazí detail přeletu s provedenými změnami.

Alternativní scénář případu užití:

1. Uživatel provádí UC04 — Přihlášení přeletu a v kroku 5 ve formuláři nalezne sekci *Fotky přeletu*.
2. Uživatel stiskne tlačítko *Vybrat fotku* a vybere na svém zařízení fotografii.
3. Scénář případu užití dále pokračuje jako v UC04.

3.4.2.7 UC07 — Odebrání přeletu ze soutěže:

Případ užití umožňuje odebírat přelety pilotů ze soutěží, ke kterým pilot přelet přihlásil během přihlašování přeletu v UC04. Tím se v dané soutěži stávají neviditelnými a v soutěži nejsou zaznamenány.

Hlavní scénář případu užití:

1. Přihlášený uživatel otevře menu a stiskne na položku *Deník*.
2. Aplikace zobrazí seznam přihlášených letů.
3. Uživatel nalezne přelet, který chce odebrat ze soutěže.
4. Uživatel si zobrazí detail letu a stiskne tlačítko *Odebrat ze soutěže*.
5. Aplikace odebere tlačítko pro odebrání letu ze soutěže a v letovém deníku přidá označení, které identifikuje lety, které nejsou součástí soutěže.

3.4.2.8 UC08 — Změna hesla k přihlašování

Přihlášený uživatel má možnost změnit své heslo ke svému účtu v případě potřeby.

Hlavní scénář případu užití:

1. Přihlášený uživatel otevře sekci *Profil*.
2. V této sekci otevře nastavení.
3. V nastavení se nachází tlačítko *Změnit heslo*.
4. Po stisknutí zobrazí aplikace formulář s možností zadat nové heslo.
5. Uživatel vyplní formulář a potvrdí akci stisknutím tlačítka *Změnit heslo*.

6. a) Uživatel vyplnil správně formulář a scénář pokračuje krokem 6.
 - b) Uživatel nevytvořil silné heslo nebo špatně vyplnil formulář. Aplikace na to upozorní a uživatel se vrací do kroku 4.
 - c) Uživatel stiskne tlačítko *Zpět*. Aplikace vrátí uživatele na obrazovku nastavení. Skončí scénář případu užití.
7. Aplikace informuje uživatele o úspěšné změně hesla a vrátí uživatele do nastavení aplikace.

Alternativní scénář případu užití:

Tento scénář se týká **nepřihlášeného uživatele**, který zapomněl heslo k přihlášení.

1. Přihlášený uživatel otevře sekci *Profil*.
2. Uživatel stiskne tlačítko *Zapomenuté heslo*.
3. Aplikace zobrazí formulář, ve kterém uživatel vyplní uživatelské jméno, e-mailovou adresu a datum narození.
4. Uživatel stiskne tlačítko *Odeslat*.
5. a) Vyplněné údaje jsou v pořádku, pokračujeme ve scénáři případu užití.
- b) Vyplněné údaje jsou chybné nebo nejsou vyplněny povinné položky. Vracíme se ke kroku 3.
- c) Uživatel nechce odeslat formulář a stiskne tlačítko *Zrušit* a vrátí do sekce *Profil*.
6. Systém vytvoří nové heslo a zašle přihlašovací údaje na e-mailovou adresu uživatele.

3.4.2.9 UC09 — Zobrazení profilu pilota a celkový počet jeho přeletů

Uživatel má možnost prohlížet profily ostatních pilotů.

Hlavní scénář případu užití:

1. Uživatel otevře sekci soutěží.
2. V žebříčku bodování uživatel stiskne jméno pilota, kterého chce zobrazit.
3. Aplikace zobrazí detail pilota s jeho statistikami.

Alternativní scénář případu užití:

1. Uživatel otevře sekci soutěží.
2. V žebříčku letů zobrazí detail letu, který chce zobrazit.
3. V detailu letu stiskne jméno pilota.
4. Aplikace zobrazí detail pilota s jeho statistikami.

3.4.2.10 UC10 — Odeslání soukromé zprávy pilotovi

Případ užití umožňuje odesílání zpráv mezi přihlášenými uživateli aplikace.

Hlavní scénář případu užití:

1. Uživatel otevře sekci profilu.
2. Uživatel stiskne tlačítko *Zprávy*.
3. Aplikace zobrazí uživatele, se kterými si uživatel psal.
4. Uživatel stiskne na uživatele, kterému chce odeslat zprávu.
5. Aplikace zobrazí zprávy s uživatelem a pole pro napsání nové zprávy.

6. Uživatel napíše zprávu a stiskne tlačítko *Odeslat*.

Alternativní scénář případu užití: Tento scénář se týká případu, kdy si uživatelé ještě nikdy neposílali zprávy.

1. Uživatel nalezne v žebříčkách soutěží pilota, kterému chce odeslat zprávu.
2. Uživatel zobrazí profil pilota a stiskne tlačítko *Odeslat soukromou zprávu*.
3. Aplikace zobrazí formulář, do kterého uživatel může napsat zprávu.
4. Uživatel napíše zprávu a stiskne tlačítko *Odeslat*.
5. Aplikace potvrdí úspěšné odeslání soukromé zprávy.

3.4.3 Mapování požadavků na případy užití

Pro ověření pokrytí všech funkčních požadavků alespoň jedním případem užití jsem vytvořil tabulku 3.10.

	UC01	UC02	UC03	UC04	UC05	UC06	UC07	UC08	UC09	UC10
F1		X	X							
F2	X									
F3				X	X	X	X			
F4								X		
F5									X	
F6										X
F7									X	

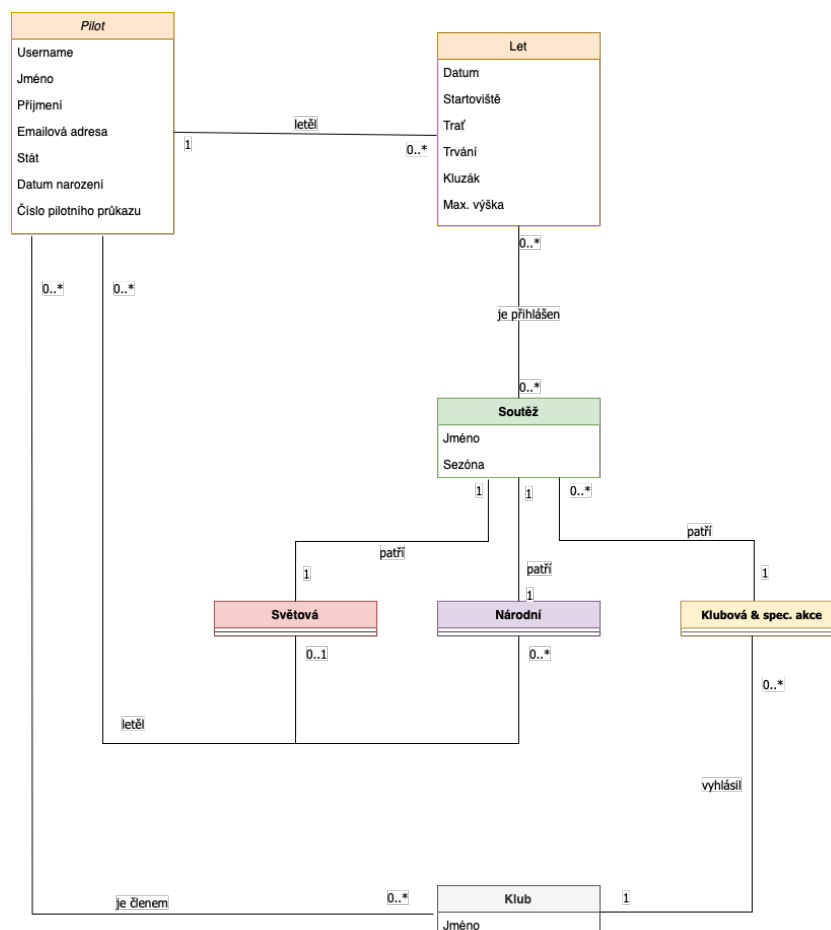
■ **Tabulka 3.10** Mapování požadavků na případy užití

3.5 Doménový model

V rámci analýzy platformy XContest byly prozkoumány její funkcionality a identifikovány klíčové entity, které tvoří doménový model. Cílem analýzy bylo komplexně porozumět fungování platformy a jejím základním informačním oblastem. Analýza doménového modelu povede k efektivnějšímu návrhu a implementaci platformy XContest, čímž se zlepší uživatelský komfort a celková funkčnost aplikace.

Mezi důležité entity, které doménový model na obrázku 3.11 obsahuje, patří:

- Pilot
- Let
- Soutěž
- Světová
- Národní
- Klubová & spec. akce
- Klub



■ Obrázek 3.11 Doménový model navrhované mobilní aplikace

Entita *Pilot* reprezentuje piloty, kteří používají XContest, a jejichž profil a letecké aktivity tvoří jádro dat. Každý pilot, který letěl a svůj let přidal do letového deníku, má možnost přihlásit svůj *Let* do *Soutěže*. Aby tak mohl učinit, musí být v soutěži přihlášen samotný pilot. Na XContest probíhají tři druhy soutěží, kterých se může pilot účastnit. *Světová* soutěž, která je jediná na XContest a kdokoliv se jí může účastnit. *Národní* soutěže odpovídají jednotlivým národnostem, které působí na XContest. Piloti se mohou účastnit více národních soutěží, když splní jejich podmínky. Pokud je pilot členem klubu, může se účastnit klubových soutěží a speciálních akcí.

Analýza doménového modelu přináší cenné poznatky o struktuře a fungování XContest. Tyto poznatky mohou být dále využity ke tvorbě mobilní aplikace a rozvoji jejích funkcí.

Návrh mobilní aplikace

Hlavním cílem je vytvoření prototypu mobilní aplikace, která bude uživatelům umožňovat vést letový deník pro paragliding a přinese funkcionality, které uživatelé znají z webového rozhraní služby. Pro splnění tohoto cíle jsou v této kapitole popsány zvolené podporované platformy a architektura aplikace. Na závěr kapitoly je vytvořena ikona aplikace, barevné schéma a jsou vytvořeny wireframy mobilní aplikace, které ukazují základní části aplikace a její ovládání.

4.1 Název navrhované aplikace

Než se začnou popisovat jednotlivé části návrhu mobilní aplikace, je vhodné si vytvořit její název. Název aplikace hraje důležitou roli, která ovlivňuje vnímání aplikace a jak si ji lidé budou pamatovat. Název by měl být relevantní k funkcionalitě aplikace, měl by být dostupný a odlišovat se od konkurence. Návrhů, jaký název by mohl být vhodný, bylo mnoho. Nakonec bylo vybrán název **XCApp**.

Splňuje podmínku dostupnosti a uživatelé si tento název spojí s XContest a jeho propagandou aplikací XCTrack, která slouží pro záznam letu. Zároveň je název krátký, snadno zapamatovatelný a použitelný po celém světě. Na rozdíl od aplikace XCTract, která podle svého názvu slouží k „trackování“přeletu, XCApp vyjadřuje, že se jedná o více komplexní aplikaci. Název aplikace XCApp bude ve zbytku práce používáno jako název navrhované a implementované aplikace této bakalářské práce.

4.2 Platforma aplikace

Téměř dvě třetiny lidí na světě používají mobilní zařízení, které se staly nedílnou součástí našich životů. [56] Každé z těchto zařízení pohání vlastní operační systém, který spravuje hardware zařízení, nabízí uživatelské rozhraní a stará se o chod aplikací. Operačních systémů se na světě objevilo mnoho. Každý přináší své výhody a nevýhody a uživatelé si mohou vybrat ten, který jim vyhovuje nejvíce podle svých preferencí a individuálních potřeb.

Jedním z nefunkčních požadavků navrhované aplikace je podpora mobilních operačních systémů Android a iOS. Tyto dva operační systémy dominují trhu s mobilními zařízeními a disponují rozsáhlou uživatelskou základnou. Jejich zastoupení na trhu mobilních zařízení je 99,24%. [57]

Vývoj multiplatformní aplikace přináší řadu benefitů, mezi které patří rozšíření dosahu a zvýšení počtu potenciálních uživatelů. Dále odolnost vůči změnám trendů na trhu, protože aplikace není závislá pouze na jedné platformě. V neposlední řadě umožňuje snazší vývoj aplikace, neboť lze aplikaci od začátku navrhovat s ohledem na více platform. Díky tomu se zjednodušuje

implementace oproti dodatečnému vývoji pro další platformy v pozdější fázi a sníží se náklady na vývoj.

Během analýzy bylo zjištěno, že absence podpory jednoho z těchto operačních systémů byl jeden z problémů, které současná řešení na poli mobilních aplikací měli. Uživatelé v případě multiplatformní aplikace nemusí hledat alternativní verze pro svůj operační systém a mohou se spolehnout na jednu univerzální verzi aplikace, která poběží na jejich preferovaném mobilním zařízení. Multiplatformní aplikace tedy zvyšují svoji dostupnost a uživatelské pohodlí.

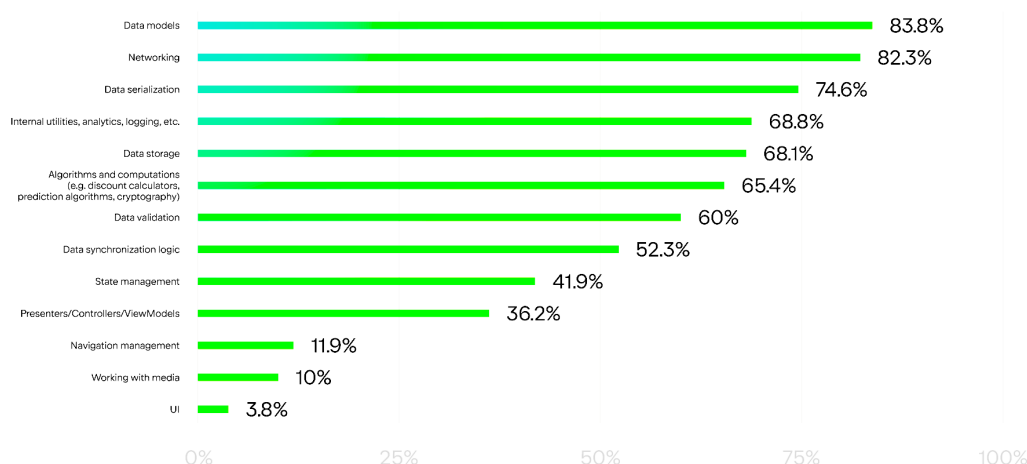
4.2.1 Vývoj multiplatformní aplikace

Jedním z možných způsobů, jak vytvořit aplikaci pro více platforem, je nativní vývoj. Aplikace vytvořené nativně jsou rychlejší, lépe responzivní a optimalizované. Hodí se na komplexní aplikace, kde je zapotřebí výkon. Mezi negativa se řadí vysoká cena, která je způsobena vytvářením stejných aplikací na různé platformy, výsledné aplikace nemusí být stejné z důvodů limitací, které platformy mají, a je zde větší prostor pro chyby, které se na každé platformě mohou projevit jinak.

