



Zadání bakalářské práce

Název:	Aplikace rugby - modul zákonného zástupce
Student:	Daniel Karlovský
Vedoucí:	Ing. Jiří Chludil
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

Aplikace rugby je mobilní aplikace pro OS Android zaměřující se na organizaci rugbyových klubů (tréninky, turnaje, soupisky hráčů, výzvy, atd...). Modul zákonného zástupce zahrnuje roli zástupce přiřaditelnou k účtu, možnosti registrace, správu svěřených hráčů (dětí) a systém k zasílání formulářů v rámci aplikace.

1. Analyzujte a popište možný způsob integrace role zákonného zástupce do vybrané aplikace.
2. Analyzujte požadavky trenérů a rodičů.
3. Analyzujte dostupné alternativní aplikace.
4. Na základě analýzy a pomocí metod softwarového inženýrství navrhnete prototyp modulu zákonného zástupce.
5. Implementujte prototyp modulu pro zákonného zástupce.
6. Implementovaný prototyp modulu podrobte vhodným integračním, akceptačním a uživatelským testům



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Bakalářská práce

Aplikace rugby - modul zákonného zástupce

Daniel Karlovský

Vedoucí práce: Ing. Jiří Chludil

13. května 2021

Poděkování

Chtěl bych poděkovat panu Ing. Jiřímu Chludilovi za vedení této práce, svým kolegům Martinu Paulovi, Janu Špetlovi a Matěji Ulmanovi za příjemnou spolupráci a svým rodičům, kteří mi byli při práci oporou. Bez všech těchto osob by dokončení této práce nebylo možné.

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

V Praze dne 13. května 2021

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2021 Daniel Karlovský. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Karlovský, Daniel. *Aplikace rugby - modul zákonného zástupce*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Abstrakt

Práce se zabývá analýzou, návrhem a implementací modulu do Aplikace rugby, nedokončené aplikace na platformě Android zaměřené na organizaci dětských rugby klubů, jejich turnajů a tréninků. Modul se soustředí na roli zákonných zástupců hráčů v těchto klubech, jejich pravomocí a obecné funkcionality registrací, správy účtů a zasílání žádostí členům klubu.

Výstupem této práce je zhotovená analýza existujícího prototypu, která obsahuje jeho detailní popis, seznam implementovaných funkcionalit, technologie použité v jeho vývoji a následné odvození funkčních a nefunkčních požadavků. Následující návrh zahrnuje zpracování daných požadavků do případů užití, průchodových aktivit, tvorbu tříd a plány jejich uživatelského rozhraní. Práce pokračuje implementací tohoto modulu a výsledky jeho uživatelských a akceptačních testů. Součástí implementace jsou také podrobné návody k instalaci a užití aplikace, spolu s programovými detaily užitečnými k rozšiřování funkcionalit modulu.

Klíčová slova Android, aplikace, účty, registrace, rugby, anonymizace, Kotlin

Abstract

This thesis deals with analysis, design and implementation of a certain module to Rugby app, an unfinished app on the Android platform centered around organising rugby clubs for children, their tournaments and trainings. The module focuses on the role of legal guardians of club players, their user privileges and general functionalities of registrations, account management and sending requests to club members.

The output of this thesis is a finished analysis of the existing prototype, which contains it's detailed description, list of it's implemented functionalities, technologies used during development and subsequent deduction of functional and non-functional requirements. Consequent design contains the processing of those requirements into use cases, main scenario activities, creation of classes and drafts for their user interface. The thesis continues with implementation of this module and outcomes of it's user and acceptance testing. Incorporated in the implementation are also detailed tutorials for installation and usage of the app, together with program details, useful for expanding module's functionalities.

Keywords Android, app, accounts, registration, rugby, anonymization, Kotlin

Obsah

Úvod	1
1 Cíl práce	3
2 Analýza	5
2.1 Existující prototyp aplikace	5
2.1.1 Architektura	5
2.1.2 Funkcionality	6
2.2 Použité technologie	10
2.3 Konkurence	14
2.3.1 TeamSnap	14
2.3.2 Spoluhráči.cz	15
2.3.3 Teamstuff	15
2.3.4 Týmuj	16
2.3.5 TeamApp	17
2.3.6 Sportnect	17
2.3.7 Výsledky konkurenční analýzy	19
2.4 Relevantní práce	19
2.4.1 FurryBall – správa licencí a uživatelů	19
2.4.2 BigCloud – modul obchodníka	20
2.4.3 Výuková aplikace Dráček II – gamifikace a personalizace	20
2.5 Aktéři	20
2.6 Funkční požadavky	21
2.6.1 F1 – Registrace	21
2.6.2 F2 – Formuláře	21
2.6.3 F3 – Role hlavního trenéra	21
2.6.4 F4 – Role zákonných zástupců	22
2.6.5 F5 – Možnost registrace hráče	22
2.6.6 F6 – Možnosti správy účtu hráče	22

2.6.7	F7 – Možnosti smazání účtu	22
2.6.8	F8 – Potvrzení registrace e-mailem	22
2.6.9	F9 – Externí registrace zákonného zástupce	23
2.6.10	F10 – Nouzový kontakt	23
2.7	Nefunkční požadavky	23
2.8	GDPR, anonymizace, pseudonymizace a mazání účtu	23
2.9	Komunikace se žadatelem	24
3	Návrh	25
3.1	Případy užití	25
3.1.1	Přihlašovací obrazovka	26
3.1.2	Formuláře	26
3.1.3	Hlavní menu	28
3.2	Aktivity uživatelů	29
3.2.1	Zasílání formulářů	29
3.2.2	Přihlášení	30
3.2.3	Registrace	31
3.2.4	Vytvoření hráče	34
3.2.5	Zrušení účtu	34
3.3	Životní cykly	36
3.4	Návrh tříd	37
3.4.1	Registrace	37
3.4.2	Formuláře	38
3.4.3	Role	39
3.5	Návrh uživatelského rozhraní	41
3.5.1	Přihlášení a prvotní heslo	41
3.5.2	Registrace	42
3.5.3	Funkce modulu přístupné z hlavního menu	43
4	Implementace	47
4.1	Změny oproti návrhu	47
4.1.1	Vynechané funkcionality	47
4.2	Instalační příručka	48
4.2.1	Serverová část	48
4.2.2	Mobilní klient	48
4.3	Uživatelská příručka	50
4.4	Programátorská příručka	56
4.4.1	Formuláře	56
4.4.1.1	Aplikační vrstva	56
4.4.1.2	Klient	59
4.4.2	Potvrzení přes e-mail	64
4.4.3	Externí registrace	64
4.4.4	Nouzový kontakt	66

5	Testování	69
6	Budoucí funkcionality	73
	Závěr	75
	Literatura	77
A	Diagramy	81
A.1	Zrušení účtu	82
B	Scénář uživatelských testů	85
C	Dotazníky	87
D	Seznam použitých zkratek	101
E	Obsah přiloženého média	103

Seznam obrázků

2.1	Aplikace rugby – doménový model	7
2.2	Aplikace rugby – přihlášení	8
2.3	Aplikace rugby – hlavní menu	8
2.4	Aplikace rugby – detail účtu	8
2.5	Aplikace rugby – tvorba turnaje	9
2.6	Aplikace rugby – zapisování turnaje	9
2.7	Android Activity – životní cyklus [1]	13
2.8	TeamSnap – ukázka funkcionalit [2]	14
2.9	TeamSnap – seznam týmů [3]	14
2.10	Spoluhráči.cz – hlavní stránka [4]	15
2.11	Teamstuff – statistiky týmu [5]	15
2.12	Týmuj – seznam událostí [6]	16
2.13	Týmuj – seznam členů a jejich rolí [6]	16
2.14	TeamApp – startovní obrazovka a hlavní menu [7]	17
2.15	Sportnect – stránka s událostí [8]	18
3.1	Use case – Kontrola hráčů	25
3.2	Use case – přihlašovací obrazovka	26
3.3	Use case – odesílání formulářů	27
3.4	Use case – příjem formulářů	27
3.5	Use case – hlavní menu	28
3.6	Diagram aktivit – formuláře	29
3.7	Diagram aktivit – přihlášení	30
3.8	Diagram aktivit – registrace	31
3.9	Diagram aktivit – interní registrace zástupce	32
3.10	Diagram aktivit – externí registrace zástupce	33
3.11	Diagram aktivit – tvorba hráče	35
3.12	Diagram aktivit – tvorba hráče	36
3.13	Diagram aktivit – tvorba hráče	36
3.14	Návrh tříd – starý účet	38

3.15	Návrh tříd – nový účet	38
3.16	Návrh tříd – Formuláře	39
3.17	Návrh tříd – informační formuláře	40
3.18	Návrh tříd – formuláře k rolím	40
3.19	Návrh tříd – role	41
3.20	Wireframe – prvotní přihlášení	42
3.21	Wireframe – registrace	43
3.22	Wireframe – přechody z hlavního menu	45
3.23	Wireframe – nastavení účtu	46
4.1	Úvodní obrazovka	50
4.2	Navigace k registraci a přihlášení	51
4.3	Hlavní menu	52
4.4	Nastavení	53
4.5	Zaslání formuláře a Inbox	54
4.6	Správa hráčů	55
5.1	Záběr z testování	71
A.1	Diagram aktivit – zrušení s dětmi	82
A.2	Diagram aktivit – zrušení bez dětí	83

Úvod

„Sport charakter nebuduje, ale odhaluje.“ [9]

Děti bývají vysoce energické a aktivní. Tuto energii si musejí nějak vybit, proto se rády hrnou do sportovních aktivit. Existují sportovní kluby, které se exkluzivně zaměřují na pořádání tréninků a turnajů spojených s těmito aktivitami. Ovšem organizace těchto klubů, pořádání sportovních událostí a uspokojení rodičovských požadavků jsou vysoce komplexní a pracovní náročné záležitosti. Ke zjednodušení těchto složitostí je vyvíjena Aplikace rugby.

Aplikace rugby je nedokončená interaktivní mobilní aplikace na platformě Android, která má být na míru přizpůsobena požadavkům rugby klubů, založena na spolupráci se členy sportovního klubu RC Říčany. Její funkce zahrnují tvorbu a správu sportovních klubů, pořádání zápasů, turnajů a tréninků, správa hráčů trenéry a rodiči, spolu s možnostmi plnění výzev hráči pro získání vizuálních úprav svých profilů.

Práce se soustředí na tvorbu a zavedení modulu pro existující prototyp této aplikace, který umožní rodičům danou aplikaci využívat ke kontrole aktivit svých dětí a interakci se členy klubu. Vývoj na této aplikaci je řešen v rámci týmu s následujícími kolegy a jejich moduly:

- Martin Paul – tréninky [10]
- Matěj Ulman – organizace turnajů, notifikace [11]
- Jan Špetl – výzvy [12]

Obsahově práce začíná analýzou, kde budou popsány funkcionality a technologie již v aplikaci implementované před zásahy těchto prací. Následně dojde k rozboru podobných řešení problematiky, ať už v profesionálních aplikacích, či v odborných pracích. Kapitola bude zakončena odvozením uživatelských požadavků, stručně popsanou komunikací se žadateli a nakonec zhodnocením vyhovění požadavků s nařízením GDPR.

Dále práce pokračuje návrhem, kde budou informace získané v analýze použity k návržení případů užití, aktivit, tříd a uživatelského rozhraní daných funkcionalit.

Následující implementace se zabývá integrací navržených funkcionalit do jádra aplikace, spolu s detailními návody k jejich instalaci, použití a rozšíření.

Testování se zabývá provedením a zhodnocením různých druhů testů na již implementovaných funkcionalitách.

Ve finální kapitole této práce budou na základě zkušeností nabraných v rámci předchozích kapitol zmíněny funkcionality, které by do budoucna mohli být implementovány k prospěchu aplikace a jejích uživatelů.

Cíl práce

Cílem této práce je vytvořit modul do Aplikace rugby, který bude umožňovat uživatelské registrace, správu a mazání účtů, zavedení uživatelské role zákonného zástupce pro správu hráčů, spolu s možnostmi zasílání žádostí členům sportovních klubů, jejich následné přijímání a reakci na ně. Součástí tohoto cíle je analýza konkurenčních řešení, odvození a bližší specifikace výše zmíněných systémů ve formě požadavků na aplikaci. Následně bude z daných požadavků proveden návrh a jejich implementace do jádra aplikace, spolu s detailní dokumentací k jejich užití a rozšíření. Implementační část bude zakončena integračními, uživatelskými a akceptačními testy k zajištění funkcionality daných částí modulu. Poslední částí cíle je z nabitých zkušeností z tvorby a zpětné vazby z testů určit možné funkcionality, které by mohli do budoucna být do této aplikace integrovány.

Analýza

V analýze se budeme zabývat rozбором existujícího prototypu, popisy technologií použitých v jeho vývoji, podobných řešení, odborných prací a následně odvozením uživatelských požadavků na modul této aplikace.

2.1 Existující prototyp aplikace

Prototyp Aplikace rugby byl vytvořen v rámci předmětů BI-SP1 a BI-SP2 vedených Ing. Jiřím Chludilem v letním semestru 2019/20 a zimním semestru 2020/21, napsaný v jazyce Kotlin.

Záměr aplikace je digitalizace a zprostředkování organizace rugby klubů, tréninků, turnajů, týmů a správy klubových členů. Primární jazyk této aplikace je čeština, s možnostmi jazykových rozšíření do budoucna. Tato práce se soustředí na vytvoření modulu do této aplikace s funkcionalitami pro zákonné zástupce hráčů a jejich podpůrných systémů.

2.1.1 Architektura

Aplikace byla navržena s třívrstvou architekturou skládající se z prezenční, aplikační a datové vrstvy.

Prezenční vrstva

Realizována ve formě mobilního klienta (front-end) na platformě Android. Je to tzv. „tenký klient“, který silně závisí na připojení a komunikaci s aplikační vrstvou aplikace. Tato komunikace je zprostředkována za pomoci REST API. Zde se využívá technologií platformy Android a prostředí Android Studio.

Aplikační vrstva

Tato vrstva (back-end) se stará o příjem, kontrolu a zpracování HTTP dotazů zaslaných z mobilních klientů, k tomu také extrakci a ukládání souvisejících dat do datové vrstvy. Implementována v prostředí IntelliJ IDEA s použitím Spring framework funkcionalit k tvorbě webových aplikací.

Datová vrstva

Vrstva využívající databázi PostgreSQL k zajištění perzistence a ukládání dat zpracovaných aplikační vrstvou. Propojení datové a aplikační vrstvy bylo realizováno s pomocí Spring framework a Hibernate ORM.

2.1.2 Funkcionalita

Zde následuje seznam s popisy funkcionalit implementovaných v tomto prototypu.

Účty

K využívání aplikace musí mít každý uživatel (kromě dočasných zapisovatelů zápasů) vytvořený účet se jménem a heslem. Tento účet musí mít také přiřazen určitý klub, v jehož rámci se konají veškeré uživateli přístupné operace. Jedinou výjimkou jsou účty administrátorů.