V průběhu let se vývoj multiplatformních aplikací stal jedním z nejoblíbenějších způsobů vytváření mobilních aplikací. Oproti nativnímu vývoji, během kterého je aplikace vyvíjena pro každou konkrétní platformu zvlášť, umožňuje multiplatformní přístup vývojářům vytvářet aplikace, které běží podobně na různých mobilních platformách. Vytváření takové aplikace s jednotnou kódovou základnou zkracuje dobu vývoje tím, že eliminuje opakující se činnosti, jako je volání API nebo správa dat napříč platformami. [58] Jakou část implementace lze napříč platformami sdílet ilustruje tento průzkum 4.1 provedený společností JetBrains.

Sdílením implementace se šetří čas během vytváření multiplatformní aplikace, snižuje se počet řádků kódu a vzniká méně prostoru pro vytváření chyb. Mezi další výhody multiplatformních aplikací patří rychlejší testování a nasazení do provozu, menší nároky na velikost vývojového týmu a snížení celkových nákladů pro vytvoření aplikace. Vytváření aplikací napříč platformami má i své nevýhody. Jednou z nich je optimalizace a výkon aplikace, který se s těmi nativními nemůže srovnávat, přestože se v průběhu času lepší.

What parts of your code were you able to share between platforms?



■ Obrázek 4.1 Průzkum společnosti JetBrains týkající se sdílení kódu napříč platformami [59]

Popularita vývoje multiplatformních aplikací v posledních letech rychle roste a na trhu se objevilo mnoho nových nástrojů. Každý z nich se hodí pro různé požadavky a má své výhody i nevýhody. Mezi ty nejznámější frameworky patří:

- Flutter, 2.3.1
- React Native, 2.3.2
- Kotlin Multiplatform(KMP). 2.3.3

Flutter je multiplatformní framework od společnosti Google využívající programovací jazyk Dart 2.3.1.1. Mezi klíčové vlastnosti patří funkce Hot Reload, díky které dokáže změny okamžitě zobrazovat, a sdílí všechny vrstvy UX a UI. Zároveň nemusí působit nativně a pravděpodobně bude vyžadovat optimalizace.

React Native je open-source framework, který je založený na ReactJS. 2.3.2.1 Používá se především pro jednoduché a nenáročné aplikace a pro tým, který ovládá JavaScript a React.

KMP je open-source nástroj, vytvořený společností JetBrains, který umožňuje vývojářům sdílet kód napříč platformami při zachování výhod nativního programování. [60] Právě tento přístup přináší velkou flexibilitu a možnost sdílení takového množství, jak je potřeba.

Na základě dostupných informací a analýzy nástrojů je pro tento projekt zvolen nástroj Kotlin Multiplatform (KMP). Přináší spoustu výhod, které od multiplatformní aplikace je vyžadováno a umožňuje implementaci rozhraní aplikace nativním řešením. To přinese využití plného potenciálu aplikace na podporovaných platformách a přinese dostatečný výkon při náročnějších funkcionalitách aplikace. V další části se bude bakalářská práce zabývat architekturou aplikace s použitím KMP.

4.3 Databáze a napojení na webové řešení

Aplikace XCAApp disponuje vlastním datovým úložištěm, ze kterého získává potřebná data. Toto řešení zajišťuje nezávislost aplikace na třetích subjektech a unifikaci integrace služeb, což přispívá k její vyšší flexibilitě a udržitelnosti pro budoucí rozvoj. Datové úložiště je realizováno formou abstraktní vrstvy, která odděluje přístup k externím API od vnitřní implementace aplikace.

Tímto přístupem je zajištěno, že změny v externích službách nebudou mít přímý dopad na vnitřní fungování aplikace. Umožňuje napojení na libovolné služby bez nutnosti zasahovat do logiky aplikace. Zároveň zajišťuje jednotný přístup k datům s využitím různých externích služeb. Další výhodou je možnost efektivně zvládat omezení přístupů k externímu API a tím zajistit stabilní a spolehlivé fungování pro uživatele. V případě aplikace XCAApp se jedná o datové úložiště Firebase Firestore a externí API, která jsou potřeba pro fungování aplikace.

4.4 Architektura aplikace

Architektura aplikace definuje strukturu a uspořádání částí aplikace. Určuje, jak jsou jednotlivé části aplikace vzájemně propojeny a jak komunikují mezi sebou. Dobře navržená architektura umožňuje aplikaci snáze udržovat a rozvíjet. Zároveň přispívá k přehlednosti pro nové vývojáře a zvýšení celkové kvality kódu.

Aplikace XCAApp používá Clean Architecture 2.1.1 ve sdíleném kódu v KMP spolu s architekturou MVVM 2.1.2 pro Android a architekturou MVI 2.1.3 pro iOS. Problematika těchto architektur je podrobně rozebrána v teoretické části bakalářské práce 2.1. Pro zjednodušení rozdělíme kód na sdílenou část a platformní. Jednotlivé části jsou podrobněji popsány níže.

4.4.1 Sdílená část

Sdílená část architektury používá nástroj KMP. Ten umožňuje tři různé úrovně sdílení kódu v aplikaci:

- sdílet 100% kódu napříč platformami,
- sdílet obchodní logiku a implementaci zpracování dat,
- sdílet pouze část izolované části aplikace.

Aplikace XCAApp používá KMP pro sdílení logiky a zachovává nativního uživatelského rozhraní, tzn. že použije KMP pro implementaci zpracování dat a obchodní logiku. Implementaci uživatelského rozhraní ponechává nativní.

Sdílený modul se stará o síť, persistenci dat a obsahuje případy užití, které pomocí dependency injection (česky vložených závislostí) 2.2 propojují kód specifický pro danou platformu se sdíleným kódem. Struktura sdíleného modulu je uspořádána s ohledem na architekturu Clean do několika vrstev abstrakce:

- infrastruktura,
- datová vrstva,
- doménová vrstva.

Infrastruktura se zabývá externími systémy a technologiemi, které aplikace potřebuje k fungování. Patří sem například databáze nebo externí API. V aplikaci XCAApp se v této vrstvě vyskytuje volání metod pro získání dat z databáze Firebase Firestore 2.4.1 a zajištění autentizace pomocí Firebase Authentication 2.4.3. Infrastrukturní vrstva neobsahuje žádnou byznys logiku a je nezávislá na ostatních vrstvách. Díky tomu se lze soustředit na byznys logiku aplikace nezávisle na zvolených nástrojích v infrastruktuře. To přináší snazší údržbu a jednodušší aktualizaci databází.

Datová vrstva odděluje infrastrukturu od byznys logiky v doménové vrstvě. V této vrstvě jsou repositáře, které abstrahují detaily implementace databáze a zjednodušují práci s daty v byznys logice. To přináší přehlednost a snazší čitelnost kódu. V případě změny v infrastruktuře stačí upravit implementaci v datové vrstvě¹ a byznys logika zůstává netknutá.

Doménová vrstva obsahuje byznys logiku celé aplikace. Disponuje doménovými modely, které definují strukturu a jejich chování používané v byznys logice. Dále obsahuje případy užití. Jak název napovídá, měl by obsahovat všechny případy užití aplikace, tj. všechny funkce a byznys scénáře, které aplikace poskytuje, a definují požadavky a kroky potřebné k jejich splnění. Prezentační vrstva pak s případy užití komunikuje pro interakci s byznys logikou a zobrazení dat uživateli. Případy užití jsou univerzální pro jakoukoli platformu a mohou být použity opakovaně. [2]

Mezi hlavní výhody, které Clean architektura přináší napříč vrstvami je oddělení byznys logiky od technologií a prezentační vrstvy, udržovatelnost kódu, snazší úpravy, rozšířitelnost aplikace a jasnější strukturu a organizaci kódu.

4.4.2 Část specifická pro platformu

Nativní implementace specifická pro platformu reprezentuje prezentační vrstvu v Clean architektuře. Tato vrstva se zabývá interakcí s uživatelem a zobrazováním dat. Data přijímá pomocí volání Use Cases z doménové vrstvy. Každá platforma využívá jiný model architektury, který se výuce pro danou platformu hodí.

¹pokud je to nutné

Android využívá architekturu MVVM, kde ViewModely používají případy užití jako bránu do sdílené části modelu. Využívá knihovnu Jetpack Compose 2.5 pro definici rozhraní.

Pro iOS se používá architektura MVI. Model i Intent je definovaný ve ViewModelech jednotlivých Views. ViewModely využívají Use Cases ze sdíleného modulu, na které jsou pomocí Factory 2.2.1 vytvořeny závislosti do KMP. Nativní rozhraní pro platformu iOS je vytvořeno pomocí frameworku SwiftUI 2.8.

4.5 Návrh rozhraní aplikace

Přestože není design aplikace hlavním předmětem této bakalářské práce, intuitivní a efektivní rozhraní hraje důležitou roli pro zajištění pozitivního uživatelského zážitku. A tak se návrh rozhraní aplikace (UI design) stává klíčovou disciplínou v oblasti vývoje softwaru.

4.5.1 Cíl návrhu rozhraní aplikace

Při navrhování UI je důležité stanovit si jasné a měřitelné cíle, které povedou k pozitivnímu uživatelskému prožitku (UX). Webová platforma XContest obsahuje mnoho funkcionalit a členitou strukturu. Neresponzivní webové rozhraní umožňuje zobrazit velký počet prvků a rozsáhlé menu pro navigaci v obsahu, ale může vést k nepřehlednosti a obtížnému ovládání na menších zařízeních. Navrhovaná mobilní aplikace by naopak měla přinést jednoduché, přehledné rozhraní, které bude disponovat důležitými funkcionalitami a nebude komplikované.

Hlavním cílem je uživatelská spokojenost při používání s jasnou navigací. Uživatelské rozhraní by mělo usnadnit a zrychlit využívání platformy XContest, být atraktivní pro uživatele, konzistentní napříč platformami a dodržovat základních design guidelines².

4.5.2 Barvy a ikona navrhované aplikace

Barvy hrají v designu UI klíčovou roli. Dobře zvolená paleta barev může posílit celkový dojem z UI a zpříjemnit uživatelský prožitek.

Paleta barev je vytvořena na základě barev, které se používají na stránkách XContest. Mezi tyto barvy patří oranžová, tmavě modrá, modrá a světle modrá. Jednotlivé barvy jsou ukázány na obrázku 4.2.



■ **Obrázek 4.2** Barevná paleta používaná při návrhu aplikace XCAApp

²Pokynů pro navrhování

Ikona aplikace je malý, ale důležitý grafický prvek, který reprezentuje aplikaci na zařízeních uživatelů a v obchodech s aplikacemi. Dobře navržená ikona by měla být poutavá a snadno rozpoznatelná. Protože se jedná o aplikaci letového deníku pro paragliding, měla by ikona na první pohled reprezentovat aplikaci XCAApp. Zároveň by měla být relevantní k funkcím aplikace a napovědět uživatelům, o jakou aplikaci se jedná.

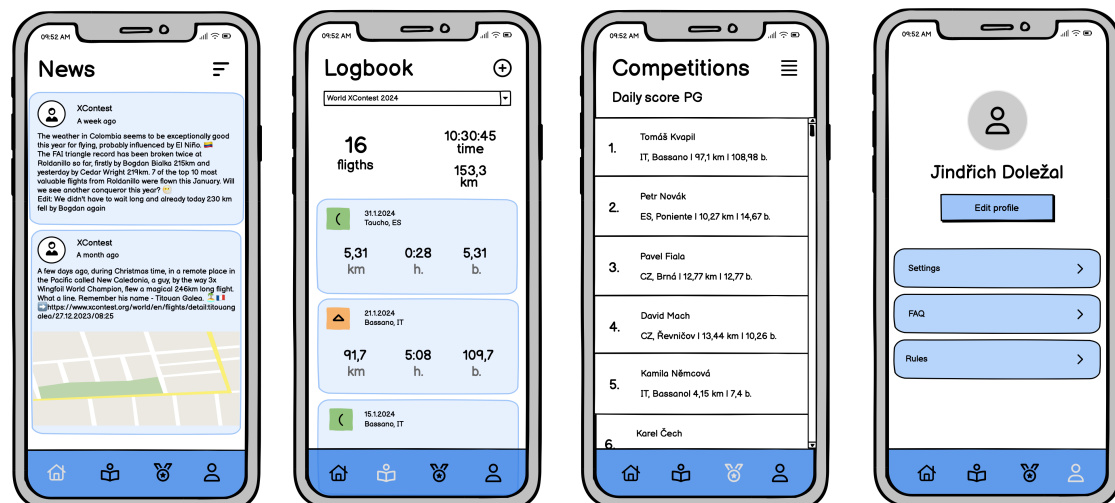


■ **Obrázek 4.3** Ikona aplikace XCAApp

Navržená ikona, kterou lze vidět na obrázku 4.3, je jednoduchá a snadno zapamatovatelná. Na modrém pozadí, které symbolizuje nebe, je mrak, kolem kterého prolétá pilot paraglidingu.

4.5.3 Prototyp aplikace

Vytvoření prototypu aplikace hraje klíčovou roli během procesu vývoje, neboť umožňuje navrhnout a otestovat prvotní nápady na design a strukturu navrhované aplikace, a to v krátkém čase a s nízkou náročností. Díky podrobné analýze a vytvoření funkčních požadavků lze snáze vytvořit prototyp aplikace.

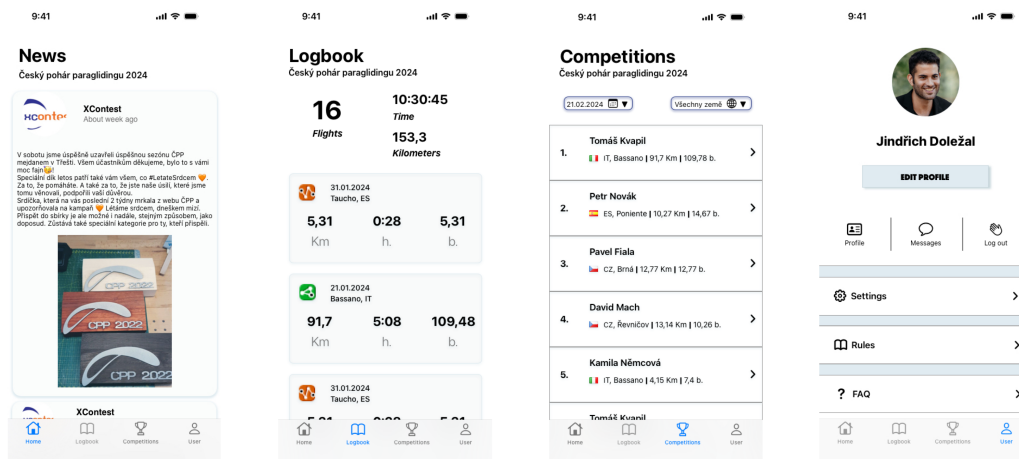


■ **Obrázek 4.4** Low-Fi prototyp aplikace vytvořený v desktopové aplikaci Balsamiq

Prvním vytvořeným prototypem je low fidelity³ prototyp v desktop aplikaci Balsamiq, který lze vidět na obrázku 4.4. Aplikace XCAApp má čtyři základní obrazovky, které pokrývají požadavky aplikace:

- Novinky,
- Deník,
- Soutěže,
- Profil.

Novinky zobrazují zprávy a příspěvky z facebookových stránek soutěží, podobně jako zobrazuje webové rozhraní XContest na úvodní stránce každé soutěže. Obrazovka *Deník* zobrazuje přihlášenému uživateli přihlášené přelety v letovém deníku k dané soutěži. Jednotlivé přelety může uživatel zobrazit a spravovat. *Soutěže* zobrazuje žebříčky soutěží, které si mohou uživatelé prohlížet. Poslední hlavní obrazovka *Profil* nabízí uživateli možnost spravovat svůj profil a nastavení aplikace. V případě odhlášeného uživatele mu obrazovka nabídne možnost se přihlásit nebo registrovat.



■ **Obrázek 4.5** Hi-fi prototyp aplikace vytvořený v aplikaci Sketch po opravení chyb objevené během testování

Pro otestování uživatelského rozhraní byl vytvořen high fidelity⁴ prototyp v aplikaci Sketch. Ten si lze prohlédnout na obrázku 4.5. Umožňuje procházení mezi obrazovkami a otestování, jak se bude aplikace chovat. Pro testování byl vytvořen krátký testovací scénář:

1. Přejděte do nastavení aplikace a změňte soutěž světovou soutěž World XContest na národní Český pohár paraglidingu. Prohlédněte si novinky na domovské obrazovce.
2. V soutěžích si vyberte Denní skóre ZK a naleznete pilota, který je první.

³nízkoúrovňový prototyp

⁴vysokoúrovňový prototyp

3. Registrujte se k XContest a v registračním formuláři se přihlaste k světové XContest soutěži a poté se přihlaste.
4. Zobrazte si detail svého posledního přidaného letu a zjistěte svoji průměrnou rychlost.
5. Zobrazte mapu svého posledního letu a v grafu nalezněte svoji maximální výšku.
6. Odhlaste se z aplikace.

Testování se zúčastnili čtyři studenti FIT ČVUT. Na základě jejich zpětné vazby byly navrženy tyto změny, které se promítnou do implementace aplikace:

1. Na obrazovku *Novinky, Deník a Soutěže* přidat nadpis, který bude informovat uživatele o tom, v jaké soutěži se nacházejí.
2. Při úspěšné registraci zároveň uživatele přihlásit do aplikace.
3. Zlepšení designu detailu přeletu.
4. Změnit název nastavení *region* na *soutěže*.
5. Používat velká písmena na začátku názvů a popisků.

Během testování nebyly nalezeny žádné velké nedostatky. Zpětné vazby studentů, kteří prototyp testovali, byly obdobné a netýkaly se navigace a uživatelské interakce s aplikací. Díky navrhnutí a otestování prototypu lze přejít na implementaci navržené aplikace XCAApp, která bude v další kapitole popsána.

Implementace

Tato kapitola bakalářské práce popisuje implementaci navržené mobilní aplikace XCAApp. Cílem této kapitoly je přiblížení procesu implementace a popsání důležitých kroků vývoje.

Implementovaná aplikace je navržena v souladu s vytvořeným návrhem architektury v předchozí kapitole. Prezentační vrstva je realizována pouze pro platformu iOS, a to z důvodu předchozích zkušeností autora s vývojem pro tuto platformu. Nicméně architektura aplikace je navržena tak, aby umožňovala snadné dopsání uživatelského rozhraní i pro platformu Android.

5.1 Vývojové nástroje

Výběr vývojových nástrojů pro vývoj multiplatformní aplikace s použitím KMP nebyl složitý, protože pro vývoj v tomto frameworku jsou preferována integrovaná vývojová prostředí od společnosti JetBrains, konkrétně Android Studio nebo IntelliJ IDEA. Oba nástroje poskytují širokou škálu podpůrných funkcí pro programovací jazyk Kotlin 2.3.3.1, na kterém je KMP založen. Android Studio navíc zajišťuje lepší integraci nástrojů pro vývoj mobilní aplikace pro platformu Android.

Pro vytváření nativního uživatelského rozhraní pro platformu iOS je nejlepším a téměř jediným řešením použít integrované vývojové prostředí Xcode od společnosti Apple. Xcode nabízí širokou škálu nástrojů pro vývoj, ladění a testování mobilních aplikací pro platformu iOS. Kromě toho umožňuje vývoj pro další platformy z ekosystému Apple. Xcode je k dispozici zdarma ke stažení, avšak pouze pro počítače s operačním systémem macOS.

Během procesu vývoje aplikace XCAApp byl používán verzovací nástroj GitHub. Ten pomohl během řešení chyb a vracení se na předchozí verze, sloužil pro zálohování projektu a přinesl možnost vyvíjet z více zařízení současně.

5.2 Začátek vývoje aplikace

Začátek vývoje aplikace může být náročný a plný úskalí. Vytvoření vhodné struktury aplikace, které bude odpovídat navržené architektuře, a instalace všech závislostí a knihoven může být pro začínající vývojáře těžký úkol. Naštěstí existuje mnoho nástrojů a zdrojů, které vývojářům v této fázi pomohou. Jedním z nich jsou veřejné šablony pro tvorbu aplikací.

Pro aplikaci XCAApp byla vybrána veřejná šablona pro vytváření multiplatformní aplikace od společnosti Matee. Tuto šablonu lze nalézt na Githubu společnosti Matee na tomto odkazu „<https://github.com/MateeDevs/devstack-native-app/tree/develop>”. Tato šablona obsahuje základní strukturu kódu, která koresponduje s architekturou aplikace XCAApp, a instalaci důleži-

tých balíčků a závislostí. Obsahuje i příklady implementace běžných funkcí a návod pro prvotní nastavení nového projektu.

Šablona s sebou přinesla spoustu užitečných nástrojů, které byly během vývoje použity. Protože bylo cílem vytvoření plně multiplatformní aplikace a vyhnutí se využití nativních nástrojů, bylo třeba zajištění persistence dat. SQLDelight 2.7 a Multiplatform Settings 2.6 toto dokáží zajistit. Pro zajištění lokalizace textů v aplikacích je použit nástroj Twine 2.9, který přináší jednoduché rozhraní, díky kterému lze jednoduše vytvářet uživatelské rozhraní ve více jazycích.

5.2.0.1 Konfigurace Firebase

Implementace platformy Firebase se ukázala jako jasná volba již od počátečních fází návrhu a vývoje aplikace. Nabízí širokou škálu služeb a možností, které pomáhají během vývoje aplikace. Klíčovými se staly služby Firebase Firestore a Authentication. Do budoucna se mohou jednoduše přidat další služby, které mohou umožňovat posílání notifikací nebo analýzu.

Ve většině mobilních aplikací se Firebase 2.4 a jeho služby používají na každé platformě zvlášť, neboť Firebase je platformně závislý. To v případě aplikace XCAApp nešlo použít, protože cílem bylo vytvořit plně multiplatformní aplikace, kde platformní implementaci představuje pouze prezentační vrstva.

Firebase Kotlin SDK poskytuje vhodné knihovny pro využití služeb Firebase v multiplatformní aplikaci vyvíjené v KMP. Knihovny z Firebase Kotlin SDK umožňují přístup k funkcionalitě Firebase již v rámci sdíleného modulu aplikace, aniž by bylo nutné psát duplicitní kód pro jednotlivé platformy. Tento přístup umožňuje abstrahovat použití služeb Firebase do vrstvy nezávislé na platformě a následně jej jednoduše propojit s nativními knihovnami Firebase.

Implementace Firebase Kotlin SDK [61] vyžadovala určitou časovou investici do nastavení veškerých potřebných konfigurací, neboť dokumentace poskytovaná tvůrci nezahrnovala podrobný postup pro integraci do multiplatformní aplikace vytvářené v KMP. Prvním krokem bylo přidání knihoven do Gradlu v souboru *libs.versions.toml*. Ukázku lze vidět zde 5.1.

```
firestore = "dev.gitlive:firebase-firestore:1.11.1"
firebase-kmm-auth = "dev.gitlive:firebase-auth:1.11.1"
```

■ Výpis kódu 5.1 Přidání knihoven z Firebise Kotlin SDK do Gradle

Druhým krokem bylo přidání konfigurace závislostí pro sadu zdrojů *commonMain* v multiplatformním projektu Kotlin v souboru *KmmLibraryConventionPlugin.kt*. Zdrojová sada *commonMain* je místo, kam se umísťuje kód sdílený napříč všemi platformami. K deklarování závislostí ke Gradlu bylo potřeba použít klíčové slovo *api* a ne *implementation*, protože poté budou mít všechny moduly, které závisí na *commonMain* modulu, také přístup k závislosti v *api*. Ukázka kódu 5.2 ukazuje tuto konfiguraci v modulech.

Dalším krokem byla změna během vytváření binárních souborů při buildu KMP v souboru *KMMConfig.kt*, jehož část lze vidět v ukázkovém kódu 5.3. Tyto soubory používají platformní implementace při používání KMP. Vytváření binárních souborů bylo změněno z dynamického na statické a exportovaly se do nich knihovny Firebase, aby je platformní implementace mohly používat a propojit s jejich platformními knihovnami.

Posledním krokem¹ bylo přidání knihoven Firebase do platformní implementace v XCodeu. Přidání souboru *GoogleService-Info.plist*, který se získá na webu v konzoli Firebase. Nakonec je třeba přidat konfiguraci Firebase do souboru *AppDelegate.swift*, kterou lze vidět zde 5.4.

¹tento krok se týká pouze platformy iOS

```
val commonMain by getting {
    dependencies {
        ⋮
        implementation(libs.bundles.settings)
        implementation(libs.bundles.sqlDelight.common)
        api(libs.firebase.kmm.auth)
        api(libs.firestore)
    }
}

val iOSMain by creating {
    dependsOn(commonMain) // getting dependencies from commonMain
    dependencies {
        implementation(libs.sqlDelight.iOSDriver)
        implementation(libs.ktor.iOS)
    }
}
```

■ **Výpis kódu 5.2** Konfigurace závislostí ve sdíleném modulu

```
KmmConfig.getSupportedMobilePlatforms(this@kmm, project).forEach {
    it.binaries.framework {
        if (this.buildType == KmmConfig.getCurrentNativeBuildType(project)) {
            baseName = nativeName
            isStatic = true
            xcf.add(this)
            export(libs.firestore)
            export(libs.firebase.kmm.auth)
        }
    }
}
```

■ **Výpis kódu 5.3** Změna v souboru KMMConfig.kt

```
// Setup firebase
if FirebaseApp.app() == nil {
    FirebaseApp.configure()
}
```

■ **Výpis kódu 5.4** Konfigurace Firebase na platformě iOS

5.3 Databáze a napojení na webové řešení

Jako databázové technologie byly pro tuto aplikaci zvoleny SQLDelight s databázovým enginem SQLite a Firebase Firestore.

První zmíněný byl využíván během začátku implementace pro vytváření dat a zkoušení fungování případů použití. Po naimportování multiplatformního Firebase se SQLDelight přestal používat, přesto byl v aplikaci ponechán, kdyby bylo potřeba ukládat lokální data nebo pro testování.

Firestore se stal primárním zdrojem dat pro aplikaci. Jeho použití je snadné, bezpečné a snadno se používá i s dalšími službami Firebase.

Bohužel se nepodařilo získat přímý přístup k externímu API XContest, který byl původně plánován. Tato situace si vyžádala dodatečnou analýzu dat, která jsou veřejně dostupná a viditelná na webovém řešení. Zároveň bylo potřeba vytvoření ukázkových dat, která umožní otestování prototypu aplikace a jeho funkčnosti.

Na základě analýzy byla vytvořena vlastní datová struktura a abstraktní vrstva pro práci s těmito daty. Základem se staly soutěže, které obsahují příspěvky, přihlášené piloty a přihlášené přelety. Krom toho je zde ještě jedna struktura pilotů, aby mohl být uživatel přihlášen do aplikace, ale nemusel se účastnit žádné soutěže.

Přestože toto řešení vyžadovalo dodatečné úsilí, zvolený přístup s vlastním datovým úložištěm a abstraktní vrstvou umožnil pokračovat v implementaci aplikace XCAApp.

5.4 Autorizace a autentizace

Aby mohl uživatel využívat letový deník a správu profilu, musí být v aplikaci přihlášen. K tomu se využívá služba Firebase Authentication. Ta nabízí jednoduchý a bezpečný způsob pro ověřování a autorizaci uživatelů. V aplikaci XCAApp se využívá pro registraci, přihlašování, odhlašování, ověření přihlášení a získání uživatelského ID.

Registrace i přihlašování se nachází na obrazovce uživatele. Během registrace vytváří uživatel nový uživatelský profil. E-mail a heslo se použije pro vytvoření nového uživatele ve Firebase Authentication a zbylé informace se použijí pro vytvoření profilu pilota v databázi. ID uživatele z Firebase Authentication se použije jako ID pilota v databázi Firestore. To zjednodušuje vyhledávání letce a jeho přihlášených letů. Při přihlašování využívá uživatel e-mail a heslo, které zadal při registraci.

Ověřování přihlášení probíhá při zobrazení obrazovky letového deníku a profilu uživatele. V případě, že uživatel není přihlášen, obrazovka letového deníku vyzve uživatele k přihlášení a obrazovka profilu uživatele nabídne přihlašovací a registrační tlačítka.

5.5 Ukázka implementace

V této části práce je ukázána implementace aplikace a detailněji popsány nástroje a frameworky, jak byly použity. Tato podkapitola je dělena na dvě části. První část se týká sdíleného kódu mezi platformami a druhá ukazuje vytváření uživatelského rozhraní pro iOS. Podobně jsme si aplikaci rozdělili i při návrhu architektury.

5.5.1 Sdílená část

V návrhu architektury aplikace jsou ve sdílené části tři vrstvy. Tyto vrstvy budou popsány s ukázkami implementace v aplikaci XCAApp.