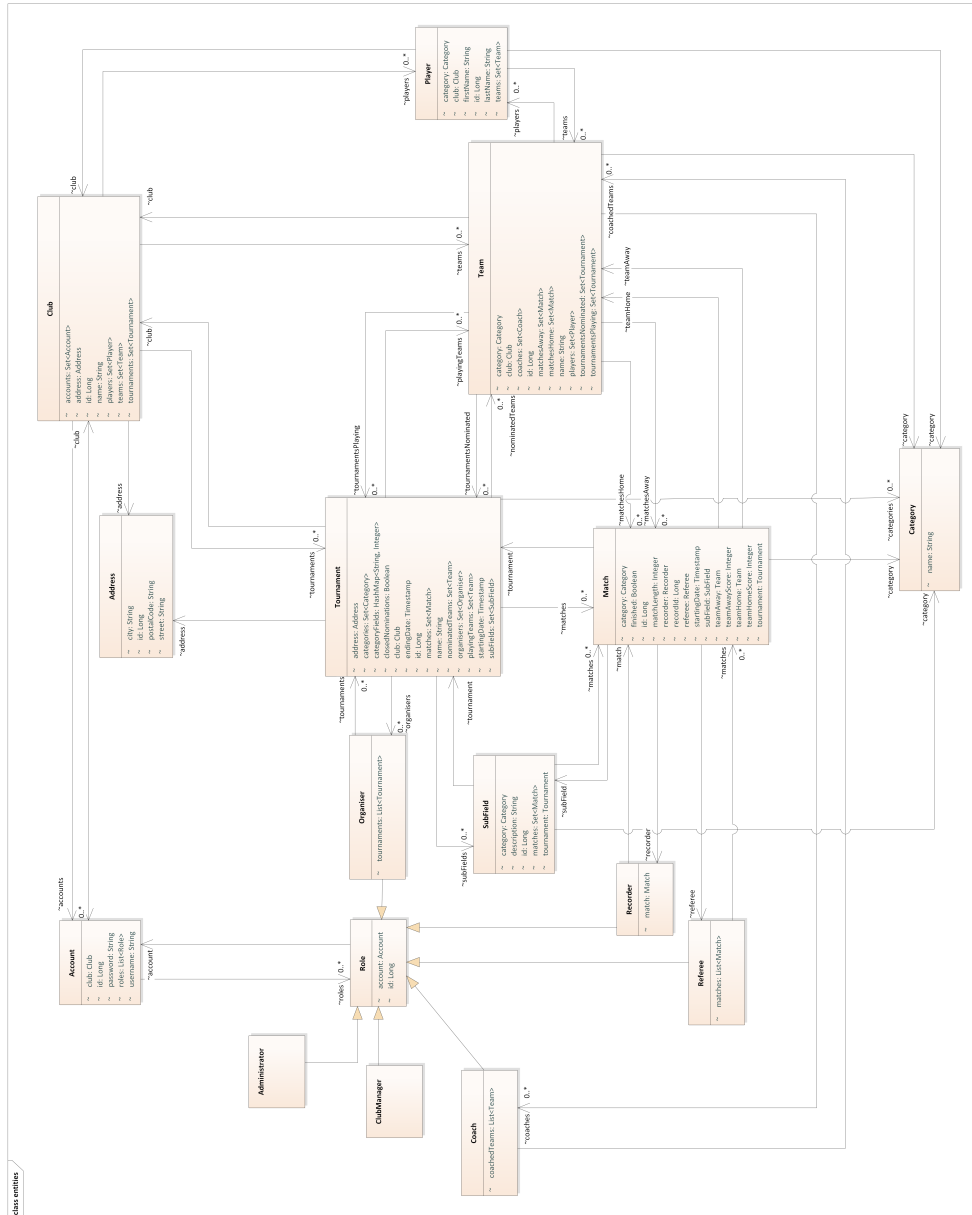
Přihlášení

První, s čím se po spuštění aplikace uživatel setká, je přihlašovací obrazovka. Po vyplnění svých přihlašovacích údajů se zde může uživatel přihlásit do svého účtu. Pro zapisovatele zápasů je zde možnost se dočasně přihlásit bez údajů, pouze za použití kódu dodaného rozhodčím. Při úspěšném přihlášení následuje obrazovka hlavního menu, kde jsou možnosti navigace k funkcionalitám daných přiřazenými rolemi. Možnost registrace prozatím chybí, bude vytvořena v rámci této práce.

Role uživatelů

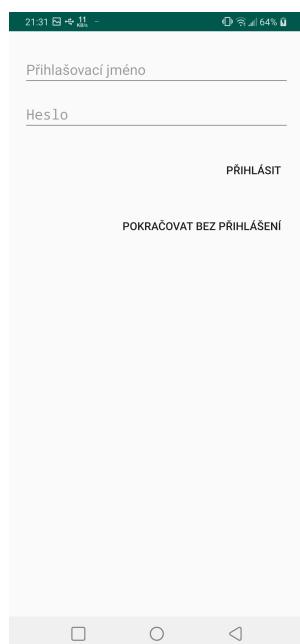
Přístup k některým funkcionalitám je uživateli omezen v závislosti na tom, zda má k účtu přiřazené příslušné role. Daná role může být účtu přidělena jiným účtem, který má pravomoc ji přidělit – ta je většinou součástí vyšší role. Účet s rolí administrátora bude vytvořen manuálně při spuštění služby. Prozatím jsou implementovány role: administrátor, správce klubu, rozhodčí, trenér, pořadatel a zapisovatel. Role hlavního trenéra, zákonného zástupce a hráče budou implementovány až v rámci bakalářských prací.

2.1. Existující prototyp aplikace



Obrázek 2.1: Aplikace rugby – doménový model

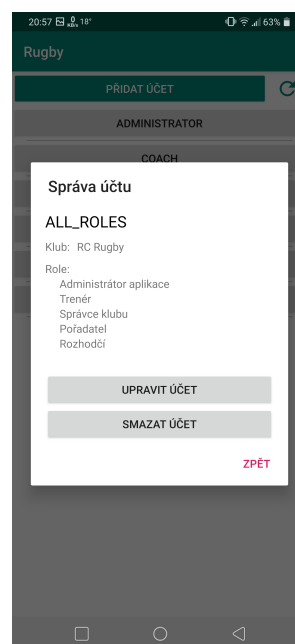
2. ANALÝZA



Obrázek 2.2: Aplikace rugby – přihlášení



Obrázek 2.3: Aplikace rugby – hlavní menu



Obrázek 2.4: Aplikace rugby – detail účtu

Správa účtu

Uživatel si prozatím může měnit pouze své jméno a heslo.

Správa klubu

Sportovní klub může v rámci aplikace vytvořit pouze administrátor. Ten poté může ke klubu přiřadit správce klubu. V této fázi implementace lze měnit pouze název klubu. Správce dále může přiřazovat role trenéra, pořadatele, a dalšího správce ostatním účtům.

Správa týmů

Správci klubu a trenéři mohou vytvořit, měnit specifikace, přidávat hráče a smazat tým. Správce navíc může k danému týmu trenéra přiřadit, čímž mu dá pravomoc nad týmem, který trenér nevytvořil. Oba mohou týmy také nominovat na turnaje, čímž umožní pořadatelům rozhodnout, zda se bude daný tým turnaje účastnit.

Správa hráčů

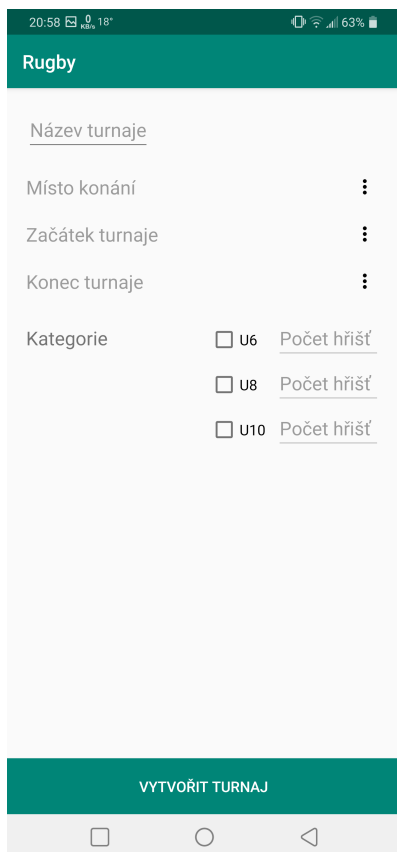
Toto má prozatím na starosti pouze správce klubu. Ten může hráče vytvořit, smazat a upravit jejich detaily.

Správa turnaje

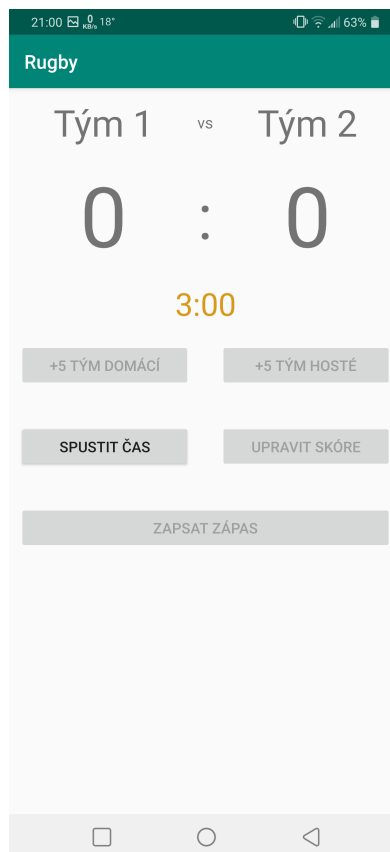
Účet s rolí pořadatele má možnost pořádat a upravovat turnaje. Turnaje mají přiřazený název, kategorii, adresu, časové rozmezí a počet hřišť. Pořadatel má k dispozici seznam týmů nominovaných na zápas a může rozhodnout, které z nich budou součástí turnaje. Je zde možnost nechat automaticky vygenerovat jednoduchý harmonogram zápasů.

Správa zápasu

Pořadatelem je určen ke každému zápasu rozhodčí, který vygenerováním a ukázáním kódu určí zapisovatele skóre pro daný zápas. Ten následně zapisuje skóre a na konci zápasu odešle výsledky. Rozhodčí může tyto výsledky po ukončení zápasu upravovat.



Obrázek 2.5: Aplikace rugby – tvorba turnaje



Obrázek 2.6: Aplikace rugby – zapisování turnaje

2.2 Použité technologie

V této části si přiblížíme technologie, které byly použity k vývoji prototypu a budou nadále používány při tvorbě modulů této aplikace.

Android studio IDE

Toto vývojové prostředí bylo využito k vývoji mobilního klienta (front-end, prezenční vrstva). Jedná se o oficiální prostředí pro vývoj Android aplikací. Je silně založeno na IntelliJ IDEA a využívá jeho editaci kódu a vývojové nástroje. K usnadnění vývoje na systému Android využívá systém pro stavbu aplikací Gradle, emulátor mobilních zařízení, kódové šablony, integraci s Git a vizualizaci XML schémat. Obsažený kódový editor napomáhá vývojářům při psaní kódu a nabízí jeho doplňování, refactoring a analýzu. Aplikace zhotovené v Android studiu jsou zkompileovány do APK formátu a zařazeny k posouzení pro přidání do digitálního obchodu Google Play. Navíc také umožňuje instalaci pluginů, které do prostředí zavádějí nové funkcionality, jako podporu dalších jazyků a technologií. [13]

Pro naše účely bylo vybráno kvůli své využívanosti, dostupnosti dokumentace a vysokou podobností s prostředím IntelliJ IDEA, spolu s jejich sdílenými funkcionalitami.

IntelliJ IDEA IDE

Využito pro vývoj aplikační vrstvy (back-end). Jde o prostředí napsané v jazyce Java a vyvíjené společností JetBrains (dříve IntelliJ). Stejně jako Android studio, i zde jsou obsaženy funkcionality jako kódový editor, systém ke stavbě aplikací (zde Maven), integrace s Git a možnosti instalace pluginů. Navíc je zde obsažen integrovaný systém pro přístup k databázím DataGrip, jenž umožňuje komunikaci s databázovými systémy jako je Microsoft SQL Server, Oracle, PostgreSQL, SQLite a MySQL. [14]

Toto prostředí bylo vybráno díky předchozím zkušenostem, podporou jazyka Kotlin a možnostmi integrace Spring framework.

Spring framework

Aplikační framework a IoC (Inversion of control) kontejner pro vývoj webových aplikací na platformě Java. Zajišťuje funkcionality jako ORM a anotace ke zjednodušení a zvýšení čitelnosti kódu, kontejnery ke konzistenci a správě Java objektů za pomoci reflexe a Spring Boot k zajištění startovních dependencies a skládání aplikace. [15]

Hibernate ORM

System pro objektově relační mapování. Jedná se o implementaci Java Persistence API (JPA) specifikací, čímž umožňuje svou použitelnost v prostředích podporujících JPA, včetně Java SE/EE aplikací. Zabývá se mapováním Java objektů na objekty relační databáze a zajišťováním jejich perzistence. [16]

H2 database engine

Open source SQL databáze s embedded a server módy. Její hlavní API jsou SQL a JDBC, ale umožňuje využívání ovladače PostgreSQL ODBC chováním se jako PostgreSQL server. Všechny její operace s daty jsou transakční. Obsahuje bezpečnostní prvky jako přístupová práva založena na rolích a šifrování hesel a dat s pomocí SHA-256 a AES. Další bezpečnostní prvek je tzv. „disabling literals“, který umožňuje ochranu před SQL injekcemi vynucováním použití parametrizovaných dotazů. [17]

Tato databáze byla použita při vývoji prototypu aplikace, v rámci prací byla nahrazena databází PostgreSQL.

PostgreSQL

Open source SQL databáze soustředící se na rozšiřitelnost a splňování SQL standardů. Zde využita jako datová vrstva aplikace. Byla vytvořena k práci s velkým rozsahem zátěže, od jednotlivých zařízení k webovým službám s velkými počty uživatelů. Je to výchozí databáze MacOS Server, ale je dostupná i pro Windows a Linux. Obsahuje vestavěné interface pro připojování k aplikacím v jazyce C, s dostupnými knihovnami třetích stran pro podporu dalších jazyků. Bezpečnostní prvky této databáze zahrnují šifrování hesel pomocí SCRAM-SHA-256 nebo MD5, přístupová práva s rolemi a podporu pro další externí autentifikační mechanismy, jako GSSAPI, SSPI, certifikáty a další. [18]

Byla využita na doporučení Ing. Jiřího Chludila a kvůli své jednoduchosti použití.

Kotlin

Nový jazyk s vysokou oblíbeností. Jedná se o open-source, staticky typovaný a pragmaticky zaměřený jazyk vytvořený pro JVM a Android. Byl navržen tak, aby kombinoval OOP a FP programovací styly a soustředil se na jednoduchost zápisu, bezpečnost, interoperabilitu a podporu dodatečných nástrojů. Původně vytvořen společností JetBrains. Bezpečnostním prvkem tohoto jazyka je tzv. „null-safety“, vytvořením „nullable“ typů a zakázáním null hodnoty u standartních typů. Návrh Kotlinu je silně založen na Javě, kompiluje stejný byte code a je s ní kompatibilní. [19]

Zde vybrán kvůli jednoduchosti, oblíbenosti a dostupné dokumentaci.

Android Espresso

Testovací framework pro Android. Umožňuje rychlé a jednoduché testování UI. Espresso automaticky synchronizuje akce testů s uživatelským rozhraním aplikace. Též zajišťuje, že se Android aktivity nainstalují před začátkem testů. Další výhodou je jeho jednoduchá instalace v rámci Android Studia. [20]

JSON

JavaScript Object Notation je datový formát pro přenos dat v rámci REST. Byl navržen pro snadné čtení a psaní uživateli a zároveň jednoduché procházení a generování počítači. Je postaven na dvou strukturách: kolekci párů hodnot/jmen a seřazeným listem hodnot. Toto jsou univerzální datové struktury realizovatelné ve všech moderních programovacích jazycích, což zajišťuje jednoduchou převoditelnost na tento formát. [21]

Zde použit pro komunikaci klienta s aplikační a datovou vrstvou.

REST API

Interface pro volání CRUD operací na server pomocí jednoduchých HTTP volání. Application Programming Interfaces (APIs) jsou obecně soubory protokolů a definic pro tvorbu a integrace aplikačního softwaru. REST API specificky umožňuje komunikaci s RESTful webovými službami. Při této komunikaci klient zasílá reprezentaci zdroje v jednom z HTTP formátů (zde JSON). REST API zprostředkovává komunikaci mezi klientem a serverem, k tomu má jedno uniformní interface. [22]

Gitlab

Webový open-source Git repozitář usnadňující verzování a týmovou práci na projektech. Jedná se o kompozitní DevOps platformu zahrnující funkcionality Gitu (tj. ukládání, verzování a merging kódu), Issue tracking, CI/CD integrace a vestavěné služby pro automatizovanou bezpečnost, kontrolu kvality kódu a správu slabín projektu. [23]

Gradle a Apache Maven

Nástroje pro automatizované sestavování programů. Oba nástroje řeší, jak jsou projekty sestaveny a spravují jejich vyžadované dependency. Spravují také procesy kompilace, testování, packaging a deployment. Gradle je silně založen na konceptech z Maven a zavedl DSL ke konfiguraci projektů založený na jazycích Groovy a Kotlin, na rozdíl od Maven, který je založen na XML konfiguraci. [24], [25]

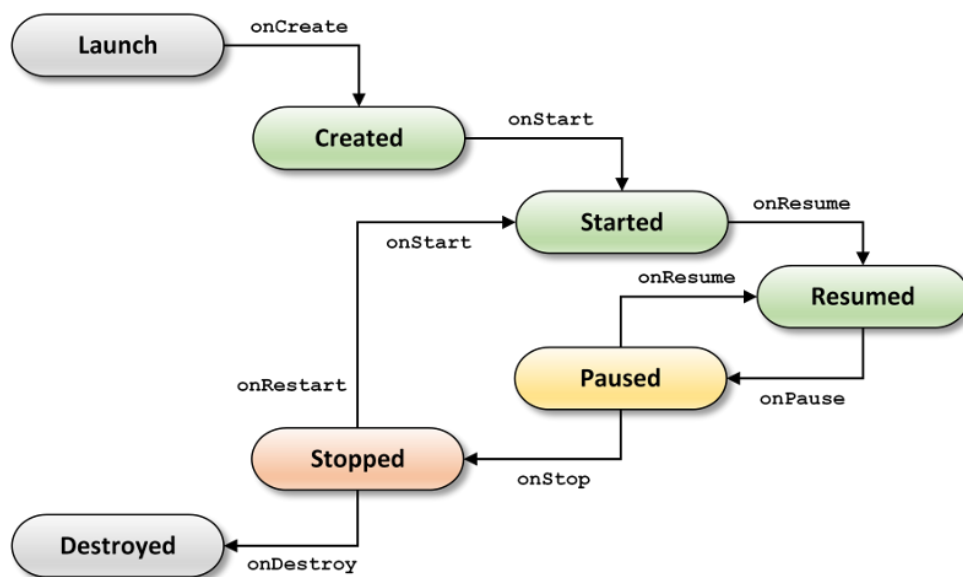
Maven je integrován v IntelliJ IDEA, Gradle v Android Studiu.

Android Activity

Nástroj pro tvorbu uživatelského rozhraní na platformě Android. Aktivita reprezentuje obrazovku s uživatelským rozhraním. Každá aktivita má vlastní životní cyklus, jehož stavy se mění v závislosti na zavolaných callback funkcích v ní zabudovaných. [26]

Android Fragment

Další z nástrojů pro tvorbu uživatelského rozhraní na platformě Android. Jedná se o komponentu použitelnou v rámci aktivity, která má vlastní layout a životní cyklus ovlivnitelný rodičovskou aktivitou fragmentu. Na rozdíl od aktivit, na jedné obrazovce může být více fragmentů. Fragmenty umožňují opětovné použití pohledů a logických komponent. [27]



Obrázek 2.7: Android Activity – životní cyklus [1]

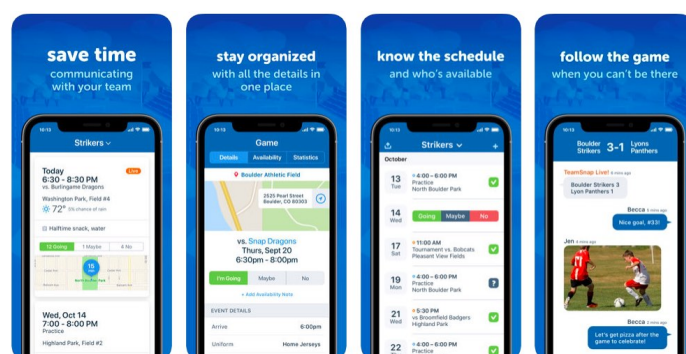
2.3 Konkurence

K obhájení tvorby a případné inspirace při vývoji aplikace byla analyzována již existující řešení podobných problematik. Zde jsou detailní popisy tematicky či implementačně nejpodobnějších z nich.

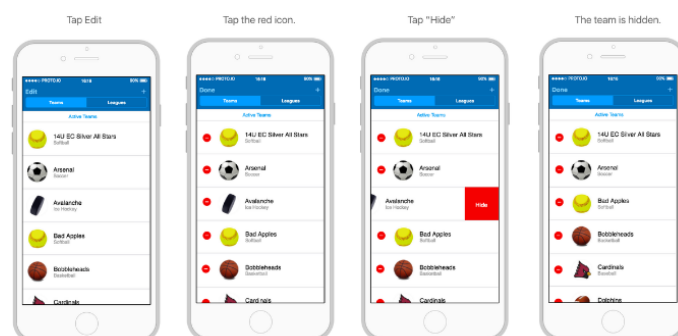
2.3.1 TeamSnap

Mobilní aplikace pro tvorbu a správu sportovních klubů, lig, týmů a turnajů. Vysoce robustní, úspěšná a oblíbená. Podporuje přes 100 různých sportů včetně amerického fotbalu a rugby. Registrace klubů a lig přes web s možností tvorby vlastní webové stránky. Umožňuje komunikace s kýmkoliv, nebo pouze s určitými skupinami. Je schopna zprostředkování správy klubů, týmů, hráčů a trenérů, automatické generace harmonogramů a jednoduchých background checks trenérů a dobrovolníků. Aplikace má 14denní demo zdarma, poté je nutno za její užití platit měsíční poplatky.

Pro účely žadatelů nevyhovující, kvůli nedostatku nástrojů pro správu zapisovatelů a rozhodčích, funkcionalit uzavřených za měsíční platby a chybějící podpoře češtiny. [28]



Obrázek 2.8: TeamSnap – ukázka funkcionalit [2]



Obrázek 2.9: TeamSnap – seznam týmů [3]

2.3.2 Spoluhráči.cz

Webová aplikace pro hledání spoluhráčů, vytváření virtuálních kluboven a inzerce sportovců. Virtuální klubovny mají omezené prostředky na pořádání zápasů a tréninků. Aplikace neumožňuje zásahy rodičů do účtů dětí a nemá mobilní aplikaci. [4]

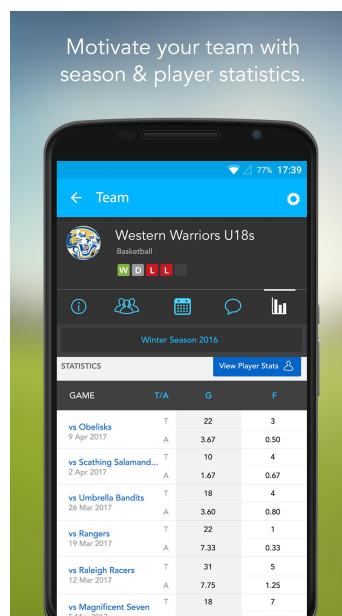
2.3.3 Teamstuff

Webová i mobilní aplikace na platformách Android i IOS. Umožňuje tvorbu klubů, týmů, akcí a má podporu pro rodiče. Je integrována s mapami k nalezení lokací zápasů.

Tato aplikace silně odpovídá našim požadavkům, avšak stále postrádá funkcionality pro správu trenérů, týmů a rozhodčích. K tomu je dost funkcionalit uzamčeno s nutností zaplatit členství a aplikace samotná získává nízká ohodnocení kvůli špatné stabilitě aplikace a tiché zákaznické podpoře. Během tvorby práce došlo kvůli pandemii COVID k uzavření serverů a aplikace již není dostupná. [5], [29]



Obrázek 2.10: Spoluhráči.cz – hlavní stránka [4]

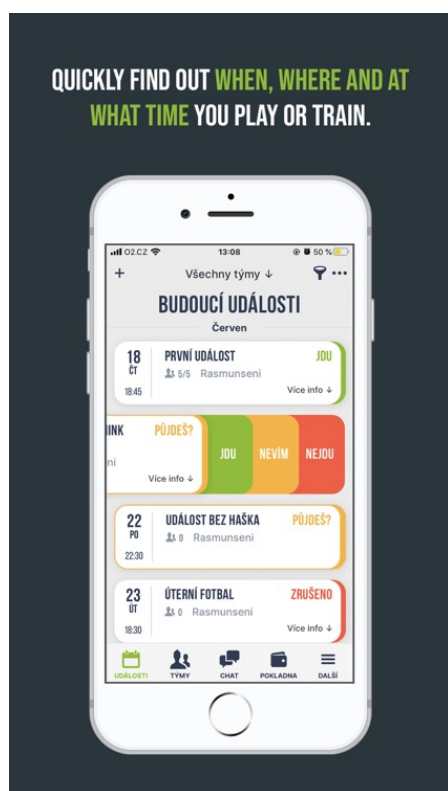


Obrázek 2.11: Teamstuff – statistiky týmu [5]

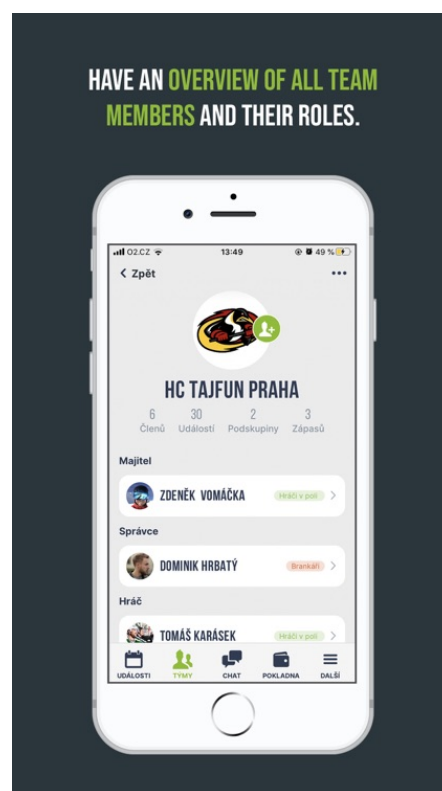
2.3.4 Týmuj

Aplikace na organizaci sportovních týmů primárně pro platformy IOS a Android, s dostupností i prohlížeči. Nemá zaměření na jeden určitý sport a podporuje jich více než 100. Poskytuje možnosti rychlé tvorby tréninků, zápasů a týmů, informace o docházce hráčů a chatovou komunikaci. U docházky lze omezovat počet účastníků v rámci událostí a získat statistiky o účasti. Aplikace je schopna zasílat e-mailové a push notifikace. Ve sféře komunikací také umožňuje tvořit a vyplňovat ankety a zobrazovat nástěnky se zprávami od vedoucích týmů. Jazyková podpora zahrnuje mj. češtinu, angličtinu a slovenštinu. K vedlejším funkcím také patří galerie fotografií a pokladna s informacemi k platbám.

Má omezenou podporu pro tvorbu klubů a neposkytuje žádné nástroje pro trenéry a rodiče. Kvůli těmto nedostatkům pro účely žadatelů nevhodná. [30]



Obrázek 2.12: Týmuj – seznam událostí [6]



Obrázek 2.13: Týmuj – seznam členů a jejich rolí [6]

2.3.5 TeamApp

Mobilní aplikace pro tvorbu a správu klubů a sportovních týmů. Aplikace sama je velice modulární a přizpůsobitelná, umožňuje individuálním uživatelům vybrat, které funkcionality budou v aplikaci přítomny. Obsahuje chatovou komunikaci, detailní správu týmů, kalendář událostí a nouzové kontakty pro členy klubů. Je schopna zasílat push a e-mailové notifikace. Každý uživatel s instalovanou mobilní aplikací má také k dispozici dynamicky se obnovující webovou stránku, která umožňuje přístup k týmům a klubům uživatelům přímo z webového prohlížeče.