5.5.1.1 Infrastruktura

V této vrstvě se nacházejí soubory s třídami, které komunikují s externími službami, jako je v případě naší aplikace služba Firebase Firestore, a načítají a manipulují s daty. Na ukázkou je zde třída *FlightService*, která se stará o operace relevantní k letům.

```
internal class FlightService internal constructor (
    private val repository: SettingsRepository
) {
    suspend fun getFlights(): Result<List<FlightDto>> {
        val contestSettingValue = repository.mySettings[0].get()
        val snapshotData = Firebase.firestore.collection("contest")
            .document(contestSettingValue).collection("flight").get()
        val flightsList = mutableListOf<FlightDto>()
        snapshotData.documents.forEach {
            val flightDto = it.toFlightDto
            flightsList.add(flightDto)
        }
        return Result.Success(flightsList)
    }
}
```

■ **Výpis kódu 5.5** Ukázka implementace třídy FlightSource a její funkce getFlights

V této vrstvě se také nacházejí modely DTO, které fungují jako zprostředkovatelé dat, kteří usnadňují výměnu dat mezi síťovým připojením a vrstvami aplikace. Pro převod mezi modely, jako například mezi DTO modelem a daty z databáze Firestore, se využívají rozšíření², která převádějí data přicházející z databáze na DTO model, který je v kódu reprezentován jako objekt.

Nakonec se v této vrstvě nalézá implementace tříd zdrojů. Ty využívají služby pro získávání dat z externích služeb a poskytují svoje metody datové vrstvě.

```
internal class FlightRemoteSourceImpl(
    private val service: FlightService,
) : FlightRemoteSource {
    override suspend fun getFlights(): Result<List<FlightDto>> =
        service.getFlights()

    override suspend fun getFlight(id: String): Result<FlightDto> =
        service.getFlight(id)
}
```

■ **Výpis kódu 5.6** Ukázka implementace třídy FlightSourceImpl

5.5.1.2 Datová vrstva

V datové vrstvě hrají repositáře podobnou roli, jako zdroje v infrastruktuře. Jejich úkolem je načítání dat ze zdrojů a převádění DTO modelů na doménové modely. V případě aplikace XCAApp mají repositáře vždy jen jeden zdroj dat, mohou jich však mít více. Například vzdálená data z Firestore a data z lokální databáze nebo externí API. V ukázce implementace 5.7 je ukázán repositář letu a jeho metody.

²V aplikaci XCAApp je lze najít ve sdíleném modulu ve složce extension

```

internal class FlightRepositoryImpl(
    private val flightRemoteSource: FlightRemoteSource,
) : FlightRepository {
    override suspend fun getAllFlights(): Result<List<Flight>> =
        flightRemoteSource.getAllFlights().map { it.map(FlightDto::asDomain) }

    override suspend fun getFlight(id: String): Result<Flight> =
        flightRemoteSource.getFlight(id).map(FlightDto::asDomain)
}

```

■ **Výpis kódu 5.7** Ukázka implementace třídy FlightRepositoryImpl

V této vrstvě se také nachází multiplatformní nastavení aplikace Multiplatform Settings. To poskytuje rozhraní pro správu nastavení aplikace a abstrahuje jejich uložení a načítání v celé aplikaci. SettingsRepository obsahuje list mySettings, který uchovává seznam objektů Setting-Config. Ta reprezentují jednotlivé položky, které se skládají z klíče a výchozí hodnoty. Každá položka má svoji podtřídu. V případě aplikace XCApp se jedná o podtřídu StringSettingConfig pro nastavení s řetězcovou hodnotou.

5.5.1.3 Doménová vrstva

Doménová vrstva obsahuje základní pravidla a obchodní procesy, které definují, co aplikace dělá. Ty se nacházejí v případech užití³, které zajišťují tok dat a poskytují metody, které může uživatel skrze uživatelské rozhraní provolávat. Případy užití tedy slouží jako brána mezi prezentační vrstvou a doménovou vrstvou. K tomu se využívají objekty Factory 2.2.1, které jsou zodpovědné za vytváření instancí objektů případů užití na základě požadavků. Factory používá injekci závislostí k určení, který případ užití má být vytvořen.

V této vrstvě jsou také doménové modely, které představují abstraktní reprezentace reality. V aplikaci XCApp se jedná o datové modely, které reprezentují strukturu dat v systému.

```

interface GetAllFlightsUseCase : UseCaseResultNoParams<List<Flight>>

internal class GetAllFlightsUseCaseImpl internal constructor(
    private var repository: FlightRepository,
) : GetAllFlightsUseCase {
    override suspend fun invoke() = repository.getAllFlights()
}

```

■ **Výpis kódu 5.8** Ukázka implementace rozhraní GetAllFlightsUseCase

5.5.2 Část specifická pro platformu

Tato část reprezentuje prezentační vrstvu v architektuře aplikace na platformě iOS. Během vývoje bylo použito několik nástrojů, které práci usnadnily. Prvním z nich byl Swift Package Manager, který pomáhá s instalací balíčků a správou závislostí v kódu. Dalším nástrojem byl Twine 2.9, který pomáhá s vytvářením lokalizovaných řetězců. Soubor *strings.txt* obsahuje řetězce, které jsou používány v aplikaci.

³V kódu aplikace je lze najít jako Use Case


```
[[Logbook View]]
  [logbook_view_notSingIn_label]
    cs = Bohužel, nejste přihlášen.
      Pro používání letového deníku je třeba se přihlásit.
    en = Unfortunately, you are not logged in.
      You need to be logged in to use the flight log.
    sk = Bohužiaľ, nie ste prihlásený.
      Ak chcete používať denník letov, musíte byť prihlásení.
```

■ **Výpis kódu 5.9** Ukázka lokalizace řetězců pro nástroj Twine

Twine následně generuje soubor *Localizable.strings* na základě obsahu v *strings.txt*. Při úpravě souboru *strings.txt* je nutné dodržovat stanovenou syntaxi řetězců a veškeré provedené změny je třeba promítnout do překladových souborů pomocí funkcionality Twinu, aby byly změny řetězců dostupné ve všech jazykových mutacích aplikace. V souboru *Localizable.strings* se nachází enum *L10n*, který v sobě obsahuje všechny vygenerované řetězce a při zavolání identifikátoru vrátí lokalizovaný řetězec, viz. ukázka 5.10.

```
struct LogbookNoLoginView: View {
  var body: some View {
    Text(L10n.logbook_view_notSingIn_label)
      .font(.title2)
      .padding()
      .frame(alignment: .center)
  }
}
```

■ **Výpis kódu 5.10** Ukázka použití lokalizovaných řetězců ve SwiftUI

Během celého vývoje nativní části pro iOS byl dodržován Swift Style Guide pro zachování konzistence a kvality napsaného kódu. Swift Style Guide doporučuje osvědčené postupy pro formátování, konvence pojmenování a obecný styl kódování při psaní kódu Swift. Pravidla není těžké dodržovat a spousta z nich vyžaduje samotné vývojové prostředí XCode během implementace. Díky zachování těchto konvencí je kód lépe čitelný a pomáhá novým vývojářům se rychleji zorientovat.

Pozorný čtenář si může při procházení kódu všimnout, že v nativní implementaci pro iOS se nachází složky obsahující doménovou vrstvu a datovou vrstvu. To je základní struktura, která se používá při vytváření čistě nativních aplikací. U aplikace *XCAApp* se do doménové vrstvy ukládá vygenerovaný soubor *XCAAppKMPShared*, který se vytváří při buildu KMP a obsahuje veškerou sdílenou logiku, na kterou jsou vytvořeny závislosti a může se s nimi v prezentační vrstvě pracovat.

Aby mohly být používány případy užití v prezentační vrstvě, musí být vytvořeny závislosti související s případy užití v KMP. Tyto závislosti jsou v souboru *KMPUseCases*. K poskytnutí těchto závislostí používá Koin a k načtení konkrétních případů užití používá *KMPDependency*. Toto nastavení umožňuje snadno načíst a použít tyto případy použití jinde ve vašem projektu tím, že se na ně jednoduše zeptá Koinu.

5.5.2.1 Prezentační vrstva

Do prezentační vrstvy pro platformu iOS patří ViewModel, View a FlowController.

```
// MARK: State
@Published private(set) var state: State = State()

struct State {
    var isLoading: Bool = true
    var posts: [Post] = []
}

// MARK: Intent
enum Intent {
    case loadPosts
}

func onIntent(_ intent: Intent) {
    executeTask(Task {
        switch intent {
            case .loadPosts: await loadPosts()
        }
    })
}

// MARK: Private
private func loadPosts() async {
    do {
        state.isLoading = true
        state.posts = try await getPostsUseCase.execute()
        state.isLoading = false
    } catch {
        print("Error during loading posts...")
        state.isLoading = false
    }
}
```

■ **Výpis kódu 5.11** Ukázka Viewmodelu pro zobrazení příspěvků

ViewModel představuje datový model pro prezentaci dat na dané obrazovce. Komunikuje s doménovou vrstvou a zajišťuje přípravu dat pro zobrazení ve View. Z modelu MVI se ve ViewModelu nalézá Model a Intent. Model představuje stav aplikace a obsahuje informace potřebné pro vykreslení uživatelského rozhraní. Pomocí Intentů měníme stavy. Zde 5.11 je zkrácená ukázka ViewModelu pro zobrazení příspěvků pro jednotlivé soutěže. Ukázka ukazuje jednoduchou implementaci stavů(State) a Intent.

View představuje grafické uživatelské rozhraní psaná ve SwiftUI pro zobrazení dat poskytovaných ViewModelem. View také reaguje na uživatelské interakce, ale neobsahuje žádnou logiku, tu zajišťuje ViewModel. Během vytváření View byla snaha dělení na různé moduly, aby byly View přehlednější a méně se opakoval kód.

FlowController řídí přechody mezi jednotlivými obrazovkami aplikace. Obsahuje logiku řízení přechodů a reakcí na uživatelské vstupy. Základem je *AppFlowController*, který se stará

```
import SwiftUI
import UIKit

struct PostsView: View {
    @ObservedObject private var viewModel: PostsViewModel

    init(viewModel: PostsViewModel) {
        self.viewModel = viewModel
    }

    var body: some View {
        return VStack {
            if viewModel.state.posts.isEmpty && viewModel.state.isLoading {
                PrimaryProgressView()
            } else {
                ScrollView {
                    ForEach(viewModel.state.posts, id: \.self) { post in
                        PostsCardView(post: post)
                    }
                }
                .padding()
            }
        }
        .lifecycle(viewModel)
        .navigationTitle(L10n.posts_view_toolbar_title)
    }
}
```

■ **Výpis kódu 5.12** Ukázka View pro zobrazení příspěvků

o nastavení *MainFlowControlleru* a nastavení vzhledu pro navigační bar a TabBar. *MainFlowController* nastavuje chování pro jednotlivé položky v TabBaru a přechod mezi nimi.

V TabBaru se přepíná mezi čtyřmi základními obrazovkami. Každá z těchto obrazovek má vytvořený svůj balíček, ve kterém jsou potřebná View, FlowController a ViewModely. Počet ViewModelů se odvíjí od toho, kolik logiky jednotlivé obrazovky potřebují.

V prezentační vrstvě se nachází balíček *UIToolKit*, který obsahuje základní View, která se dají použít napříč jednotlivými obrazovkami. Mezi taková View patří například tlačítko, textové pole, načítací obrazovka nebo zobrazení obrázku. Dále *UIToolKit* obsahuje nahrané obrázky, které se využívají v aplikaci, a definované barvy, které se používají napříč uživatelským rozhraním.

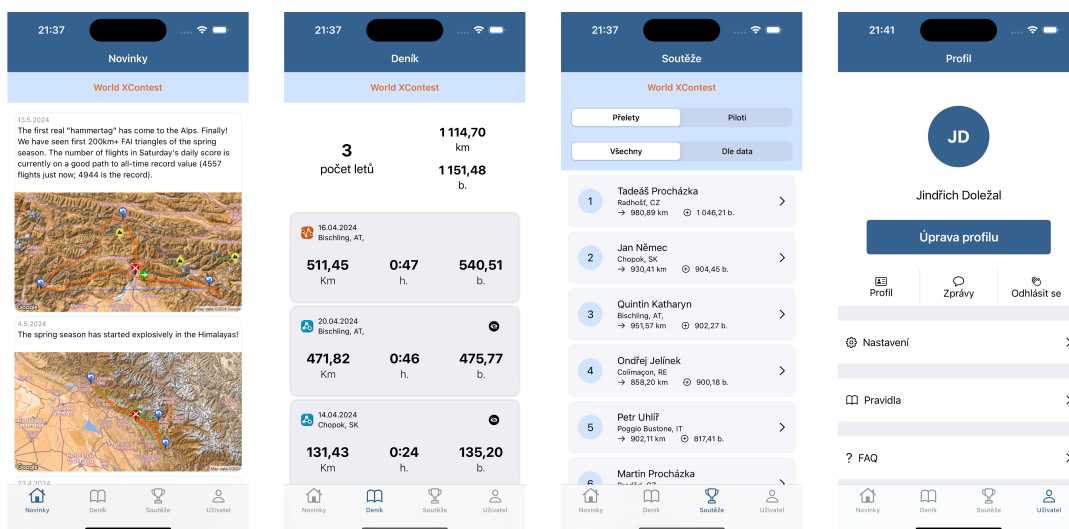
5.6 Zhodnocení implementace

Během implementace mobilní aplikace XCAApp se objevilo několik překážek, které ovlivnily průběh vývoje a výsledek práce. Díky vhodně vytvořené architektuře aplikace se s nimi podařilo vypořádat a vytvořit funkční prototyp.

Hlavní překážkou bylo nezískání přístupu k externímu API služby XContest. To vyžadovalo dodatečné úsilí pro vytvoření databázového modelu a potřebných datových struktur. Dále to ovlivnilo implementaci některých definovaných případů užití a u některých z nich znemožnilo jejich implementaci.

Jedním takovým je UC04, který vyžadoval vložení IGC souboru, jeho validaci a získání dat o přeletu pilota. Bez přístupu k API služby XConest nebyla možnost soubor validovat ani získat data ze souboru. Jednou z možností bylo vytvoření vlastní validace souboru a parsování dat ze souboru, bohužel k tomu již nebyl dostatek času během implementace aplikace. Z časových důvodů nebyl implementován ani UC10, pro který nebyla provedena dostatečná analýza fungování. Tento případ užití souvisel s funkčním požadavkem F6.

Dalším případem užití, který není implementován v aplikaci, je UC06. Jeho implementace vyžadovala napojení služby Firebase Storage 2.4.2, která umožňuje nahrávání souborů, a v databázové službě Firebase Firestore použití referencí na konkrétní soubory v úložišti.



Obrázek 5.1 Ukázka uživatelského rozhraní implementovaného prototypu aplikace XCAApp

Přes uvedené překážky se podařilo vytvořit funkční prototyp mobilní aplikace XCAApp, jejíž ukázkou si lze prohlédnout na obrázku 5.1. Další ukázky lze nalézt v příloze A. V prototypu jsou implementovány zbylé funkční požadavky a případy užití. Jsou implementovány i funkcionality navíc, které nebyly v případech užití specifikovány, jako například úprava údajů profilu uživatele nebo filtrování přeletů v soutěži. Následující kapitola se bude zabývat testováním tohoto prototypu a ověřením jeho funkčnosti.

Kapitola 6

Testování

Tato kapitola popisuje testování vyvinuté aplikace z předchozí kapitoly. Aplikace byla otestována integračními testy a bylo provedeno uživatelské testování.

6.1 Testování aplikace

Testování je klíčovou součástí vývoje softwaru a v případě mobilních aplikací hraje důležitou roli pro zajištění spolehlivosti a funkčnosti. Pro tento účel slouží ve vývojovém prostředí Xcode nástroj XCTest 2.10, který umožňuje testovat různé aspekty mobilní aplikace. Testování v aplikaci XCAApp se soustředilo na funkčnost funkcí ve ViewModelech a prověřovalo i navigaci mezi FlowControllery. Tyto testy posloužili jako klíčová kontrola, že se funkce ve ViewModelech a navigace chovají v souladu s očekáváním a zajišťují tak plynulý chod aplikace.

Pro pokrytí komplexnější úloh v aplikaci byl vytvořen testovací scénář 6.2.2, který lze nalézt v uživatelském testování. Tento scénář pokrývá téměř všechny implementované případy užití a možné situace, které mohou nastat. Díky tomuto scénáři byli opraveny chyby, které se týkaly uživatelského rozhraní, špatné navigaci mezi obrazovkami a špatného formátování číselných údajů.

Během testování byly nalezené chyby opraveny, aby následné uživatelské testování nebylo narušováno možným výskytem problémů a uživatelé se mohli plně soustředit na hodnocení uživatelského rozhraní a zážitku z aplikace.

6.2 Uživatelské testování

Uživatelské testování je důležitou součástí procesu vývoje mobilní aplikace, která pomáhá zajistit, že software splňuje potřeby uživatelů a je snadno použitelná. Poskytuje cennou zpětnou vazbu, na základě které lze provést nezbytné úpravy a optimalizace, a tím zajistit, že aplikace bude plně funkční a intuitivní pro cílové uživatele.

Pro účely uživatelského testování byla mobilní aplikace instalována na zařízení s operačním systémem iOS. Toto řešení zajistilo, že testování bude probíhat v prostředí, které je pro koncové uživatele přirozené a intuitivní. Testování bylo prováděno prezenčně, s přímou účastí testujících uživatelů. Tento přístup umožnil získat cennou zpětnou vazbu a pozorovat chování uživatelů při interakci s aplikací v reálném čase.

Uživatelské testování bylo strukturováno do tří na sebe navazujících fází. V úvodní fázi byli testující uživatelé požádáni o zodpovězení vstupního dotazníku. Tento krok umožnil získat základní informace o profilu testujících uživatelů. Následovala hlavní část testování, v rámci které uživatelé plnily předem připravené scénáře a úkoly. Tyto scénáře korespondovaly s klíčovými

funkcionalitami a případy užití aplikace. Závěrečná fáze spočívala v zodpovězení výstupního dotazníku. Ten umožnil uživatelům vyjádřit celkový dojem z aplikace, identifikovat silné a slabé stránky a navrhnout možná vylepšení.

6.2.1 Vstupní dotazník

Cílem vstupního dotazníku bylo zjistit základní informace o profilu testovacích uživatelů, jejich zkušenostech a orientaci v dané problematice. Testujícím uživatelům byly položeny následující otázky:

- Studujete nebo jste vystudoval vysokou školu? Pokud ano, jakou?
- Jste v současné době zaměstnán? Pokud ano, co je náplní vaší práce?
- Používáte telefon s operačním systémem Android nebo iOS?
- Máte zkušenosti s létáním paraglidingu?
- Znáte službu XContest? Popř. k čemu ji používáte?

Pokud testovací uživatel neměl zkušenost se službou XContest, byla mu tato služba stručně představena před samotným testováním. Tester uživateli ukázal základní funkcionality a vysvětlil princip fungování služby, aby uživatel mohl plnohodnotně absolvovat připravené testovací scénáře a neznalost služby XContest nebyla překážkou.

6.2.2 Testovací scénář

Testujícímu uživateli byl poskytnut mobilní telefon s předinstalovanou aplikací XCApp. Následně mu tester spustil prezentaci s připraveným testovacím scénářem, který uživatel krok za krokem plnil:

1. Otevřete aplikaci XCApp na mobilním zařízení.
2. Přečtěte si novinky na domovské obrazovce. Poznáte z této obrazovky, v jaké soutěži se nacházíte?
3. Přepněte se prosím na českou národní soutěž Český pohár paraglidingu.
4. Nalezněte přelet s nejvyšším bodovým hodnocením dne 17.4.2024. Jaké maximální výšky pilot během letu dosáhl?
5. Zobrazte si prosím profil pilota letu. Kolik má přihlášených letů v soutěži?
6. Registrujte se v aplikaci. Údaje vyplňte dle přiložených instrukcí zadavatele.
7. Změňte prosím datum narození profilu dle instrukcí zadavatele.
8. Zkontrolujte své údaje v detailu svého profilu.
9. Odhlaste se z aplikace.
10. Přihlaste se prosím do aplikace přihlašovacími údaji, která Vám poskytne zadavatel.
11. Změňte heslo k profilu, aby se k němu zadavatel nedostal.
12. Zobrazte si prosím letový deník. Profil uživatele má nahrané lety v aplikaci. Pokud se nezobrazují, jak je zobrazit?

13. Kolik letů máte přihlášených v deníku? Kolik z těchto letů je přihlášených v soutěži?
14. Zobrazte si detail svého letu s nejméně body a zjistěte svoji průměrnou rychlost.
15. Odeberte tento let ze soutěže.
16. Odstraňte let s nejméně body v letovém deníku.

6.2.3 Výstupní dotazník

Po úspěšném projití testovacího scénáře byly testovacímu uživateli položeny otázky z výstupního dotazníku. Jedná se o nejdůležitější fázi testování, během které se od uživatelů získá zpětná vazba a jejich celkový dojem z mobilní aplikace XCAApp.

- Jak byste celkově ohodnotili svou zkušenost s používáním aplikace? Narazili jste během testování na nějaké problémy?
- Které funkce se vám líbily nejvíce a proč? Které naopak méně a proč?
- Chyběla vám nějaká důležitá funkce nebo možnost, kterou jste v aplikaci postrádali?
- Jak byste zlepšili nebo změnili uživatelské rozhraní a ovládání aplikace?
- Máte nějaké další návrhy nebo připomínky, které by mohly pomoci zlepšit aplikaci?

6.2.4 Testování s uživateli

Testování se účastnilo celkem 8 uživatelů. Záměrem bylo zajistit, že aplikace bude otestována jak na uživateli, kteří doposud neměli žádnou zkušenost se službou XContest, tak i na cílové skupině koncových uživatelů, pro které je aplikace primárně určena. Zároveň byli uživatelé vybíráni tak, aby testovací vzorek byl co nejrozmanitější a byl otestován na různorodé skupině potenciálních koncových uživatelů aplikace.

6.2.4.1 Uživatel 1

Vstupní dotazník

- Studujete nebo jste vystudoval vysokou školu? Pokud ano, jakou?
„Ano, jsem studentem bakalářské programu na FIT ČVUT.“
- Jste v současné době zaměstnán? Pokud ano, co je náplní vaší práce?
„Nejsem zaměstnán, ale mám brigády v oblasti e-commerce.“
- Používáte telefon s operačním systémem Android nebo iOS?
„Android.“
- Máte zkušenosti s létáním paraglidingu?
„Nemám.“
- Znáte službu XContest? Popř. k čemu ji používáte?
„Neznám, ale mám kamarády, kteří službu používají. Osobní zkušenost se službou nemám.“

Průběh testování

Během testování uživatel narazil na problém s přepnutím soutěže, které hledal na obrazovce *Soutěže*. Rychle ale pochopil, že přepínání soutěží se nachází v nastavení aplikace. U bodu 11 si nebyl uživatel jistý, zda zadal heslo správně a rád by viděl možnost validace. U dalších kroků neměl uživatel problém a vždy rychle pochopil, jak s aplikací pracovat a splnit úkol.

Výstupní dotazník

- Jak byste celkově ohodnotili svou zkušenost s používáním aplikace? Narazili jste během testování na nějaké problémy?
„Aplikace je hezká. Ze začátku jsem měl problém s přepnutím soutěže, hledat v nastavení pro mě byla poslední možnost. Čekal jsem, že v záložce Soutěže se bude nacházet možnost soutěže přepínat. Možná by bylo lepší to nazvat pouze Soutěž.“
- Které funkce se vám líbily nejvíce a proč? Které naopak méně a proč?
„Líbí se mi filtrování soutěží. Líbí se mi podrobné statistiky u přeletů. Naopak nebyl jsem si jistý s validací u hesla. Zároveň bych přidal i picker pro zadávání data narození, pohlaví i státu.“
- Chyběla vám nějaká důležitá funkce nebo možnost, kterou jste v aplikaci postrádali?
„Přidání mapy do detailu letu. Možnost přidávat let by byla také super.“
- Jak byste zlepšili nebo změnili uživatelské rozhraní a ovládání aplikace?
„Občas mi tam chyběly vysvětlivky. Přestože je zde připravená stránka FAQ, mám rád když jsou věci intuitivní a rovnou na místě.“

- Máte nějaké další návrhy nebo připomínky, které by mohly pomoci zlepšit aplikaci?
„Začal bych přidáním validace hesla a možnosti resetovat heslo, pokud ho uživatel zapomněl. Pokud se doplní ty další věci, co jsem tady zmiňoval, bylo by to skvělé a stačilo by to.”

6.2.4.2 Uživatel 2

Vstupní dotazník

- Studujete nebo jste vystudoval vysokou školu? Pokud ano, jakou?
„Ano, studuji magisterský obor na FIT ČVUT.”
- Jste v současné době zaměstnán? Pokud ano, co je náplní vaší práce?
„Ano, pracuji na poloviční úvazek. Jsem vývojář mobilní aplikace.”
- Používáte telefon s operačním systémem Android nebo iOS?
„iOS”
- Máte zkušenosti s létáním paraglidingu?
„Ne.”
- Znáte službu XContest? Popř. k čemu ji používáte?
„Neznám.”

Průběh testování

Podobně jako uživatel 1 měl uživatel 2 problém s přepnutím soutěže. Po nápovědě uživatel úspěšně přepnul soutěž. S dalšími úkoly neměl uživatel problém a došel až ke konci testovacího scénáře.

Výstupní dotazník

- Jak byste celkově ohodnotili svou zkušenost s používáním aplikace? Narazili jste během testování na nějaké problémy?
„Aplikace funguje dobře. Během testování aplikace nespadla ani se nezasekla. Nečekal jsem, že přepnutí letu bude tak složité a bude se muset do nastavení.”
- Které funkce se vám líbily nejvíce a proč? Které naopak méně a proč?
„Líbil se mi žebříček přeletů i to, že je to seřazeno podle bodů. Jsou tam také detailní data, takže si člověk může najít cokoliv co potřebuje. Naopak, u zadávání data narození bych čekal picker, abych to nemusel psát.”
- Chyběla vám nějaká důležitá funkce nebo možnost, kterou jste v aplikaci postrádali?
„Mapa. Zobrazení mapy toho letu by byla super.”
- Jak byste zlepšili nebo změnili uživatelské rozhraní a ovládání aplikace?
„Validace dat, které jsou v registraci. Krom hesla také datum a u jména přidat možnost doplnění na velké písmeno, pokud uživatel začne psát malým.”
- Máte nějaké další návrhy nebo připomínky, které by mohly pomoci zlepšit aplikaci?
„Získat přístup k API služby. To by určitě posunulo vývoj celé aplikace kupředu.”

6.2.4.3 Uživatel 3

Vstupní dotazník

- Studujete nebo jste vystudoval vysokou školu? Pokud ano, jakou?
„Ano, studuji bakalářský obor na fakultě architektury ČVUT.“
- Jste v současné době zaměstnán? Pokud ano, co je náplní vaší práce?
„Ne. Pouze studuji. Ve volném čase pomáhám s projektováním výstavby domu.“
- Používáte telefon s operačním systémem Android nebo IOS?
„Ani jeden. Mám tlačítkový telefon Nokia.“
- Máte zkušenosti s létáním paraglidingu?
„Nemám.“
- Znáte službu XContest? Popř. k čemu ji používáte?
„Ne.“

Průběh testování

Podobně jako ostatní uživatelé nastal problém při přepnutí soutěže v aplikaci. Uživatel se snažil nalézt možnost přepnutí soutěže na obrazovce Soutěže. Vylučovací metodou nakonec našel možnost přepnout soutěž v nastavení. Při změně hesla šel uživatel intuitivně do profilu uživatele a až poté do nastavení, kde se heslo mění. Během počítání přeletů, které jsou v soutěži, byla nutnost proklikat si detaily jednotlivých přeletů. Nakonec si uživatel zjistil, že jsou také lety označeny v letovém deníku. V dalších úkolech neměl uživatel problém a došel na konec testovacího scénáře.

Výstupní dotazník

- Jak byste celkově ohodnotili svou zkušenost s používáním aplikace? Narazili jste během testování na nějaké problémy?
„Celkově bych ohodnotil aplikaci body 7/10. Měl jsem občas problém pochopit, jak aplikace funguje, ale to bude asi tím, že nejsem uživatelem XContest. Například jsem měl trochu problém pochopit, že pokud změním soutěž, tak celá aplikace a její data se vztahují k dané soutěži.“
- Které funkce se vám líbily nejvíce a proč? Které naopak méně a proč?
„Líbilo se mi označení nepřihlášených letů v soutěži. Hezký designový prvek. Nelíbilo se mi, že si jednotlivé sekce pamatují, kde jsem skončil. Když přejdu do jiné sekce a vrátím se, chtěl bych být zase na té úvodní obrazovce a neproklíkávat se zpět.“
- Chyběla vám nějaká důležitá funkce nebo možnost, kterou jste v aplikaci postrádali?
„Rád bych viděl rozřazení jednotlivých sekcí... třeba soutěží, aby tady bylo postranní menu pro navigaci.“
- Jak byste zlepšili nebo změnili uživatelské rozhraní a ovládání aplikace?
„Bylo to přehledné, až na změnu přepínání soutěže je aplikace hezká.“
- Máte nějaké další návrhy nebo připomínky, které by mohly pomoci zlepšit aplikaci?
„Přidal bych animaci paraglidisty při načítání nebo přechodech. To by se mi líbilo.“

6.2.4.4 Uživatel 4

Vstupní dotazník

- Studujete nebo jste vystudoval vysokou školu? Pokud ano, jakou?
„Ano, studuji bakalářský obor na FIT ČVUT.”
- Jste v současné době zaměstnán? Pokud ano, co je náplní vaší práce?
„Ano, pracuji jako vývojář mobilních aplikací primárně pro platformu iOS.”
- Používáte telefon s operačním systémem Android nebo IOS?
„iOS”
- Máte zkušenosti s létáním paraglidingu?
„Mám kamarády, kteří létají.”
- Znáte službu XContest? Popř. k čemu ji používáte?
„Slyšel jsem o ní, ale nikdy jsem ji nepoužíval.”