Má ovšem nedostatečné možnosti správy trenérů, rozhodčích a zápasů v rámci klubů. Nastavení modularity UI této aplikace také může být pro průměrného uživatele moc komplexní a matoucí, zvláště pro mladé děti. [7]



Obrázek 2.14: TeamApp – startovní obrazovka a hlavní menu [7]

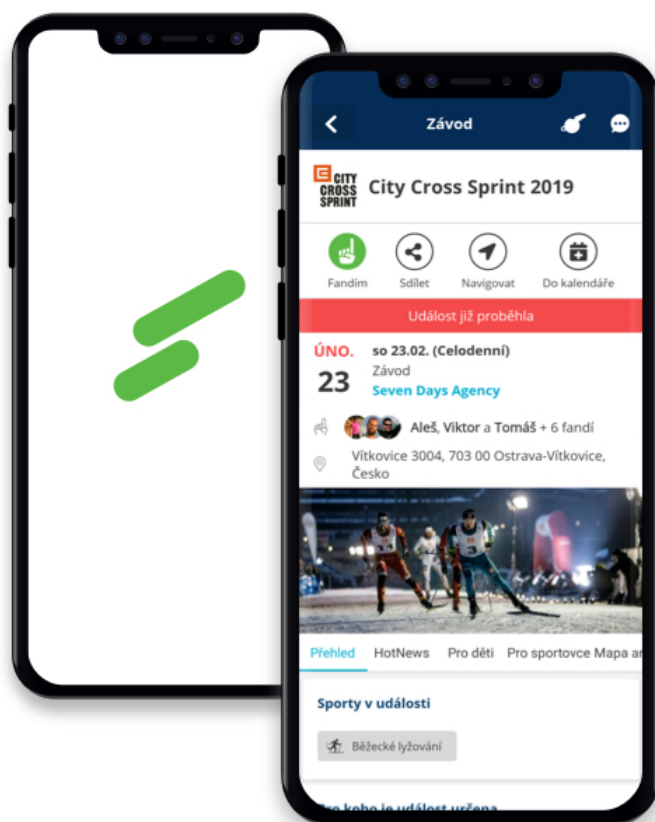
2.3.6 Sportnect

Komplexnější projekt označující se jako „sportovní sociální síť“, zaměřený na propojování sportovců, sportovních fanoušků, trenérů, organizátorů a rodičů sportujících dětí. Nemá specifické zaměření na jeden sport: „*Registrováno je 216 sportů, přes 1000 týmů, klubů a skupin.*“ [8]. Umožňuje vyhledávání sportovních akcí v okolí uživatelů a možnosti na jejich přihlášení, dále zobrazuje

2. ANALÝZA

přehledy sportovních klubů a týmů s informací, zda momentálně nabírají nové členy. Uživatelé dále mají možnosti sdílení svých atletických úspěchů, osobních rekordů, výsledků zápasů, fotografií a videí. Rodiče zde mohou vytvářet profily svým dětem, či se propojit s již existujícím účtem. K tomu zjednodušuje komunikaci mezi rodičem, dítětem a trenérem, eviduje informace o docházce dětí na tréninky a zjednodušuje potřebné platby. Co se týká jejich řešení klubů a týmů, aplikace vede evidenci členů, docházky trenérů a členských příspěvků, tréninkové plány a nominace na zápasy.

Toto řešení je velice podobné naší aplikaci. Ovšem, i když má funkcionality, které u nás nejsou zatím plánovány, stále postrádá funkce jako komplexnější správu dětí rodiči, nouzový kontakt, jednoduché vytváření týmů v rámci klubu, uspořádání hráčů do kategorií a role zapisovatelů a rozhodčích. [8]



Obrázek 2.15: Sportnect – stránka s událostí [8]

2.3.7 Výsledky konkurenční analýzy

Z pozorování a porovnávání konkurenčních řešení byly vybrány následující funkcionality, které hodláme zavést do naší aplikace a na trhu jsou v rámci jedné aplikace postrádány:

- Jednoduché, srozumitelné UI
- Sledování docházky
- Možnosti hromadných registrací
- Zavedení všech potřebných rolí (včetně rodičů, rozhodčích a zapisovatelů)
- Tvorba a přizpůsobení klubů a správa jejich členů.
- Hlubší možnosti přizpůsobení turnajů (dělení na podhřiště, pořadí zápasů, ...)
- Nouzový kontakt pro rodiče
- Detailnější sledování hráčů
- Možnosti rodičovské kontroly
- Podpora českého jazyka

2.4 Relevantní práce

K inspiraci a seznámení se s momentální „state of the art“ byly analyzovány následující bakalářské a diplomové práce. V jejich výběru hrálo roli zaměření v oblastech: rodičovská kontrola, registrace/mazání účtů, zaslání zpráv, uživatelské rozhraní a obecná tvorba Android aplikací.

2.4.1 FurryBall – správa licencí a uživatelů

Z abstraktu práce:

„Tato bakalářská práce se zabývá návrhem a realizací prototypu webové aplikace pro prodej a správu licencí renderovacího softwaru FurryBall a podporu jeho uživatelů.“ [31]

Ačkoliv se nejedná o mobilní aplikaci, lze zde nalézt podobné řešení backend implementace skrze Symfony framework a Doctrine knihovny pro zajištění datové perzistence. Pro správu účtů je zde využít FOSUserBundle, který zprostředkovává potřebné funkcionality jako registraci, přihlašování atd. Tento balíček je u nás nevyužitelný, jelikož ho nepodporuje Spring.

Relevantními výňatky zde jsou registrace a správa účtů, jejich návrh a implementace.

2.4.2 BigCloud – modul obchodníka

Tato práce se zabývá návrhem a tvorbou modulu pro registraci a správu uživatelů (obchodníků firmy) a zavedení jimi vyžadovaných funkcionalit, jako generace přehledů plateb a faktur, do hybridního cloudového systému BigCloud. Součástí práce je detailní popis a vyobrazení návrhu uživatelského rozhraní, spolu s možnostmi tvorby a správy účtů.

Pro nás podstatnou částí práce pro inspiraci jsou registrace, možnosti zobrazení účtů, jejich hierarchie a omezené možnosti nastavení. [32]

2.4.3 Výuková aplikace Dráček II – gamifikace a personalizace

Práce se soustředí na popis a implementaci responzivního uživatelského rozhraní do výukové aplikace Dráček II a její úpravu ke zjednodušení přístupu dětem s poruchami učení. Aplikace je též vyvíjena na platformě Android. Analýza práce se detailně zabývá teorií a historií grafiky, výukových her a vývojem populační gramotnosti.

Cílová skupina této aplikace (děti prvního stupně ZŠ, děti s poruchami učení) spadá do skupiny potenciálních uživatelů Aplikace rugby, proto budou při vývoji zohledněny její postupy tvorby. Důležitými koncepty, které budou z této práce převzaty, jsou filozofie návrhu uživatelského rozhraní (jen méně barevné), chybové hlášky pro případy uživatelské chyby a metodiky testování implementovaného návrhu. [33]

2.5 Aktéři

Zde si vyslovíme a popíšeme, kdo přesně jsou potenciální uživatelé naší aplikace související s tímto modulem.

Zákonný zástupce

Zákonný zástupce (ZZ, popř. rodič) je osoba, která zodpovídá za jiné osoby, které buďto již jsou, nebo se chtějí stát hráči v rámci jednoho z klubů. Daná osoba potřebuje mít v aplikaci možnosti správy účtů svých svěřenců. Tato práce je zaměřena na splnění potřeb této role. Specificky je tato role určena rodičům, poručníkům nebo jiným osobám s legálním statutem zákonného zástupce hráčů.

Hráč

Hráč je osoba, která se chce účastnit tréninků a zápasů pořádaných daným rugby klubem. Tento uživatel může být nezletilý, tudíž je zde potřeba zákonného zástupce, který za jeho aktivity bude zodpovídat. V rámci tohoto modulu se budeme soustředit pouze na jeho vazby se zákonným zástupcem.

Hlavní trenér

Hlavní trenér je člen klubu, který je zodpovědný za správu týmů, hráčů a trenérů v rámci jedné určité kategorie. Tento trenér zprostředkovává registrace ZZ a nepřímo i hráčů.

Trenér

Trenér je člen klubu zodpovědný za správu týmů, jejich hráčů a slouží jako jejich dozor během tréninků a turnajů. V tomto modulu zodpovídají za přijímání či odmítání hráčů na tréninky a turnaje.

2.6 Funkční požadavky

Na základě předchozí analýzy prototypu aplikace, konkurenčních řešení, podobných prací a předběžnou komunikací se žadatelem během vývoje daného prototypu byly sepsány následující požadavky na funkcionality, které by finální podoba aplikace měla obsahovat.

2.6.1 F1 – Registrace

Uživatel bude mít možnost si registrovat účet již po pár stisknutí tlačítek. K registraci je nutno aby uživatel vyplnil svůj e-mail, dvakrát své zvolené heslo a sportovní klub, ve kterém chce vykonávat své aktivity. Po zdařilé registraci se bude uživatel moci přihlásit ke svému účtu v aplikaci vyplněním svého e-mailu a hesla do kolonek na přihlašovací obrazovce, po čemž bude přesunut do hlavního menu. Účet takto vytvořený nebude mít přiřazeny žádné role.

2.6.2 F2 – Formuláře

Budou vytvořeny možnosti zaslání a přijímání formulářů k automatickému vykonání správních úkonů uvnitř aplikace. Tvorba i přijímání těchto formulářů bude přímo dostupná z hlavního menu.

Formulářem je myšlena specifikovaná žádost o vykonání určitého správného úkonu s příloženými informacemi potřebnými k jeho vykonání a možnostmi zaslání zodpovědné osobě. Jediným druhem formuláře vytvořeným v rámci toho požadavku bude Přiřazení/Odebrání role.

2.6.3 F3 – Role hlavního trenéra

V rámci účtů bude vytvořena role hlavního trenéra, která bude mít přiřaditelnou kategorii. V rámci tohoto modulu bude implementována pouze částečně, z hlediska registrace zákonných zástupců a hráčů. Tato role bude přiřaditelná správci klubu, formulářem Přiřazení/Odebrání role, viz F2 – Formuláře.

2.6.4 F4 – Role zákonných zástupců

V rámci účtů bude vytvořena role zákonného zástupce (ZZ), která bude mít k dispozici možnosti vytváření účtů pro hráče a jejich správy. Zástupce bude mít možnosti manipulovat s účty svých hráčů, viz F6 – Možnosti správy účtu hráče. Součástí této role budou možnosti zaslání formulářů trenérům klubu s hráči souvisejícími. Specificky budou implementovány následující formuláře:

- Přeražení hráče jinému zástupci
- Přihlášení/Odhlášení hráče z týmu/tréninku

K tomu bude tato role přiřaditelná hlavními trenéry, formulářem Přiřazení/Odebrání role, viz F2 – Formuláře.

2.6.5 F5 – Možnost registrace hráče

Účet s rolí ZZ bude mít k dispozici nástroje k registraci účtů s rolí hráče v rámci aplikace. Tato registrace bude zástupci přístupná z hlavního menu a proběhne vyplněním e-mailu, hesla a data narození hráče.

2.6.6 F6 – Možnosti správy účtu hráče

Účet s rolí ZZ bude mít k dispozici nástroje pro správu svěřených účtů s rolí hráče v rámci aplikace. Správou se rozumí nastavení rodičovské kontroly, úprava detailů účtu a možnosti přijímání a reakce na formuláře hráči zaslány. V rámci tohoto požadavku budou pro hráče vytvořeny zástupcům zaslatelné formuláře:

- Změna detailu účtu
- Žádost o Přihlášení/Odhlášení hráče z týmu/tréninku

2.6.7 F7 – Možnosti smazání účtu

Účet s rolí ZZ bude mít několik možností zrušení (dočasné nebo trvalé) svého účtu za účelem zachování či smazání svěřených účtů s rolí hráče. ZZ bude moci rozhodnout, zda bude hráč osamostatně (18+), přeražen jinému ZZ, nebo také zrušen. Každý uživatel s dočasně zrušeným účtem bude mít možnost jej obnovit za pomoci vygenerovaného odkazu zasláního v e-mailové komunikaci při výmazu.

2.6.8 F8 – Potvrzení registrace e-mailem

Na zadaný e-mail bude zaslán hypertextový odkaz, po jehož stisknutí bude daná osoba převedena na webovou stránku, kde bude informována že asociovaný účet byl úspěšně vytvořen. Tímto bude její účet aktivován a bude možno se na něj přihlásit.

2.6.9 F9 – Externí registrace zákonného zástupce

Bude vytvořen způsob registrace ZZ upřednostňován hlavním trenérem za pomoci e-mailové komunikace. Hlavní trenér vypíše přes e-mail formulář k registraci, budoucí ZZ jej vyplní a zašle. Poté bude automaticky vytvořen účet s prvotním heslem, které ZZ získá e-mailem a využije k přihlášení do tohoto účtu.

2.6.10 F10 – Nouzový kontakt

Účet s rolí trenéra bude mít k dispozici možnost kontaktovat osobu s rolí ZZ skrze telefonní komunikaci v případě nehody (zranění, ztráty, ...) spojené se svěřeným hráčem v rámci tréninků a zápasů. Telefonní komunikací je myšleno, že trenér bude moci rozhodnout, zda zavolat na číslo ZZ, čímž bude přeměrován na funkci volání svého zařízení, nebo zaslat vygenerovanou a upravitelnou SMS, též v rámci funkcionalit zařízení.

2.7 Nefunkční požadavky

Na základě předešlé analýzy byly také vytvořeny následující nefunkční požadavky, které přibližují specifika implementace.

N1 – Specifika platformy

Implementace modulu bude vytvořena na platformu Android 9.0 a výše, napsána bude v jazyce Kotlin.

N2 – Specifika jazyka

Jazyk uživatelského rozhraní modulu bude čeština.

N3 – Specifika formulářů

Formuláře v jejich inboxu bude možno filtrovat dle data zaslání, typu a abecedně podle jména odesílatele.

2.8 GDPR, anonymizace, pseudonymizace a mazání účtu

V požadavku F7 – Možnosti smazání účtu byl zjištěn potenciální problém spojený s dodržáním General Data Protection Regulation (GDPR) nařízení EU zavedeného 27. dubna 2016. Toto nařízení omezuje možnosti manipulace s osobními údaji.

V původním požadavku byly možnosti znovunastolení zrušeného účtu, což by bylo možné zřídit tzv. pseudonymizací, která by zašifrovala údaje spojené s účtem a vytvořila klíč, který by mohl být použit pro obnovení daného účtu. Pseudonymizovaná data jsou narozdíl od dat anonymizovaných stále brány jako údaje o identifikovatelné osobě podle bodu (26):

„Osobní údaje, na něž byla uplatněna pseudonymizace a jež by mohly být přiřazeny fyzické osobě na základě dodatečných informací, by měly být považovány za informace o identifikovatelné fyzické osobě.“[34]

Pro data anonymizovaná dle stejného bodu to samé již neplatí:

„Zásady ochrany osobních údajů by se proto neměly vztahovat na anonymní informace, totiž informace, které se netýkají identifikované či identifikovatelné fyzické osoby, ani na osobní údaje anonymizované tak, že subjekt údajů není nebo již přestal být identifikovatelným. Toto nařízení se tedy netýká zpracování těchto anonymních informací, včetně zpracování pro statistické nebo výzkumné účely.“[34]

V rámci tohoto nařízení je pseudonymizace popisována jako validní forma šifrování dat [dle bodu (156), části Zákonost zpracování, bod 4e), článku 25 bod 1 a článku 32 bod 1a), definována v části Definice, bod 5)]. Jelikož se ale na pseudonymizovaná data stále vztahuje GDPR, je jejich pseudonymizace v rámci dočasného zrušení irelevantní, bude proto navržen způsob tohoto zrušení skrze prostou změnu stavu účtu, kterou aplikace při přihlášení zkontroluje a znemožní zrušeným účtům pokračovat.

2.9 Komunikace se žadateli

Tato aplikace byla navržena na základě komunikací mezi členy klubu RC Říčany, Ing. Jiřím Chludilem a autory těchto prací.

Tvorba dotazníku Na základě informací zadaných Ing. Jiřím Chludilem byl vytvořen dotazník (viz Přílohy – Dotazníky) sloužící k získání zpětné vazby od členů klubu. Informace získané tímto dotazníkem pomohou změnit detaily návrhu aplikace na přání zákazníků.

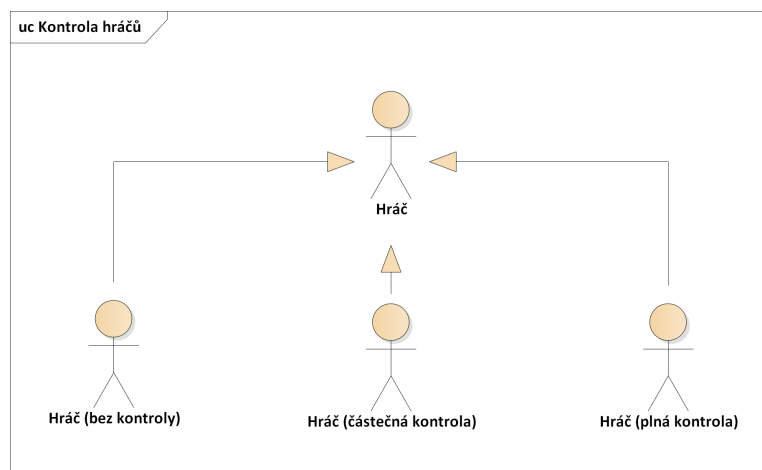
Problémy Během tvorby prací došlo ke zpožděním a omezením kontaktu se členy klubu RC Říčany, způsobených onemocněními a vytížeností jejich rozvrhu spojených s pandemií COVID-19. Z těchto důvodů musely být velké části návrhu a implementace této práce tvořeny nezávisle, buďto bez, nebo jen s minimální a opožděnou zpětnou vazbou.

Návrh

V návrhu se budeme soustředit na zpracování informací získaných v analýze a jejich využití ke koncepci a popisu odvozených funkcionalit.

3.1 Případy užití

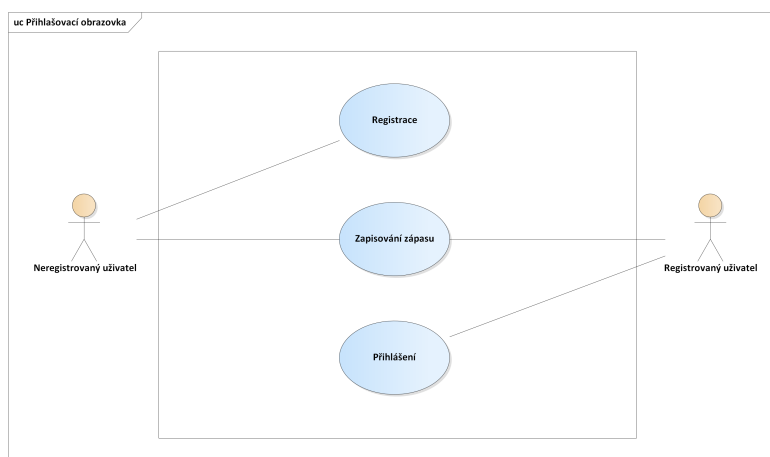
Zde se zaměříme na obecné případy užití a jejich vztahy s různými uživatelskými rolemi. Tyto případy užití jsou seskupeny dle jejich plánované dostupnosti v rámci jednotlivých obrazovek aplikace.



Obrázek 3.1: Use case – Kontrola hráčů

3.1.1 Přihlašovací obrazovka

Po spuštění aplikace bude uživatel přesměrován na startovní obrazovku, kde bude mít možnost se buďto přihlásit (pokud již má účet), registrovat (pokud účet nemá), nebo zapisovat zápas. K zapisování zápasu není potřeba účet, ale je nutno mít rozhodčím zadán vygenerovaný kód k zapisování daného zápasu. Toto zapisování je již v prototypu implementováno.



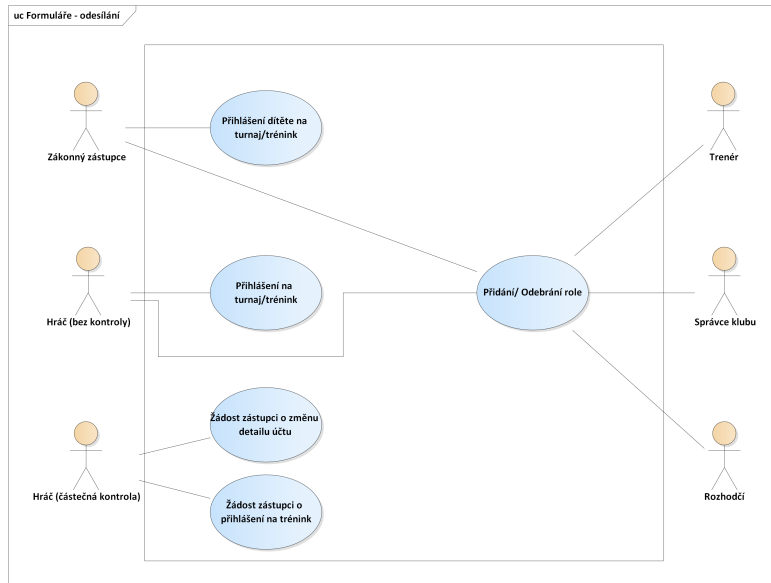
Obrázek 3.2: Use case – přihlašovací obrazovka

3.1.2 Formuláře

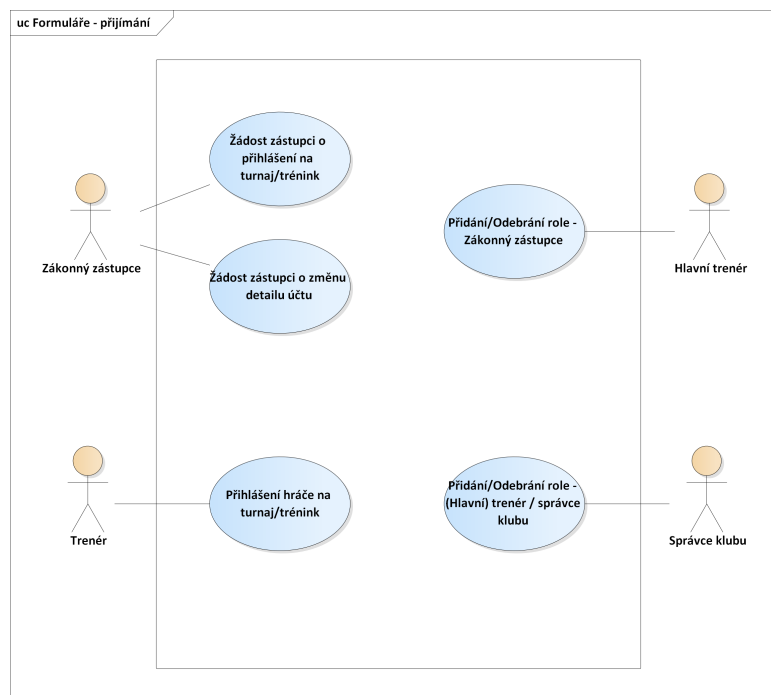
Z menu bude dostupná možnost zaslání formulářů, ale jen některé budou uživatelům dostupné bez příslušné role. Formulář „Přidání/Odebrání role“ bude k dispozici všem, zatímco „Přihlášení dítěte na turnaj/trénink“ a „Přeřazení hráče“ budou dostupné pouze zákonnému zástupci. Hráči bez rodičovské kontroly budou moci se na trénink/turnaj tímto způsobem přihlásit sami, ovšem hráči s částečnou kontrolou musí skrze formuláře zasílat žádosti svému zástupci, včetně změn detailů účtu. Hráč s úplnou kontrolou nemá k zasílání formulářů přístup vůbec.

Příjem formulářů do inboxu bude také rolemi omezen. Správci klubu budou přijímat formuláře s žádostmi k přiřazení či odebrání trenérů a dalších správců, podobně hlavní trenéři budou přijímat od zákonných zástupců. Žádosti od hráčů budou přicházet zákonným zástupcům a přihlášení na tréninky trenérům.

3.1. Případy užití



Obrázek 3.3: Use case – odesílání formulářů

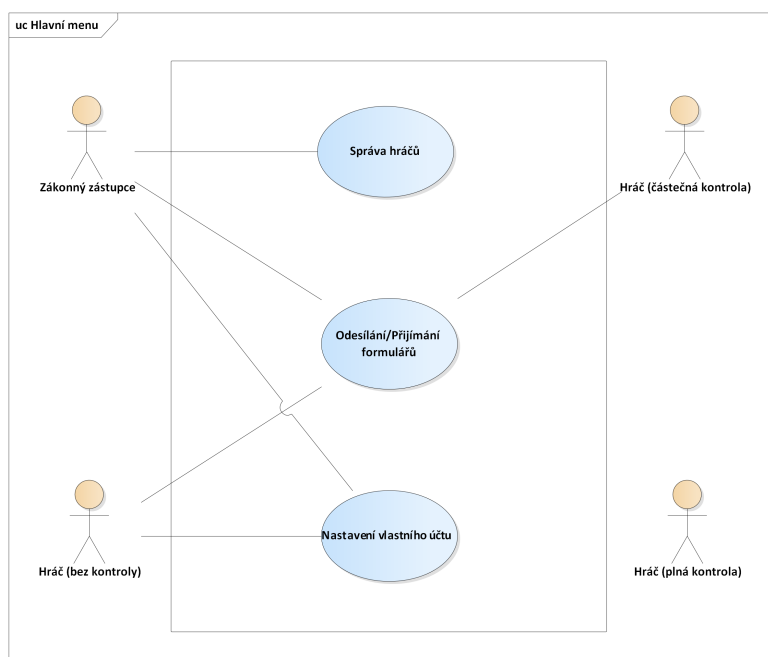


Obrázek 3.4: Use case – příjem formulářů

3. NÁVRH

3.1.3 Hlavní menu

V hlavním menu bude zástupcům dostupná správa jeho hráčů a k tomu nastavení jeho účtu. Nastavovat účet může také hráč bez rodičovské kontroly. Zákonný zástupce a hráči, kteří nemají plnou rodičovskou kontrolu mohou také zasílat a přijímat formuláře.



Obrázek 3.5: Use case – hlavní menu

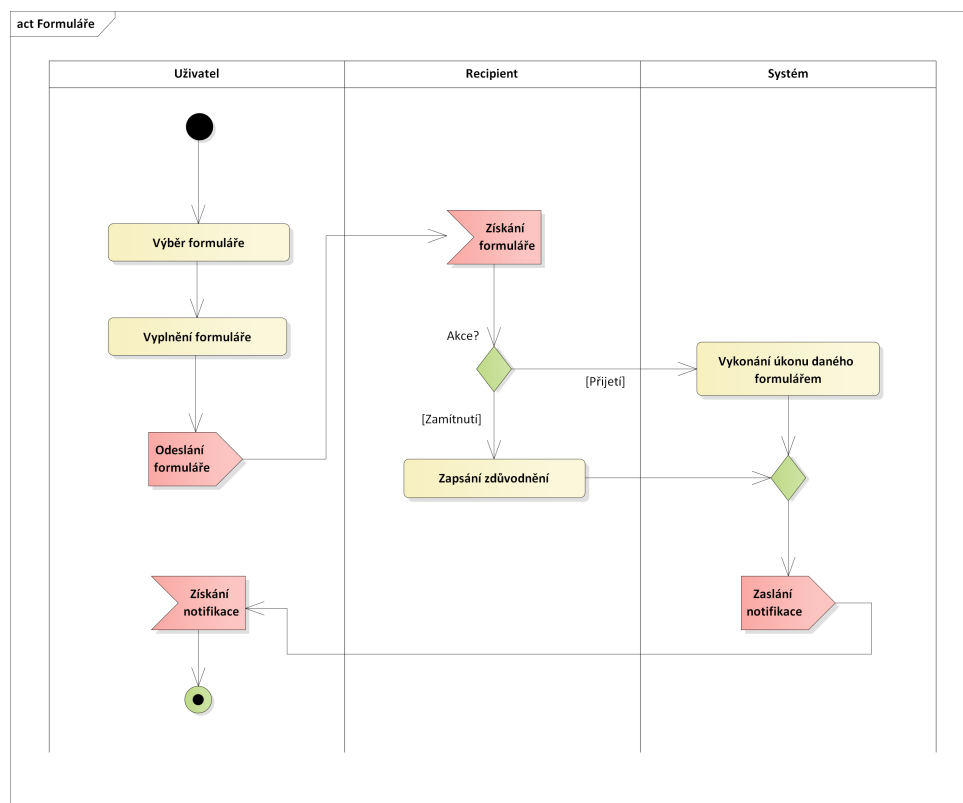
3.2 Aktivity uživatelů

V této sekci budou přiblíženy očekávané průchody řešených funkcionalit.

3.2.1 Zasílání formulářů

V původní verzi aplikace musela k vykonání správních úkonů zodpovědná osoba manuálně procházet různá menu a vyhledávat uživatele. K minimalizaci nutnosti užití tohoto procesu bude zaveden systém pro vyplňování, zasílání a příjem formulářů.

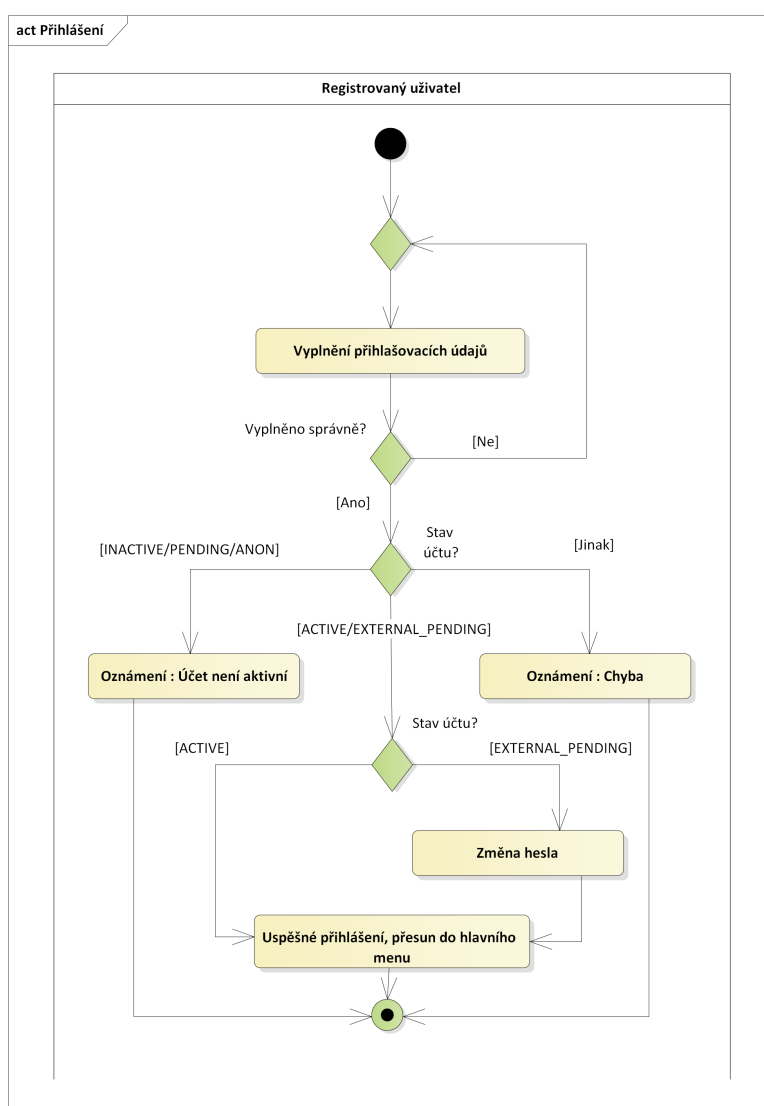
Uživatel, který chce změnit stav účtu svého, či k sobě svěřeného může v aplikaci zasílat žádosti zodpovědné osobě ve formě těchto formulářů. Recipient formuláře má možnost žádost odmítnout (s nutností udat důvod ve formě zpětné zprávy), či přijmout, což vykoná úkon daný typem vybraného formuláře. Ať už byl tento formulář přijat, či zamítnut, jeho odesílatel bude informován prostřednictvím zprávy v inboxu.