Průběh testování

Uživatel zaváhal u úkolu přepnutí soutěže, ale rychle se zorientoval a úspěšně soutěž přepnul. Projít zbytkem scénáře nebyl pro uživatele problém. Během procházení měl připomínky na fungování aplikace, které shrnul během výstupního dotazníku.

Výstupní dotazník

- Jak byste celkově ohodnotili svou zkušenost s používáním aplikace? Narazili jste během testování na nějaké problémy?
„Celkově mám z aplikace příjemný pocit. Líbí se mi, že je to nativní aplikace a ne WebView. Kvůli tomu, že nejsem uživatel XContest, byly pro mě některé věci těžší pochopit, například který let je v dané soutěži. Později mě napadlo, co signalizuje ikonka u letů v letovém deníku, ale musel jsem si nejprve proklikat přelety a odvodit si to z nich.”
- Které funkce se vám líbily nejvíce a proč? Které naopak méně a proč?
„Líbí se mi zeď s příspěvky. Přepínání soutěží není moc intuitivní. Mohlo by se to přepínat přímo v tom nadpisu.”
- Chyběla vám nějaká důležitá funkce nebo možnost, kterou jste v aplikaci postrádali?
„V soutěžích bych rád viděl vyhledávání přeletů podle jména pilotů.”
- Jak byste zlepšili nebo změnili uživatelské rozhraní a ovládání aplikace?
„Do nastavení bych nedával změnu hesla, ale dal bych ji do úpravy profilu nebo přidal tuto možnost k detailu pilota. Určitě bych přidal picker pro výběr data narození a pohlaví uživatele. Z pohledu vývojáře bych prodloužil načítací obrazovky, aby byla aplikace plynulejší.”
- Máte nějaké další návrhy nebo připomínky, které by mohly pomoci zlepšit aplikaci?
„Krom těch věcí co jsem zmiňoval, tak mě nic jiného nenapadá.”

6.2.4.5 Uživatel 5

Vstupní dotazník

- Studujete nebo jste vystudoval vysokou školu? Pokud ano, jakou?
„Ano, studuji bakalářský obor na FA ČVUT.“
- Jste v současné době zaměstnán? Pokud ano, co je náplní vaší práce?
„Ne.“
- Používáte telefon s operačním systémem Android nebo IOS?
„Dlouho jsem používal mobilní telefon s Androidem. Nedávno jsem si pořídil nový s iOS.“
- Máte zkušenosti s létáním paraglidingu?
„Nemám.“
- Znáte službu XContest? Popř. k čemu ji používáte?
„Neznám.“

Průběh testování

Během procházení kroků scénáře testování uživatel narazil na několik problémů. Jedním z hlavních důvodů byla neznalost služby. Uživatel nemohl nalézt název soutěže, ve které se nachází a měl problém s nalezením možnosti přepnutí soutěže. Problém nastal i při úkolu zobrazení letů pilota v letovém deníku a přepnutí soutěže, protože v dané soutěži nebyl uživatel přihlášen. Zbylé úkoly uživatel prošel bez větších obtíží.

Výstupní dotazník

- Jak byste celkově ohodnotili svou zkušenost s používáním aplikace? Narazili jste během testování na nějaké problémy?
„Hezká aplikace. Jediné co mě vadilo bylo přepínání soutěží. Z aplikací, které se věnují fotbalu jsem zvyklý na přepínání soutěží v bočním menu.“
- Které funkce se vám líbily nejvíce a proč? Které naopak méně a proč?
„Letový deník a zobrazení detailu těch letů. Je to podrobný a to se mi líbí. To přepínání soutěží bych dal do sekce soutěží a nedával to do nastavení.“
- Chyběla vám nějaká důležitá funkce nebo možnost, kterou jste v aplikaci postrádali?
„Asi ne, nic mě nenapadá.“
- Jak byste zlepšili nebo změnil uživatelské rozhraní a ovládání aplikace?
„Nic mě nenapadá. Přijde mi to fajn.“
- Máte nějaké další návrhy nebo připomínky, které by mohly pomoci zlepšit aplikaci?
„Nemám. Když se změní to přepínání soutěží, tak bych byl spokojený.“

6.2.4.6 Uživatel 6

Vstupní dotazník

- Studujete nebo jste vystudoval vysokou školu? Pokud ano, jakou?
„Ano, studuji na magisterském oboru FIT ČVUT.“
- Jste v současné době zaměstnán? Pokud ano, co je náplní vaší práce?
„Ano, pracuji na part-time jako full-stack developer.“
- Používáte telefon s operačním systémem Android nebo IOS?
„iOS.“
- Máte zkušenosti s létáním paraglidingu?
„Nemám“
- Znáte službu XContest? Popř. k čemu ji používáte?
„Neznám.“

Průběh testování

Testovací uživatel neměl během celého testovacího scénáře žádné potíže. Všechny kroky scénáře rychle pochopil a intuitivně se pohyboval v rozhraní aplikace.

Výstupní dotazník

- Jak byste celkově ohodnotili svou zkušenost s používáním aplikace? Narazili jste během testování na nějaké problémy?
„Bylo to dobrý. Během testování se objevily věci, které mě trochu překvapily. Například u registrace profilu je delší odezva a já nevěděl, zda se něco děje nebo zda jsem jen špatně stisknul tlačítko. A když jsem zadal špatné heslo, neměl jsem dostatek času si pročíst chybové hlášení.“
- Které funkce se vám líbily nejvíce a proč? Které naopak méně a proč?
„Nejvíce se mi líbí detail letu. Je to přehledné se všemi daty a tabulkami. Naopak nejméně si mi asi líbilo přepínání soutěží. Nejsem uživatel stránek XContest tak nevím, jak často se ta soutěž mění, ale pokud často, čekal bych ji na dostupnějších místě.“
- Chyběla vám nějaká důležitá funkce nebo možnost, kterou jste v aplikaci postrádali?
„Obrazovka s letovým deníkem by chtěla asi trochu upravit. Dokážu si představit, že až bych měl více letů, začne to být nepřehledné. Přidal bych filtry, podobně jako je tomu u soutěží.“
- Jak byste zlepšili nebo změnili uživatelské rozhraní a ovládání aplikace?
„Přepínání soutěží. To jsem již zmiňoval. Ale jinak se mi aplikace líbí.“
- Máte nějaké další návrhy nebo připomínky, které by mohly pomoci zlepšit aplikaci?
„To by věděli spíše uživatelé kteří službu používají. Pro mě je to dostačující. Taková informační aplikace. Ale dokážu si představit, že by přibyla funkce nahrávání letu.“

6.2.4.7 Uživatel 7

Vstupní dotazník

- Studujete nebo jste vystudoval vysokou školu? Pokud ano, jakou?
„Vystudoval jsem magisterský obor na FIT ČVUT.“
- Jste v současné době zaměstnán? Pokud ano, co je náplní vaší práce?
„Pracuji jako Android vývojář.“
- Používáte telefon s operačním systémem Android nebo IOS?
„Android“
- Máte zkušenosti s létáním paraglidingu?
„Ano, létám.“
- Znáte službu XContest? Popř. k čemu ji používáte?
„Ano. Používám ji pro trackování letů a zároveň pro sledování ostatních pilotů.“

Průběh testování

Testovací uživatel neměl během celého testovacího scénáře žádné potíže. Všechny kroky scénáře rychle pochopil a intuitivně se pohyboval v rozhraní aplikace.

Výstupní dotazník

- Jak byste celkově ohodnotili svou zkušenost s používáním aplikace? Narazili jste během testování na nějaké problémy?
„Aplikace mi přijde hezká. Nejsem si jistý s přepínáním soutěží. Dal bych to asi na více míst, aby to bylo více intuitivní pro uživatele. Já znám XContest tak mi pohyb v aplikaci nedělal problém.“
- Které funkce se vám líbily nejvíce a proč? Které naopak méně a proč?
„Mně se líbilo zpracování soutěží a detailu letu, protože to zobrazení na webu je nepřehledný. Do deníku se člověk popravdě tak často nedívá, jako právě do žebříčku soutěží. Trochu se mi nelíbí, že jsou novinky jak první obrazovka aplikace, rád bych tam viděl ty soutěže a novinky posunul někam jinam v aplikaci.“
- Chyběla vám nějaká důležitá funkce nebo možnost, kterou jste v aplikaci postrádali?
„Na obrazovce soutěží bych přidal i záložku dnes, protože všechny lety uživatele málokdy zajímají. Líbila by se mi tam možnost filtrovat podle místa. To používám jako inspiraci, jak v dané lokalitě létat. Dále bych určitě přidal mapu letu.“
- Jak byste zlepšili nebo změnil uživatelské rozhraní a ovládání aplikace?
„Aplikace se mi líbí. Líbí se mi zvolené barvy. V deníku bych možná změnil indikaci nepřihlášeného letu na červený text, že tento let není přihlášen v soutěži. Ono se tady ta funkce moc často nepoužívá. Dále bych přidal validaci hesla a možnost zobrazení hesla. Občas někde chyběly načítací obrazovky.“
- Máte nějaké další návrhy nebo připomínky, které by mohly pomoci zlepšit aplikaci?
„Zaměřil bych si na zobrazení soutěží. To vnímám jako klíčovou funkci. Jinak si myslím, že více aplikace nepotřebuje. Obsahuje vše, co na webu většinou používám a myslím, že každý uživatel by byl šťastný.“

6.2.4.8 Uživatel 8

Vstupní dotazník

- Studujete nebo jste vystudoval vysokou školu? Pokud ano, jakou?
„*Ne, mám vystudovanou elektrotechnickou střední školu.*”
- Jste v současné době zaměstnán? Pokud ano, co je náplní vaší práce?
„*Podnikám a provozuji vydavatelskou činnost.*”
- Používáte telefon s operačním systémem Android nebo IOS?
„*Android.*”
- Máte zkušenosti s létáním paraglidingu?
„*Ano, jsem pilot.*”
- Znáte službu XContest? Popř. k čemu ji používáte?
„*Ano, znám. Používám k zaznamenávání a vyhodnocování letu.*”

Průběh testování

Uživatel narazil na problém při přepínání soutěže. Při poskytnutí nápovědy soutěž přepnul. Zbylé kroky v testovacím scénáři prošel bez zaváhání.

Výstupní dotazník

- Jak byste celkově ohodnotili svou zkušenost s používáním aplikace? Narazili jste během testování na nějaké problémy?
„*Je to hezká aplikace, přehledná. Rozhodně lepší než hledat let na stránkách XContest. Díky této aplikaci člověk získá rychlý přehled toho, co potřebuje vědět, tzn. svoje přelety a přelety ostatních.*”
- Které funkce se vám líbily nejvíce a proč? Které naopak méně a proč?
„*Nejvíce se mi líbí zobrazení letového deníku a možnost prohlížet si své přihlášené lety. Myslím, že je tam vše, co je potřeba.*”
- Chyběla vám nějaká důležitá funkce nebo možnost, kterou jste v aplikaci postrádali?
„*Přidal bych mapu a zobrazení trajektorie letu. A potom bych přidal seznam oblíbených pilotů, abych mohl sledovat své kamarády a nemusel je složitě hledat. Mohlo by mi chodit i oznámení o tom, že někde letěli nebo že aktuálně letí, to by byla zajímavá funkce.*”
- Jak byste zlepšili nebo změnili uživatelské rozhraní a ovládání aplikace?
„*Asi bych potřeboval delší čas používat aplikaci. Takhle z prvního dojmu mně to přijde dobrý.*”
- Máte nějaké další návrhy nebo připomínky, které by mohly pomoci zlepšit aplikaci?
„*To by také nejspíše vyplynulo z delšího používání. Služba XContest a její používání je chaotický a náročný, ale díky této aplikaci by se to stalo přehlednější a jednodušší na používání. Pokud tato aplikace nahradí webové stránky XContest a já si budu moci prohlížet svoje lety a lety svých kamarádů, byl bych spokojený.*”

6.2.5 Vyhodnocení uživatelského testování

Uživatelské testování XCAApp se zúčastnili uživatelé různého zaměření a někteří testovací uživatelé neměli se službou XContest předchozí zkušenost. Cílem uživatelského testování aplikace XCAApp bylo identifikovat chyby v uživatelském rozhraní, zhodnotit celkový uživatelský zážitek při používání aplikace a získat návrhy na vylepšení a další směřování vývoje aplikace.

Všichni testovaní uživatelé hodnotili prototyp aplikace XCAApp kladně a v mnoha názorech a nalezených nedostatcích v uživatelském rozhraní se jejich hodnocení překrývala. Žádná závažná chyba nebyla během testování nalezena. Uživatelé, kteří mají zkušenosti s vývojem mobilních aplikací, si všimli rychlé odezvy a chybějících načítacích obrazovek, díky kterým by byla aplikace více plynulá. Řešením by bylo přidání časovače, který by na obrazovkách nechal načítací obrazovku zobrazenou delší dobu.

Při procházení scénáře testování narazili uživatelé na několik problémů, které byli spojené s návrhem rozhraní aplikace. Mezi hlavní změny, které z testování vyplynuly, patří:

- změnit možnost změnit soutěž v aplikaci,
- změnit název obrazovky *Soutěže* na *Soutěž*,
- přidat další textové pole, pro zopakování zadání hesla,
- přidat možnost zobrazit heslo, které uživatel zadal,
- přidání kontroly při odhlašování, aby se uživatel neodhlásil omylem,
- přidání kontroly při odebírání přeletu ze soutěže,
- zadávání data narození, pohlaví a státu pomocí pickru,
- nechat zobrazené chybové hlášení při registraci, přihlašování nebo změně hesla déle,
- zvětšení dotykové plochy některých tlačítek.

Většina navrhovaných změn je snadno proveditelná. Největším a nejvíce diskutovaným problémem, který měli uživatelé během testování, bylo přepínání soutěží. Nabízejí se dvě možná řešení:

1. stisknutí názvu soutěže na obrazovce novinek, letového deníku a žebříčku soutěže, přeměruje uživatele na obrazovku nastavení, aby mohl soutěž přepnout,
2. místo názvu soutěže se bude nalézat picker pro výběr soutěže a uživatel by si ji mohl měnit z výše vyjmenovaných obrazovek.