Obrázek 3.6: Diagram aktivit – formuláře

3.2.2 Přihlášení

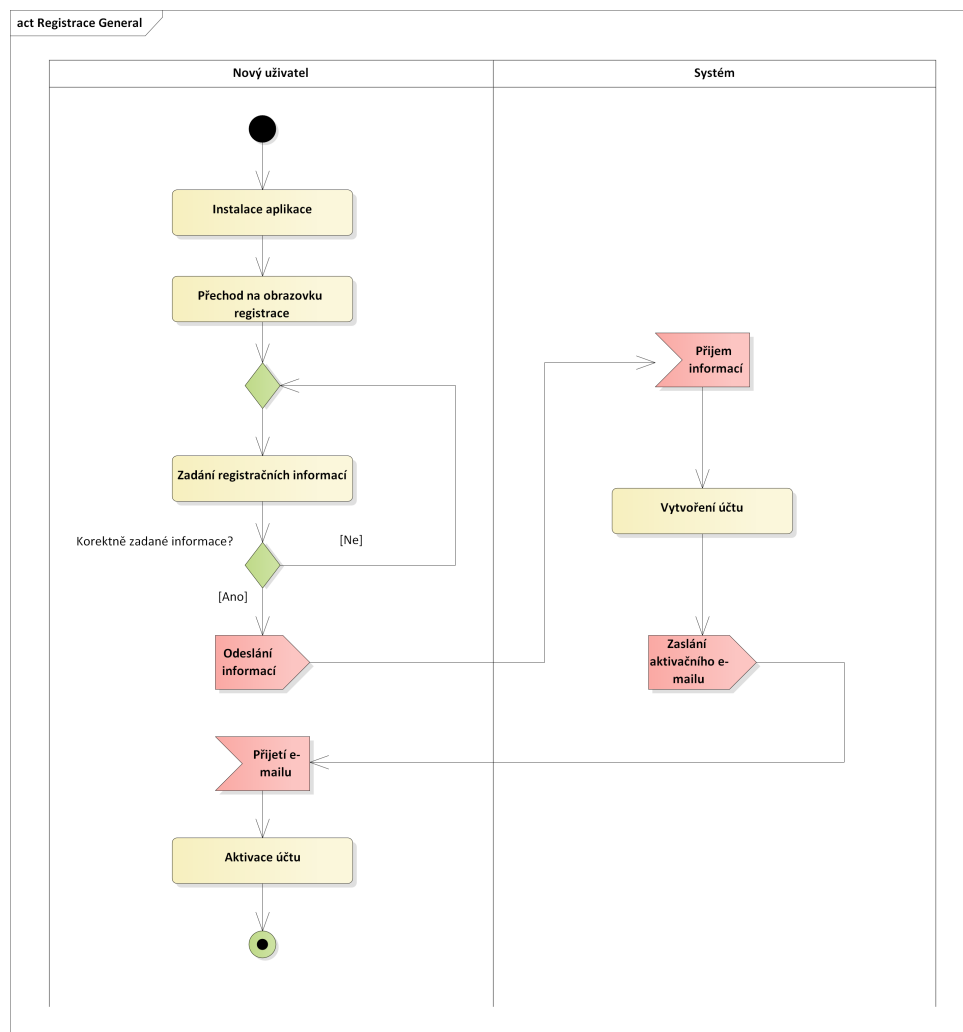
Přihlášení registrovaného uživatele proběhne vyplněním e-mailu/přezdívky a hesla. Poté bude následovat kontrola stavu přihlašujícího se účtu, kde aplikace zapře přístup neaktivním účtům a oznámí problém. Je-li účet ve stavu EXTERNAL_PENDING, vyskočí obrazovka, která požádá uživatele o změnu prvotního hesla. Dále bude tento uživatel stejně, jako ti se stavem ACTIVE přesunut do hlavního menu aplikace.



Obrázek 3.7: Diagram aktivit – přihlášení

3.2.3 Registrace

Na základě prvotních požadavků byly vytvořeny dvě různé možnosti registrace účtu s rolí zákonného zástupce. Interní registrací je myšlena typická registrace uživatelem v aplikaci (viz požadavek F1 – Registrace) a následné zaslání žádosti (F2 – Formuláře) o roli hlavnímu trenérovi. Externí registrace slouží k emailem řízené hromadné registraci uživatelů hlavním trenérem.

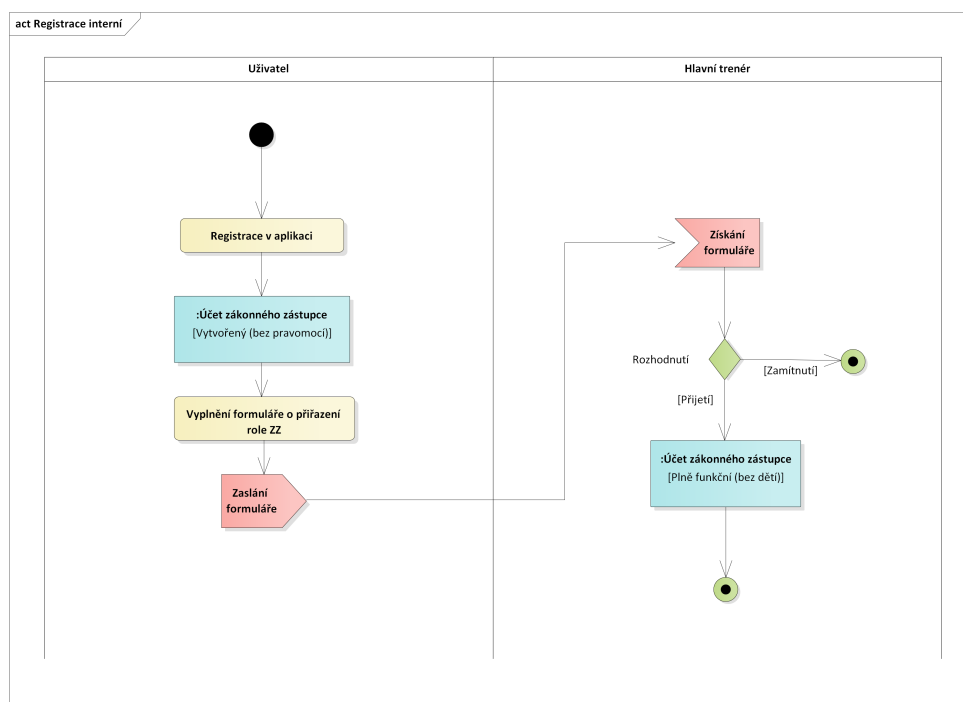


Obrázek 3.8: Diagram aktivit – registrace

3. NÁVRH

Interní registrace zástupce

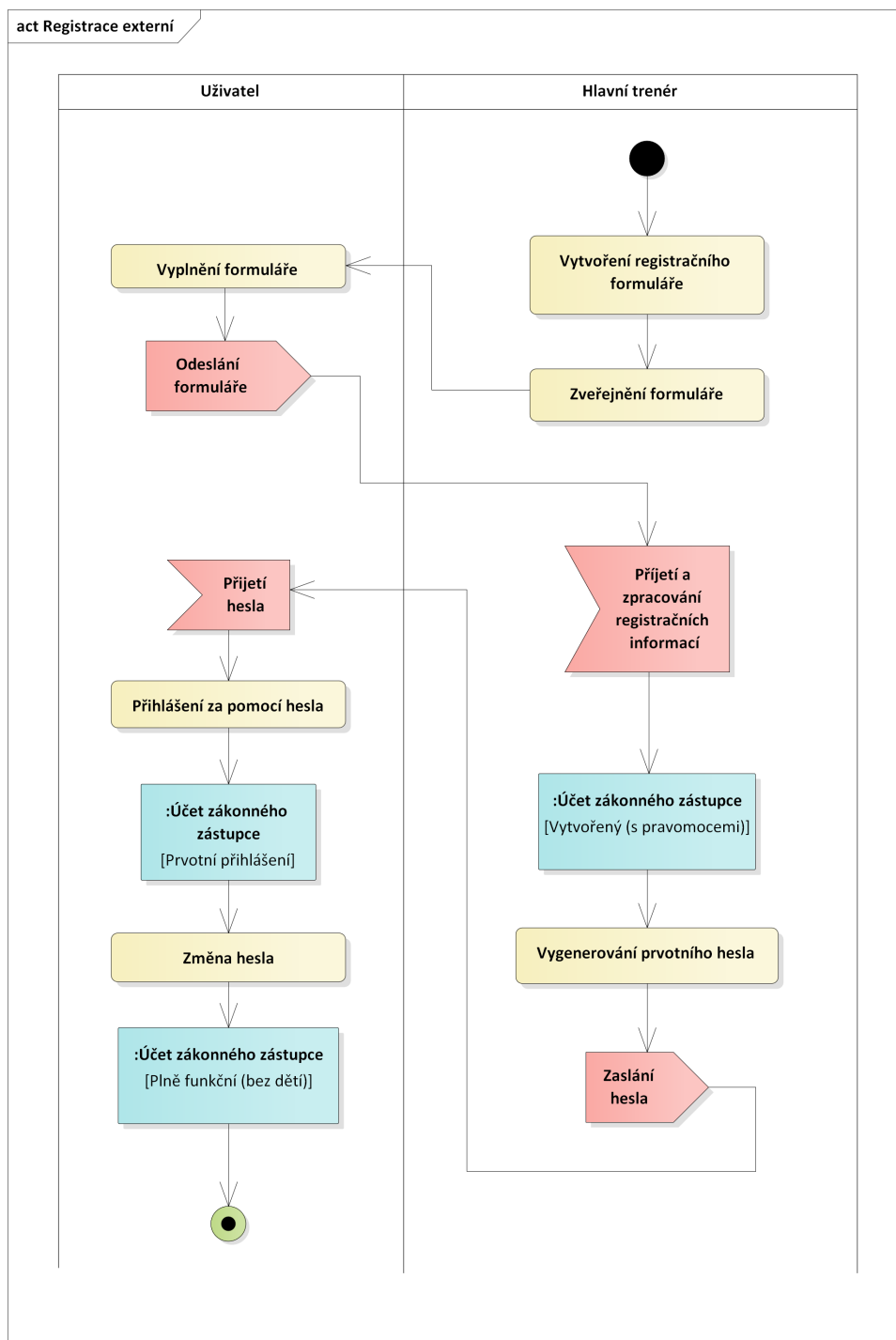
Registrující se uživatel si sám vytvoří v aplikaci svůj účet a následně zašle danému hlavnímu trenérovi formulář k přiřazení role zákonného zástupce.



Obrázek 3.9: Diagram aktivit – interní registrace zástupce

Externí registrace zástupce

Registrující se uživatel vyplní a odešle přes e-mail hlavním trenérem vytvořený formulář k registraci. Po určené uzávěrce trenér formuláře zpracuje a zašle zpět prvotní heslo k nově vygenerovaným účtům. Uživatel se poté přihlásí k tomuto účtu a nastaví si vlastní heslo.



Obrázek 3.10: Diagram aktivit – externí registrace zástupce

3.2.4 Vytvoření hráče

Účet s rolí zákonného zástupce má možnost vytvořit účet spravovaného hráče, viz [Diagram aktivit – tvorba hráče]. Dle volby zástupce bude mít daný účet nastavený typ rodičovské kontroly, který udává hráčovu pravomoc nad daným účtem. Implicitně má rodič plnou kontrolu nad účtem dítěte. Účet hráče musí nutně mít platné datum narození, jelikož účet bez rodičovské kontroly vyžaduje, aby hráč byl starší 18 let. Po vytvoření může zástupce spravovat detaily účtu a zasílat za něj formuláře.

3.2.5 Zrušení účtu

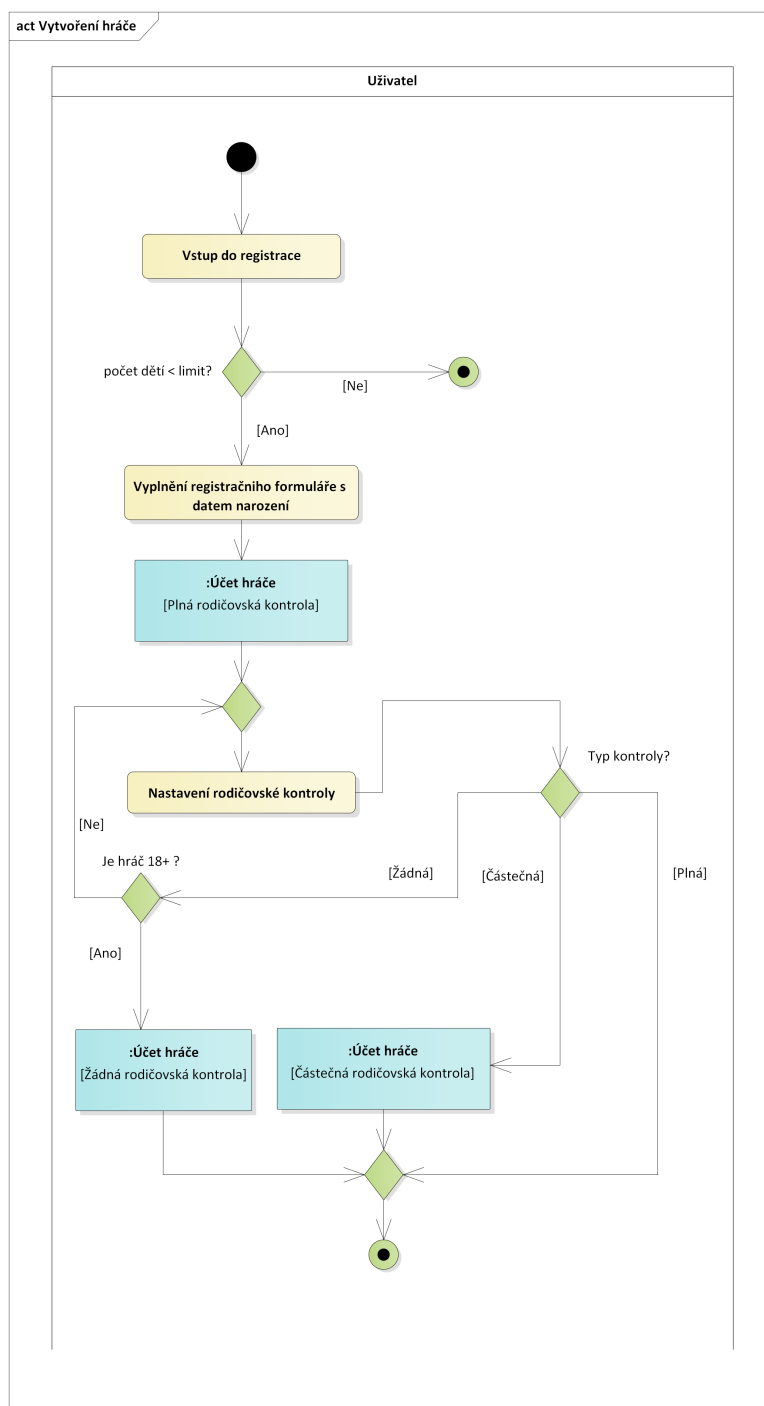
Kvůli případnému zachování hráčova účtu je připravena speciální procedura ke zrušení účtu pro zástupce s přiřazenými dětmi. V rámci této procedury zástupce rozhodne o účtech svých svěřených hráčů (v diagramech [Zrušení účtu] jako „dětí“).

Zrušení s dětmi

Pro zástupce s přiřazenými hráči je zde možnost účty hráčů osamostatnit (pouze hráči 18+), přiřadit hráče jinému zástupci nebo účty permanentně zrušit. Po vypořádání se s hráči je zrušení stejné jako u účtu bez dětí.

Zrušení bez dětí

Identické s běžným zrušením účtu. K běžnému zrušení účtu budou k dispozici dva způsoby výmazu – dočasné zrušení, kde bude změněn stav účtu, čímž bude uživateli znepřístupněn dokud jím nebude znovu obnoven (přes e-mailový odkaz), nebo permanentní zrušení, u kterého dojde k anonymizaci informací a účet již nebude obnovitelný. V případě dočasného zrušení bude uživateli na zadaný e-mail zaslán odkaz na případné obnovení účtu.

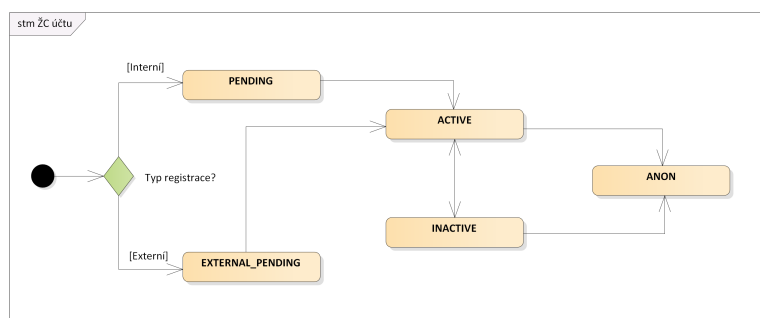


Obrázek 3.11: Diagram aktivit – tvorba hráče

3.3 Životní cykly

Účet

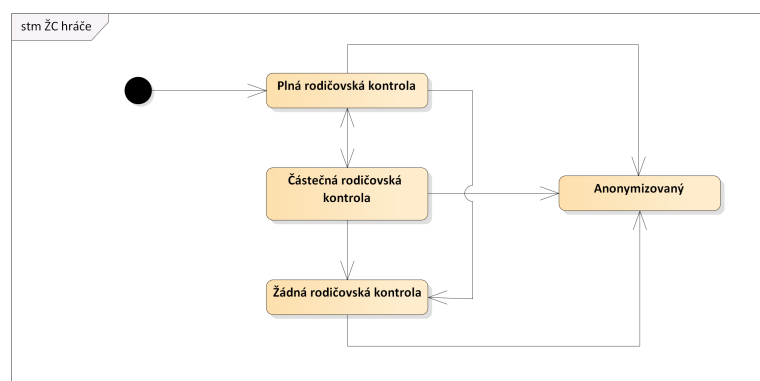
Diagram [Diagram aktivit – tvorba hráče] popisuje možné změny stavů účtu. Účely těchto stavů jsou blíže popsány v sekci Návrh tříd, části Registrace. Pro případy anonymizace neaktivního účtu, což je v případě uživatelské žádosti nutnost dle článku 17 v GDPR [34], budou administrátorovi přiděleny nástroje k provedení anonymizace daného účtu.



Obrázek 3.12: Diagram aktivit – tvorba hráče

Hráč

V diagramu [Diagram aktivit – tvorba hráče] lze vidět možné přechody mezi stavy účtu hráče. Nové vytvořený hráč bude mít vždy plnou rodičovskou kontrolu.



Obrázek 3.13: Diagram aktivit – tvorba hráče

3.4 Návrh tříd

3.4.1 Registrace

Účet

Pro účely kontroly stavů účtů byla do třídy **Account** přidána enumerace, která obsahuje stavy, ve kterých se účet může nacházet. Tyto stavy jsou:

PENDING Stav pro účty, které čekají na potvrzení e-mailem a dokončení registrace. V případě úspěšného potvrzení Přejdou do stavu **ACTIVE**. V tomto stavu není možné se do účtu přihlásit.

ACTIVE Stav pro úspěšně registrovaný účet v plném provozu. Uživatel s účtem v tomto stavu se může normálně přihlásit. Při dočasném zrušení uživatelem přejde do stavu **INACTIVE**, v případě trvalého výmazu do stavu **ANON**.

INACTIVE Dočasně neaktivní účet. Uživateli byl při zrušení účtu zaslán odkaz na obnovení. Pokud tento odkaz spustí, přejde účet opět do stavu **ACTIVE**. V tomto stavu není možné se do účtu přihlásit.

EXTERNAL_PENDING Účet, který byl registrován externě, čekající na přihlášení prvotním heslem. V případě úspěšného přihlášení bude uživatel vyzván ke změně hesla, po čemž mu bude změněn stav na **ACTIVE**.

ANON Anonymizovaný účet. Stav slouží převážně k indikaci anonymizovaného účtu v řádcích databáze.

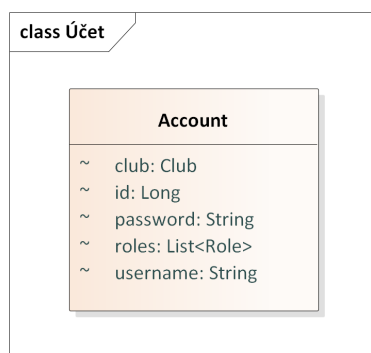
K tomu bude přidán atribut **birthdate**, který reprezentuje datum narození uživatele, spolu s **phone**, ve kterém je uloženo telefonní číslo, jenž bude použito u nouzového kontaktu.

Tabulka registrací

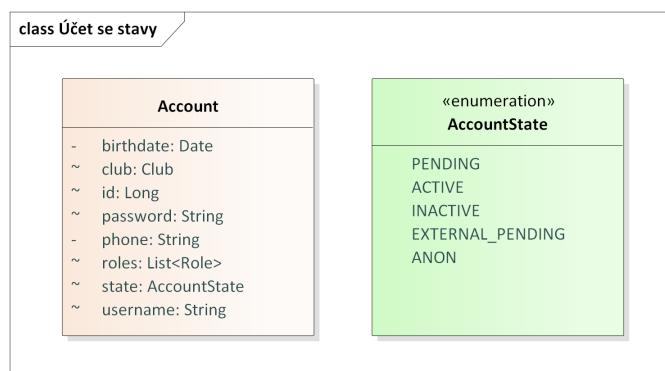
Kvůli kontrole a zasílání e-mailů uživatelům bude v rámci aplikační vrstvy zavedena třída **Registration**, ve které budou dočasně uloženy informace o uživateli. Tato třída bude následně kontrolována databázovým daemone, který bude uživatelům zasílat dané e-maily. Zmíněný daemon bude dodán třetí stranou (vedoucím práce).

Detaily této třídy budou upřesněny v rámci implementace na základě komunikace s vedoucím. Co ovšem bude obsahovat určitě, jsou e-maily uživatelů, vygenerovaná hesla u externích registrací a časovou stopu pro zrušení neaktivovaných účtů po uplynutí určité čekací doby.

3. NÁVRH



Obrázek 3.14: Návrh tříd – starý účet



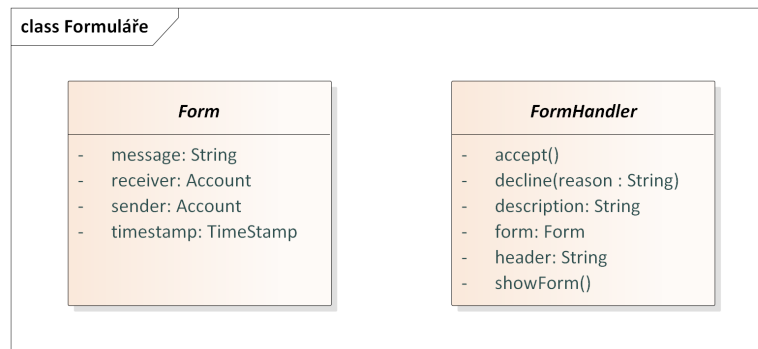
Obrázek 3.15: Návrh tříd – nový účet

3.4.2 Formuláře

Pro tvorbu různých typů formulářů byla vytvořena dvojice abstraktních tříd zařizujících jejich přenos do databáze a následné zobrazení s definicemi funkcí pro příjem či odmítnutí. K vytvoření nového typu formuláře je nutno implementovat třídy od nich dědící a k tomu fragment definující prostředí ve kterém budou uživatelem tvořeny.

Form Třída pro přenos relevantních informací daného formuláře. Obsahuje informace o odesílateli, příjemci a nepovinně krátkou zprávu od odesílatele.

FormHandler Třída upřesňující způsoby zobrazení informací uložených ve formuláři a funkce upřesňující akce v případě jeho přijetí a odmítnutí. Tato třída je pouze na straně klienta.



Obrázek 3.16: Návrh tříd – Formuláře

Typy formulářů

V rámci této práce byly navrženy následující typy formulářů.

InfoForm Tento formulář je určen k zasílání zpětné vazby uživatelům v případě úspěšného přijetí či odmítnutí jiného typu formuláře a uživatel jej sám nemůže přímo zaslat. Daný typ formuláře ukládá navíc pouze informace o vlastní hlavičce zprávy, která bude zobrazena v detailu formuláře z inboxu.

RoleForm Formulář k přiřazení nebo odebrání jedné určité role. Obsahuje informace o dané roli, subjektu a akci odebrání či přidání.

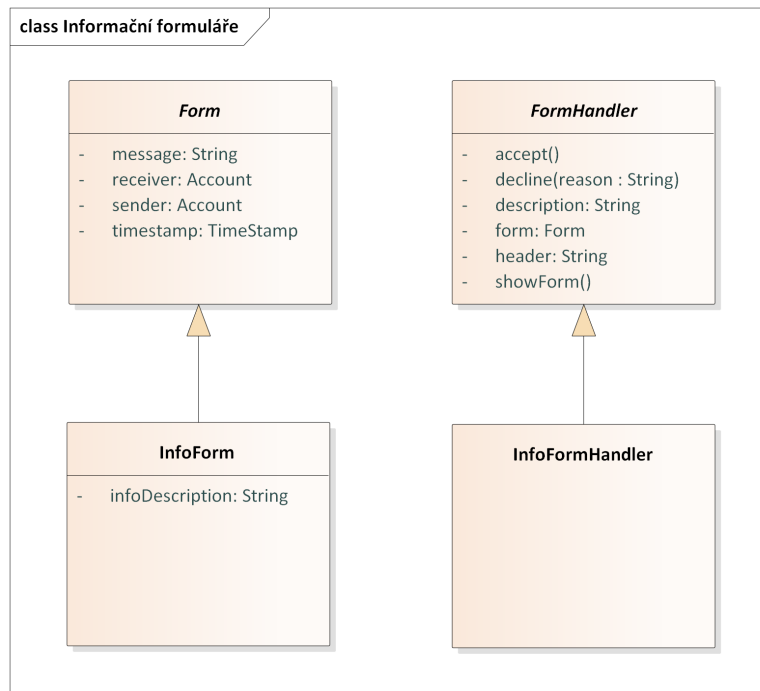
PlayerTransferForm Účel tohoto formuláře je přeřazení hráče od jednoho zástupce k jinému. Jediná přebývající informace je role hráče, kterého chce zástupce přesunout.

Návrh zbytku formulářů byl kvůli rozsahu práce vynechán. Návod na jejich implementaci lze vidět v kapitole Implementace, sekce Programátorská příručka – Formuláře

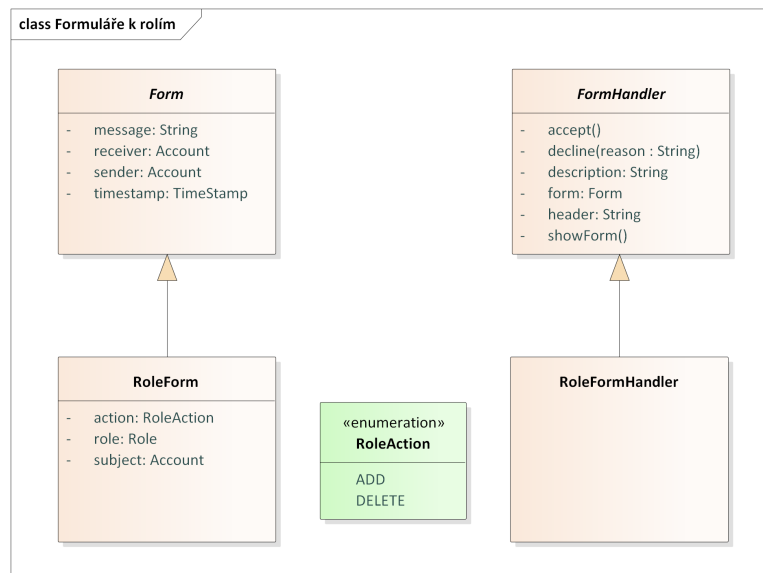
3.4.3 Role

K abstraktní třídě **Role** byly implementovány její podtřídy **Parent** a **Player**, reprezentující role rodiče, resp. hráče. Pro iniciální jednoduchost vztahů a správy hráčů byla mezi těmito třídami zavedena vazba 1:M, tedy hráč má jednoho rodiče, rodič má nula nebo více hráčů. Tato vazba by měla být do budoucna změněna na M:M. Hráč má navíc atribut **parentalControl**, který udává úroveň jeho rodičovské kontroly.