Díky provedenému uživatelskému testování se podařilo identifikovat slabé i silné stránky ve zpracování rozhraní aplikace. Uživatelé kladně hodnotili zpracování detailu letu, jednoduché zpracování žebříčku soutěže a snadný přístup k letovému deníku. Naopak, letový deník by mohl obsahovat jednoduchý filtr pro procházení letů a měl by být seřazen podle data letu. V současné době je řazen podle výše bodů.

Na základě zpětné vazby testovacích uživatelů jsou navržena vylepšení, která by měla vylepšit uživatelské rozhraní aplikace. Ideálním řešením, pro které v této práci již nebyl prostor, by bylo opravení chyb, implementace navržených změn a provést další kolo uživatelského testování, které by ověřilo, zda navržené změny skutečně vedou ke zlepšení použitelnosti a uživatelského zážitku z používání aplikace.

Během uživatelského testování měli uživatelé možnost navrhnout další funkce, které by v aplikaci rádi měli, a jakým směrem by aplikace mohla směřovat. Doporučenému vývoji a směřování aplikace se věnuje následující kapitola.

Zhodnocení aplikace a doporučení dalšího vývoje

Výsledkem práce je multiplatformní mobilní aplikace, která umožňuje uživatelům pohodlně spravovat své letové deníky, prohlížet informace o soutěžích a upravovat své uživatelské profily přímo z mobilních zařízení. Architektura aplikace je navržena pro platformu iOS i Android, čímž je zajištěna široká dostupnost pro cílovou skupinu. Zároveň je aplikace dostupná v českém i anglickém jazyku.

Díky implementovanému prototypu mobilní aplikace a provedenému uživatelskému testování lze identifikovat výhody i nevýhody aplikace, které přináší. Mezi výhody aplikace, které z testování uživatelů vyplynuly, patří:

- snadnost použití aplikace oproti webovému řešení,
- minimalistický přístup k designu a funkcionalitám zajišťuje přehlednost a jednoduchost rozhraní,
- používání aplikace je snazší než používání webového řešení,
- dostupnost na platformách iOS a Android pokrývá velkou část cílové skupiny uživatelů.

Z uživatelského testování nevyplývaly žádné závažné nevýhody, které aplikace přináší. Jedna z možných potenciálních nevýhod aplikace je absence funkcí aplikace v porovnání s webovým řešením. To ovšem na druhou stranu dělá aplikaci přehlednější a zvyšuje dostupnost klíčových vlastností. Mezi nevýhody, které mají obecně mobilní aplikace oproti webovému řešení, je nutnost vyvíjet službu na více platformách. V tomto případě udržovat současné webové řešení a vyvíjet multiplatformní mobilní aplikaci pro iOS a Android.

Nasazení této mobilní aplikace do provozu přinese řadu přínosů nejen pro cílovou skupinu uživatelů, ale i pro službu XContest. Především zvýšení konkurenceschopnosti oproti alternativním řešením na trhu a následně k rozšíření uživatelské základny. Uživatelé získají pohodlnější a přístupnější způsob správy svých letových záznamů, prohlížení žebříčků soutěží a správy uživatelských profilů přímo z mobilních zařízení. Zároveň to umožní zpřístupnit klíčové funkce širšímu publiku, včetně méně zkušených pilotů nebo těch, kteří doposud neměli možnost efektivně spravovat své letové záznamy.

Návrh architektury aplikace je vytvořen s ohledem na snížení potřebných nákladů pro vývoj mobilní aplikace XCAApp. Nutnost vývojářů pro iOS a Android je kompenzována sdílenou částí kódu, která se využívá mimo prezentační vrstvu aplikace. To vede k minimalizaci opakování kódu a zrychlení vývoje aplikace. Zároveň se tak zjednodušuje a snižuje náročnost práce vývojářů,

kteří se tak mohou plně věnovat tvorbě uživatelského rozhraní. Mezi další výhody patří rychlejší testování, snadná rozšiřitelnost a jednodušší údržbu.

Výhody mobilní aplikace XCAApp převyšují její nevýhody a potenciální dodatečné náklady na vývoj a údržbu.

7.1 Doporučení dalšího vývoje aplikace

Uživatelé během uživatelského testování hodnotili aplikaci kladně a ocenili její funkčnost a užitečnost v praxi. Zaznamenaný zájem ze strany uživatelů zdůrazňuje značný potenciál aplikace a podporuje její další rozvoj.

Mezi potenciální nové vlastnosti, které by rozšířili současné funkcionality v aplikaci, patří:

- přidání mapy s trajektorií letu,
- přidání grafů popisující výšku a další parametrů během letu,
- přidání filtrů do letového deníku pro snazší orientaci v přidaných letech,
- přidání více filtrů do soutěží, aby mohli uživatelé vyhledávat lety podle lokace nebo pilota,
- přidání podpory více soutěží a možnosti se do nich přihlašovat,
- přidání možnosti sledování přeletu podobně jako má aplikace XCTrack.

Pro nasazení aplikace do obchodu s aplikacemi je nezbytná implementace pro platformu Android. Vzhledem k již vytvořenému uživatelskému rozhraní na platformě iOS a implementované sdílené části kódu v KMP by tato implementace neměla být časově náročná.

Další nutnou podmínkou pro nasazení aplikace je získání API od služby XContest. Současný prototyp využívá data z databáze Firestone, která však nejsou napojena na webové řešení.

7.2 Přínosy aplikace

Tato část bakalářské se zaměřuje na hlavní přínosy aplikace XCAApp pro své cílové uživatele, mezi které se řadí piloti paraglidingu a uživatelé služby XContest. Aplikace XCAApp usnadňuje a zefektivňuje využívání služby XContest a pomáhá uživatelům při správě letového deníku.

Piloti paraglidingu čelí mnohým výzvám, které jsou spojeny s létáním a využíváním služby XContest. Mezi tyto výzvy patří:

- nepřehledné rozhraní webového řešení,
- nepohodlné ovládání na mobilních zařízeních,
- nedostatek času,
- nutnost pořízení zařízení, které podporuje nahrávání přeletů.

Aplikace XCAApp se snaží tyto výzvy ulehčit. Nepřehledné rozhraní webového řešení způsobuje, že je pro uživatele orientace a užívání této služby náročné a vyčerpávající. Kvůli nepohodlnému ovládání na mobilních zařízeních jsou uživatelé nuceni používat službu výhradně na desktop zařízeních. Proto aplikace XCAApp přináší jednoduché rozhraní nejpoužívanějších funkcionalit, které jsou také dostupné na webovém řešení. Tím se snižuje náročnost využívání této služby a má pozitivní vliv na uživatelský zážitek z využívání služby, který je zaznamatelný již při prvním použití aplikace.

Paragliding představuje náročný koníček s vysokými nároky na časovou investici a pečlivé plánování. Piloti tráví značné množství času přípravou, samotným letem a následnou dokumentací. Mobilní aplikace XCApp pro paragliding přináší značné zjednodušení těchto procesů. Eliminuje nutnost přenášení notebooků do terénu a umožňuje pilotům efektivní správu veškerých aspektů letu přímo z mobilního zařízení. Nahrávání letů, vedení deníku a provádění analýz v reálném čase zabere pouhé minuty, čímž se uvolňuje drahocenný čas pro samotné létání. Implementace mobilní aplikace tak posouvá paragliding na novou úroveň dostupnosti a atraktivity, čímž jej otevírá širšímu publiku.

Nutnost pořízení zařízení, které podporuje nahrávání přeletů (například Android zařízení s aplikací XCTrack), zvyšuje celkovou finanční náročnost pro piloty. Díky tomu, že je aplikace multiplatformní, nejsou uživatelé nuceni využívat několik zařízení, aby byli schopni využívat službu XContest a nahrávali si své přelety. To uživatelům ušetří několik tisíc korun.

Mobilní aplikace XCApp je cenným nástrojem pro piloty paraglidingu, který jim umožňuje správu letového deníku, prohlížení přeletů a správu uživatelského profilu. Aplikace má velký potenciál zjednodušit používání služby a snížit časové i finanční nároky uživatelům.

Kapitola 8

Závěr

Hlavním cílem této bakalářské práce bylo navrhnout a implementovat mobilní aplikaci, která by uživatelům poskytovala funkce pro správu letového deníku, prohlížení informací o soutěžích a správu uživatelského profilu. Tímto způsobem měla aplikace doplnit stávající webové řešení a zpřístupnit tyto funkce uživatelům na mobilních zařízeních.

K dosažení vytyčeného cíle byla provedena analýza stávajícího webového řešení. Během analýzy byl popsán princip fungování uživatelského rozhraní, jeho funkcionality a identifikace klíčových vlastností. Dále byla provedena analýza alternativních aplikací v dané oblasti, která sloužila a získání celkového přehledu na trhu mobilních aplikací. Díky nabytým znalostem a informacím byli vytvořeny funkční požadavky a případy užití.

Dalším krokem bylo vytvoření návrhu optimálního řešení aplikace. Během návrhu bylo nalezeno vhodné řešení pro vývoj multiplatformní aplikace, které bude jednoduché na vývoj i údržbu a zároveň bude umožňovat využít plný potenciál mobilních zařízení, na kterých aplikace poběží. Díky tomu bylo možné navrhnout vhodnou architekturu, která umožňuje vyvinout udržitelnou a škálovatelnou aplikaci. Před implementací byli vybrány vhodné technologie a nástroje, které zajistili efektivní a bezpečný vývoj mobilní aplikace.

V implementační části je realizováno podle návrhu a zvolených technologií finální řešení. To se skládá ze sdílené části, která zajišťuje přístup k datům v databázi a jejich následné zpracování. Prezentační vrstva, která je platformně závislá, byla implementována pouze pro platformu iOS a prezentační vrstva pro platformu Android je připravena k implementaci. Během vývoje mobilní aplikace se objevilo několik problémů. Hlavním problémem bylo nezískání přístupu k API webového řešení. Kvůli tomu se muselo využít znalosti z analýzy a dodatečně analyzovat a připravit data, která jsou na stránkách webového řešení dostupná. Kvůli času, který byl věnován do analýzy a návrhu nebyl dostatek času implementovat veškeré funkční požadavky a případy užití, které byly vytvořeny. Přesto byla většina z nich implementována.

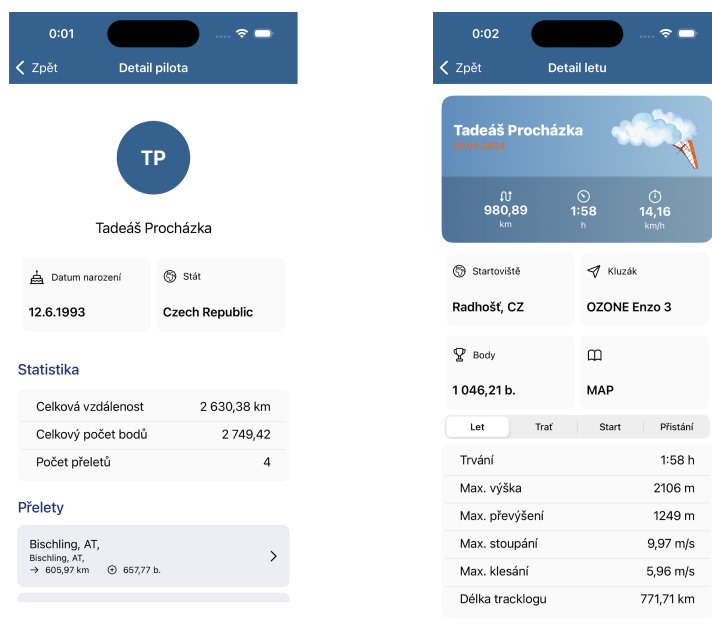
Ve vytvořeném prototypu mobilní aplikace byli otestovány případy užití. Tyto testy otestovaly velkou část funkcionalit aplikace. Dále bylo provedeno hloubkové uživatelské testování, které odhalilo několik nedostatků v rozhraní aplikace a byla získána cenná zpětná vazba pro další vývoj a směřování aplikace.

Na závěr bakalářské práce byli vyhodnoceny výhody a nevýhody vyvinuté mobilní aplikace pro paragliding, proveden ekonomický a business dopad aplikace na uživatele a pro společnost, která by aplikaci nasadila do provozu a dále vyvíjela po boku současného webového řešení, a doporučeny další kroky při vývoji mobilní aplikace.

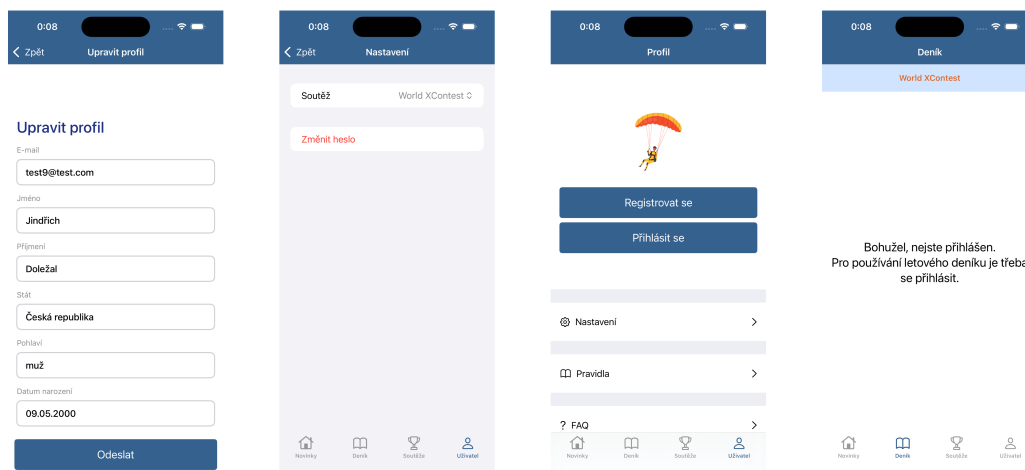
Ukázka uživatelského rozhraní aplikace



■ **Obrázek A.1** Ikona aplikace na ploše a spouštěcí obrazovka prototypu aplikace XCAApp



■ Obrázek A.2 Ukázka profilu pilota a detailu letu



■ Obrázek A.3 Ukázka obrazovky změnu údajů profilu, nastavení, obrazovky profilu a deníku pro nepřihlášeného uživatele

Bibliografie

1. *History of Paragliding and Speedflying* — *ushpa.org* [online]. USHPA, 2023. Dostupné také z: <https://www.ushpa.org/page/history-of-paragliding.aspx>. [Accessed 13-02-2024].
2. BHARATH. *The Clean Architecture — Beginner's Guide* — *betterprogramming.pub* [online]. 2022. Dostupné také z: <https://betterprogramming.pub/the-clean-architecture-beginners-guide-e4b7058c1165>. [Accessed 21-04-2024].
3. MARTIN, Robert C. *Clean Coder Blog* — *blog.cleancoder.com* [online]. 2012. Dostupné také z: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>. [Accessed 21-04-2024].
4. GALLARDO, Estefanía García. *What Is MVVM Architecture?* — *builtin.com* [online]. 2023. Dostupné také z: <https://builtin.com/software-engineering-perspectives/mvvm-architecture>. [Accessed 23-04-2024].
5. UGAYA40. *File:MVVMPattern.png - Wikimedia Commons* — *commons.wikimedia.org* [online]. 2012. Dostupné také z: <https://commons.wikimedia.org/w/index.php?curid=19056842>. [Accessed 23-04-2024].
6. KIM, Geoffrey. *Mastering MVVM: A Comprehensive Guide to the Model-View-ViewModel Architecture* — *dev.to* [online]. 2023. Dostupné také z: <https://dev.to/mochafreddo/mastering-mvvm-a-comprehensive-guide-to-the-model-view-viewmodel-architecture-221g>. [Accessed 23-04-2024].
7. AMMANAVAR, Anup. *SwiftUI — MVI Architecture* — *betterprogramming.pub* [online]. 2020. Dostupné také z: <https://betterprogramming.pub/mvi-architecture-for-swiftui-apps-cff44428394>. [Accessed 23-04-2024].
8. SATRIAWAN, M Yogi. *Mastering the MVI Architecture: A Comprehensive Guide* — *satriawaaan* [online]. 2023. Dostupné také z: <https://medium.com/@satriawaaan/mastering-the-mvi-architecture-a-comprehensive-guide-28b2cafee1c9>. [Accessed 23-04-2024].
9. SATRIAWAN, M Yogi. *MVI architecture design* [online]. 2023. Dostupné také z: https://miro.medium.com/v2/resize:fit:1400/format:webp/1*P5B0Cv9ftI1rMgCC2p4Uzw.png. [Accessed 24-04-2024].
10. ASHRAFI, Siamak (Ash). *SwiftUI is Model View Intent (MVI)* — *zoewave.medium.com* [online]. 2019. Dostupné také z: <https://zoewave.medium.com/swiftui-is-model-view-intent-mvi-fd142b12fc81>. [Accessed 23-04-2024].
11. KARIA, Bhavya. *A quick intro to Dependency Injection: what it is, and when to use it* — *freecodecamp.org* [online]. 2018. Dostupné také z: <https://www.freecodecamp.org/news/a-quick-intro-to-dependency-injection-what-it-is-and-when-to-use-it-7578c84fa88f/>. [Accessed 10-05-2024].

12. SAMUEL OLORUNTOBA, Anish Singh Walia. *SOLID: The First 5 Principles of Object Oriented Design* | DigitalOcean — *digitalocean.com* [online]. 2024. Dostupné také z: <https://www.digitalocean.com/community/conceptual-articles/s-o-l-i-d-the-first-five-principles-of-object-oriented-design#dependency-inversion-principle>. [Accessed 10-05-2024].
13. *Documentation* — *hmlongco.github.io* [online]. [B.r.]. Dostupné také z: <https://hmlongco.github.io/Factory/documentation/factory/>. [Accessed 10-05-2024].
14. *Documentation* — *hmlongco.github.io* [online]. [B.r.]. Dostupné také z: <https://hmlongco.github.io/Factory/documentation/factory/gettingstarted>. [Accessed 10-05-2024].
15. *Flutter architectural overview* — *docs.flutter.dev* [online]. [B.r.]. Dostupné také z: <https://docs.flutter.dev/resources/architectural-overview>. [Accessed 23-04-2024].
16. SHIVANANDHAN, Manish. *How to Build Mobile Apps with Flutter* — *freecodecamp.org* [online]. 2023. Dostupné také z: <https://www.freecodecamp.org/news/how-to-build-mobile-apps-with-flutter/>. [Accessed 23-04-2024].
17. *Flutter: Advantages, Disadvantages and Future Scopes - GeeksforGeeks* — *geeksforgeeks.org* [online]. 2023. Dostupné také z: <https://www.geeksforgeeks.org/flutter-advantages-disadvantages-and-future-scopes/>. [Accessed 23-04-2024].
18. *Dart overview* — *dart.dev* [online]. [B.r.]. Dostupné také z: <https://dart.dev/overview>. [Accessed 23-04-2024].
19. STAFF, Coursera. *What Is React Native? Beginner's Guide + FAQ* — *coursera.org* [online]. 2023. Dostupné také z: <https://www.coursera.org/articles/what-is-react-native>. [Accessed 24-04-2024].
20. JETBRAINS. *The Six Most Popular Cross-Platform App Development Frameworks* — *kotlinlang.org* [online]. 2024. Dostupné také z: <https://kotlinlang.org/docs/cross-platform-frameworks.html#react-native>. [Accessed 24-04-2024].
21. KAMIENSKI, Norbert. *React Native Pros and Cons - Pagepro* — *pagepro.co* [online]. 2023. Dostupné také z: <https://pagepro.co/blog/react-native-pros-and-cons/>. [Accessed 24-04-2024].
22. WILLIAMS, Alexander T. *The Pros and Cons of Using React Today* — *thenewstack.io* [online]. 2023. Dostupné také z: <https://thenewstack.io/the-pros-and-cons-of-using-react-today/>. [Accessed 24-04-2024].
23. JETBRAINS. *Kotlin Multiplatform* — *kotlinlang.org* [online]. 2024. Dostupné také z: <https://kotlinlang.org/docs/images/kotlin-multiplatform.svg>. [Accessed 24-04-2024].
24. JETBRAINS. *The Six Most Popular Cross-Platform App Development Frameworks* — *Kotlin Multiplatform* — *kotlinlang.org* [online]. 2024. Dostupné také z: <https://kotlinlang.org/docs/cross-platform-frameworks.html#kotlin-multiplatform>. [Accessed 24-04-2024].
25. *Compose Multiplatform UI Framework | JetBrains* — *jetbrains.com* [online]. 2024. Dostupné také z: <https://www.jetbrains.com/lp/compose-multiplatform/>. [Accessed 24-04-2024].
26. BERGMAN, Per-Erik. *Unlocking Cross-Platform Development with Kotlin Multiplatform Mobile (KMM)* — *pererikbergman* [online]. 2023. Dostupné také z: <https://medium.com/@pererikbergman/unlocking-cross-platform-development-with-kotlin-multiplatform-mobile-kmm-31e4df9a53ec>. [Accessed 24-04-2024].
27. *Kotlin overview | Android Developers* — *developer.android.com* [online]. 2023. Dostupné také z: <https://developer.android.com/kotlin/overview>. [Accessed 24-04-2024].
28. *Firebase | Google's Mobile and Web App Development Platform* — *firebase.google.com* [online]. [B.r.]. Dostupné také z: <https://firebase.google.com>. [Accessed 10-05-2024].

29. *Firestore* | *Firebase* — *firebase.google.com* [online]. 2024. Dostupné také z: <https://firebase.google.com/docs/firestore?authuser=0&hl=en>. [Accessed 10-05-2024].
30. *Cloud Storage for Firebase* — *firebase.google.com* [online]. 2024. Dostupné také z: <https://firebase.google.com/docs/storage>. [Accessed 10-05-2024].
31. *Firebase Authentication* — *firebase.google.com* [online]. 2024. Dostupné také z: <https://firebase.google.com/docs/auth>. [Accessed 10-05-2024].
32. KUMAR, Ranjith. *Jetpack Compose Introduction* — *jetpackcompose.net* [online]. 2021. Dostupné také z: <https://www.jetpackcompose.net/jetpack-compose-introduction>. [Accessed 29-04-2024].
33. *Why Compose* | *Jetpack Compose* | *Android Developers* — *developer.android.com* [online]. 2024. Dostupné také z: <https://developer.android.com/develop/ui/compose/why-adopt>. [Accessed 29-04-2024].
34. WOLF, Russell. *GitHub - russhwolf/multiplatform-settings: A Kotlin Multiplatform library for saving simple key-value data* — *github.com* [online]. [B.r.]. Dostupné také z: <https://github.com/russhwolf/multiplatform-settings>. [Accessed 10-05-2024].
35. SQUARE, Inc. *Overview - SQLDelight* — *cashapp.github.io* [online]. [B.r.]. Dostupné také z: <https://cashapp.github.io/sqldelight/2.0.1/>. [Accessed 09-05-2024].
36. *About SQLite* — *sqlite.org* [online]. 2023. Dostupné také z: <https://www.sqlite.org/about.html>. [Accessed 09-05-2024].
37. APPLE. *Swift Has Reached 1.0 - Swift Blog* — *developer.apple.com* [online]. 2014. Dostupné také z: <https://developer.apple.com/swift/blog/?id=14>. [Accessed 29-04-2024].
38. APPLE. *Swift - Apple Developer* — *developer.apple.com* [online]. [B.r.]. Dostupné také z: <https://developer.apple.com/swift/#modern>. [Accessed 09-05-2024].
39. APPLE. *SwiftUI Overview - Xcode - Apple Developer* — *developer.apple.com* [online]. [B.r.]. Dostupné také z: <https://developer.apple.com/xcode/swiftui/>. [Accessed 09-05-2024].
40. CELIS, Sabastina. *GitHub - scelis/twine: Twine is a command line tool for managing your strings and their translations.* — *github.com* [online]. [B.r.]. Dostupné také z: <https://github.com/scelis/twine>. [Accessed 10-05-2024].
41. *XCTest* | *Apple Developer Documentation* — *developer.apple.com* [online]. [B.r.]. Dostupné také z: <https://developer.apple.com/documentation/xctest>. [Accessed 16-05-2024].
42. FRYEO. *Top 5 best paragliding apps for your phone* — *Flyeo* — *flyeo.com* [online]. 2023. Dostupné také z: <https://flyeo.com/en/top-5-mobile-applications-for-paragliding/>. [Accessed 25-03-2024].
43. XCONTEST. *World XContest 2024 [home] :: XContest.org* — *world of XC paragliding* — *xcontest.org* [online]. 2024. Dostupné také z: <https://www.xcontest.org/world/en/>. [Accessed 26-03-2024].
44. PARAGLIDINGU, Česká Liga XC. *UP Krkonose open 2024 :: XContest.org* — *world of XC paragliding* — *xcontest.org* [online]. 2024. Dostupné také z: <https://www.xcontest.org/krkonose/>. [Accessed 26-04-2024].
45. KAMPELMÜHLER, Moritz. *Moritz Kampelmühler • 20.3.2024 • 62.22 km* — *xcontest.org* [online]. 2024. Dostupné také z: <https://www.xcontest.org/world/en/flights/detail:moka/20.03.2024/10:36>. [Accessed 04-04-2024].
46. DOLEŽAL, Jindřich. *MyProfile* — *xcontest.com* [online]. 2024. Dostupné také z: <https://www.xcontest.org/world/en/profile>. [Accessed 04-04-2024].
47. FLYSKYHY. *Flyskyhy* — *apps.apple.com* [online]. 2024. Dostupné také z: <https://apps.apple.com/us/app/flyskyhy/id516879039>. [Accessed 26-03-2024].

48. *Flyskyhy Soaring App* — *flyskyhy.com* [online]. Flyskyhy, 2023. Dostupné také z: <http://www.flyskyhy.com/index.html>. [Accessed 13-03-2024].
49. RECORDER, Gaggles - Flight. *Gaggles - Flight Recorder* — *flygaggles.com* [online]. 2024. Dostupné také z: <https://www.flygaggles.com/help/items/home-overview/>. [Accessed 26-03-2024].
50. RECORDER, Gaggles - Flight. *Gaggles - Flight Recorder* — *flygaggles.com* [online]. 2024. Dostupné také z: <https://www.flygaggles.com/help/items/flight-logs-manage/>. [Accessed 26-03-2024].
51. RECORDER, Gaggles - Flight. *Gaggles - Flight Recorder* — *flygaggles.com* [online]. 2024. Dostupné také z: <https://www.flygaggles.com/help/items/flight-logs-recording-overview/>. [Accessed 26-03-2024].
52. RECORDER, Gaggles - Flight. *Gaggles - Flight Recorder* — *flygaggles.com* [online]. 2024. Dostupné také z: <https://www.flygaggles.com/help/items/achievements/>. [Accessed 26-03-2024].
53. RECORDER, Gaggles - Flight. *Home* — *flygaggles.com* [online]. 2023. Dostupné také z: <https://www.flygaggles.com>. [Accessed 14-03-2024].
54. BLACKBOX, Rogallo. *Rogallo BlackBox* — *apps.apple.com* [online]. 2024. Dostupné také z: <https://apps.apple.com/us/app/rogallo-blackbox/id1450278553>. [Accessed 26-03-2024].
55. *Rogallo BlackBox | Paragliding Flight Log App* — *rogallo.app* [online]. Bastian Roessler, 2021. Dostupné také z: <https://rogallo.app>. [Accessed 14-03-2024].
56. STATISTA. *Number of smartphone users 2014-2029 | Statista* — *statista.com* [online]. 2024. Dostupné také z: <https://www.statista.com/forecasts/1143723/smartphone-users-in-the-world>. [Accessed 18-04-2024].
57. STATCOUNTER. *Mobile Operating System Market Share Worldwide | Statcounter Global Stats* — *gs.statcounter.com* [online]. 2024. Dostupné také z: <https://gs.statcounter.com/os-market-share/mobile/worldwide>. [Accessed 19-04-2024].
58. JETBRAINS. *What is cross-platform mobile development? | Kotlin* — *kotlinlang.org* [online]. 2024. Dostupné také z: <https://kotlinlang.org/docs/cross-platform-mobile-development.html#benefits-of-cross-platform-development>. [Accessed 19-04-2024].
59. JETBRAINS. *kotlinlang.org* [online]. 2024. Dostupné také z: <https://kotlinlang.org/docs/images/survey-results-q1-q2-22.png>. [Accessed 19-04-2024].
60. *What is cross-platform mobile development? | Kotlin* — *kotlinlang.org* [online]. 2024. Dostupné také z: <https://kotlinlang.org/docs/cross-platform-mobile-development.html#kotlin-multiplatform>. [Accessed 19-04-2024].
61. *GitHub - GitLiveApp/firebase-kotlin-sdk: A Kotlin-first SDK for Firebase* — *github.com* [<https://github.com/GitLiveApp/firebase-kotlin-sdk/tree/master>]. [B.r.]. [Accessed 28-04-2024].

Obsah příloh

	readme.txt	stručný popis obsahu média
	src	
	impl	zdrojové kódy implementace
	thesis	zdrojová forma práce ve formátu \LaTeX
	text	text práce
	thesis.pdf	text práce ve formátu PDF