3. NÁVRH

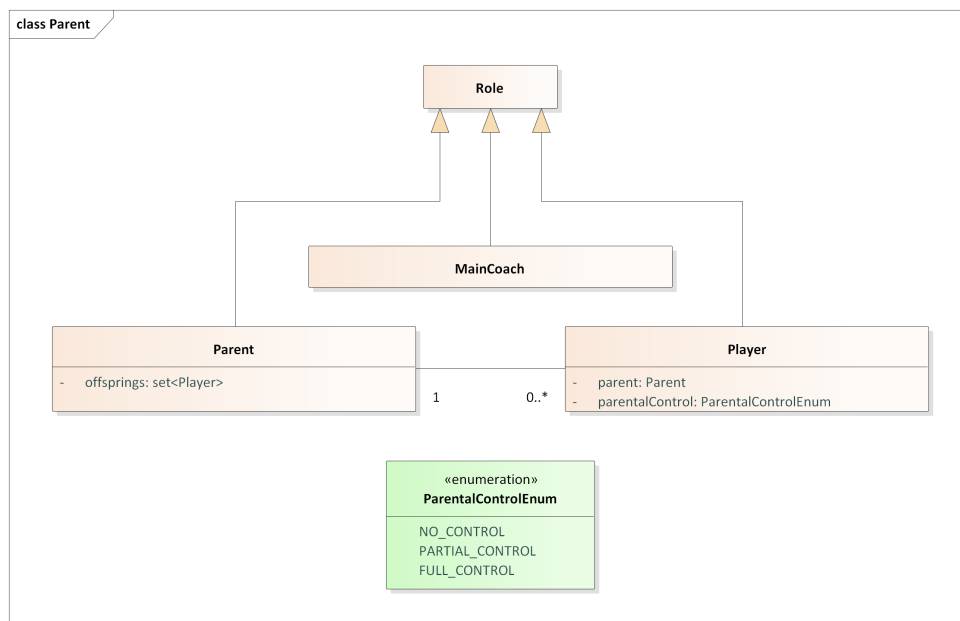


Obrázek 3.17: Návrh tříd – informační formuláře



Obrázek 3.18: Návrh tříd – formuláře k rolím

Další rolí, která byla vytvořena, je Hlavní trenér, neboli **MainCoach**, jehož detailní zavedení není subjektem této práce, proto zde bude mít pouze lehkou implementaci.



Obrázek 3.19: Návrh tříd – role

3.5 Návrh uživatelského rozhraní

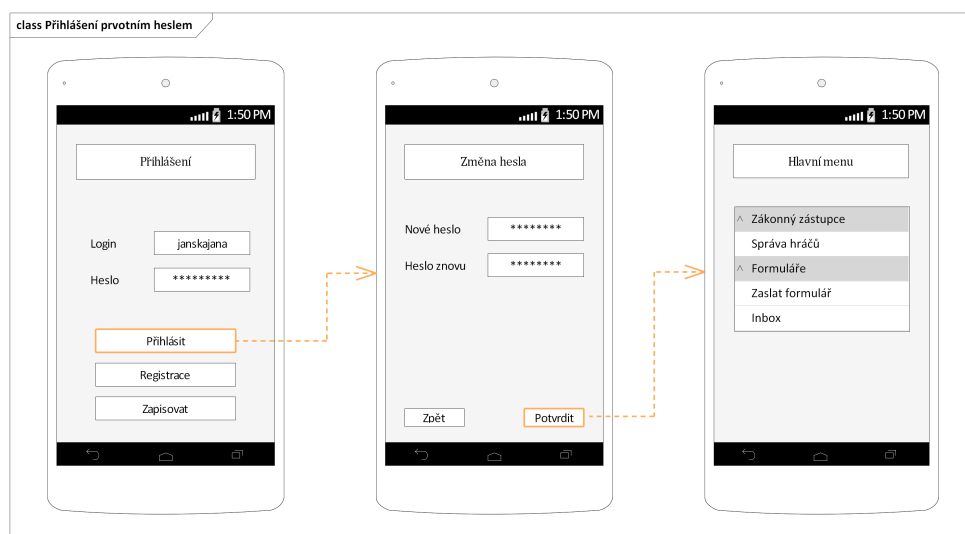
Zde budou detailněji popsány funkcionality a návrh jejich rozhraní.

3.5.1 Přihlášení a prvotní heslo

Přihlášení bude snadno dostupné úvodní obrazovky a proběhne vyplněním uživatelských detailů a stiskem tlačítka „Přihlásit“. Při úspěšném přihlášení dojde k přesunu do hlavního menu aplikace.

V rámci externí registrace bude uživatelům vygenerováno prvotní heslo, s jehož pomocí se mohou přihlásit. Po přihlášení se setkají s obrazovkou, ve které budou přinuceni nastavit si heslo vlastní. Po úspěšné změně hesla následuje, jako u běžných účtů, hlavní menu.

3. NÁVRH



Obrázek 3.20: Wireframe – prvotní přihlášení

3.5.2 Registrace

Pro interní registraci účtu je nutno spustit aplikaci a stisknout tlačítko „Registrace“. Poté bude uživatel přesměrován k registračnímu formuláři, kde vyplní e-mailovou adresu, heslo s jeho potvrzením a vybere příslušný klub. Pod těmito kolonkami bude nutno odkliknout checkbox, jehož doprovodný text uživatele informuje, že je potřeba souhlasit se smluvními podmínkami pro užívání aplikace. Po úspěšném vyplnění všech kolonek a odkliknutí zaškrťovacího políčka bude možno potvrdit registraci a uživatel bude přesunut do hlavního menu pod tímto novým účtem.

V případě chybného zadání některé kolonky (existující/neplatný e-mail, slabé heslo, nesprávné potvrzení hesla, ...) se pod kolonkou objeví rudá chybová hláška s popisem problému a bude potřeba úpravy dané kolonky pro úspěšné dokončení registrace.



Obrázek 3.21: Wireframe – registrace

3.5.3 Funkce modulu přístupné z hlavního menu

V rámci tohoto modulu bude ustanoven přístup k následujícím funkcionalitám za pomoci tlačítek z hlavního menu.

Zaslání formuláře

Formuláře budou dostupné všem účtům (kromě hráčů s plnou rodičovskou kontrolou) pod rozkliknutelným seznamem s nápisem „Formuláře“. V tomto seznamu bude dostupné tlačítko „Zaslat formulář“, které dovede uživatele k možností vyplňování a zasílání formulářů. Zde bude k dispozici seznam tlačítek nadepsaných druhy odeslatelných formulářů, po jejichž stisknutí bude zobrazeno okno s vyplnitelnými políčky pro informace nutné k zaslání formuláře a splnění daného úkonu.

Inbox

Dalším tlačítkem v seznamu bude „Inbox“, jež umožní přístup k reagování a zpracování formulářů, které byly uživateli zaslány. Každý formulář bude v této části reprezentován okénkem obsahujícím kratší nadpis a jménem (popř. mailem) odesílatele. Vedle tohoto okénka budou k dispozici dvě zaškrťovací políčka, jedno reprezentující souhlas s provedením daného úkonu asociovaného formuláře, a druhé pro odmítnutí téhož formuláře. Je zde možnost stisknutí okénka příslušného formuláře, čímž bude ukázán detail obsahující podrobnější popis úkonu a, pokud byla napsána, zprávu od odesílatele. V případě stisku políčka pro odmítnutí vyskočí okénko, do nějž bude možno vepsat zdůvodnění ve formě krátké zprávy. Pokud je uživatel spokojen se sebou zadanými políčky, má možnost stisknout tlačítko „Potvrdit“, čímž provede svá rozhodnutí

3. NÁVRH

nad formuláři, či „Zpět“ ke zrušení svých voleb a návratu do hlavního menu. Ať už přijat nebo odmítnut, každý zpracovaný formulář zašle zpět tzv. „informační formulář“, který příjemce informuje, zda mu bylo vyhověno, či ne. Tento formulář lze pouze přečíst a přijmout, čímž bude vymazán z inboxu.

Správa dětí

Možnosti role zákonného zástupce budou k dispozici pouze účtům s danou rolí, a to v podobném stylu, jako formuláře. Pod seznamem s nadpisem „Zákonný zástupce“ bude (v rámci tohoto modulu) dostupné pouze tlačítko „Správa dětí“. Tímto tlačítkem se zástupce dostane k seznamu svých dětí a možností vytvoření jejich účtů. Při stisknutí jednoho dítěte v seznamu bude zástupce přenesen k nastavení detailů daného účtu, kde bude mít schopnost tyto detaily změnit a tyto změny potvrdit, či zrušit. Na rozdíl od běžného účtu, hráč má změnitelnou kolonku „E-mail“, kde může namísto platného e-mailu být zadána a upravena přezdívka. Přezdívku je nutno změnit na platný e-mail před osamostatněním účtu hráče od rodičovské kontroly.

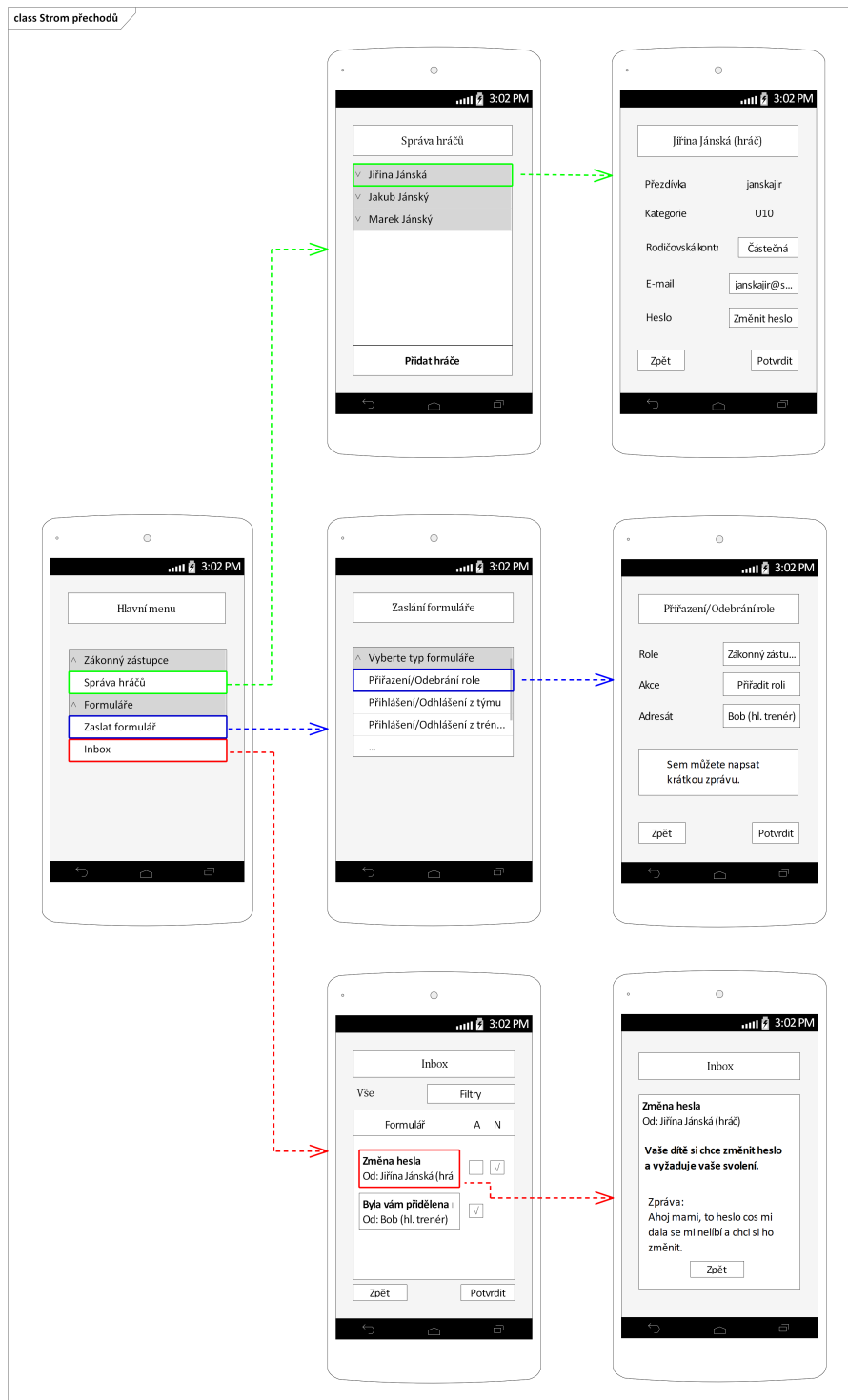
Nastavení účtu

Hlavní menu obsahuje „lištu“ s emailem přihlášeného účtu. Zde bude přidána ikona pro přechod k nastavení účtu, kde bude možno shlédnout a upravit jeho detaily. Tyto detaily jsou:

- E-mail
- Heslo
- Jméno
- Příjmení
- Klub
- Datum narození
- Telefonní číslo

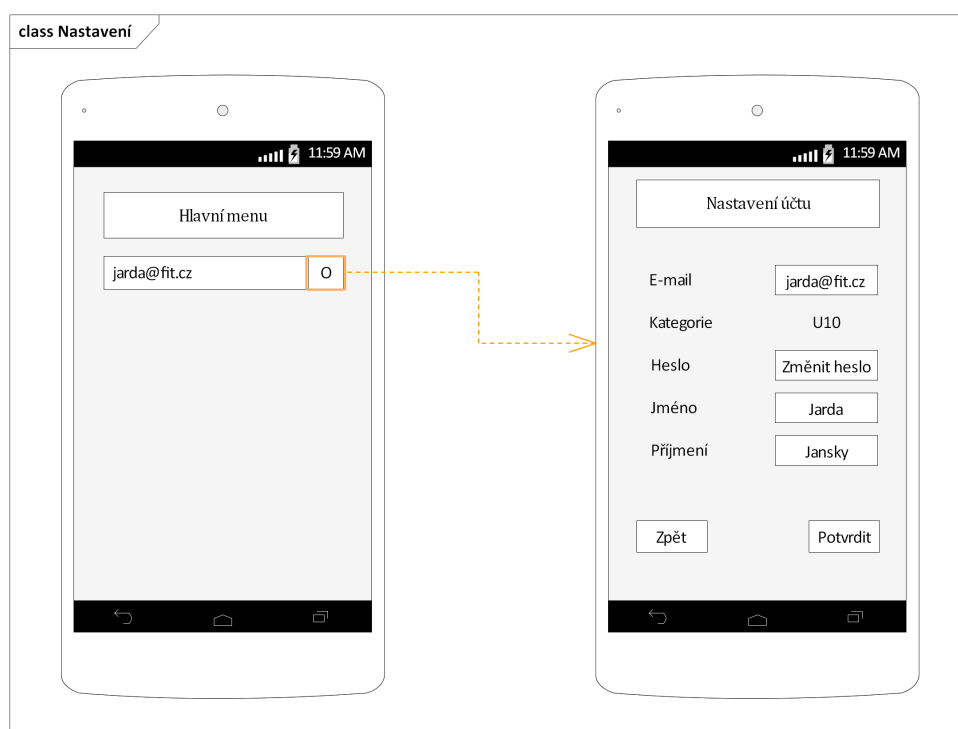
E-mail a klub zde nelze změnit, datum narození lze změnit pouze pokud předtím nebylo vyplněno, ale třeba jméno s příjmením lze měnit kdykoliv. V případě chybného zadání detailu bude uživatel upozorněn chybovou hláškou a zadané změny do opravy neproběhnou. Při stisku tlačítka na vyplnění data narození vyskočí dialogové okno, které uživatele přivítá selekcí dnů, měsíců a let. Změna e-mailu je plánována jako budoucí funkcionality v rámci formulářů.

3.5. Návrh uživatelského rozhraní



Obrázek 3.22: Wireframe – přechody z hlavního menu

3. NÁVRH



Obrázek 3.23: Wireframe – nastavení účtu

Implementace

4.1 Změny oproti návrhu

Při implementaci docházelo k drobnějším funkčním i designovým změnám původních návrhů funkcionalit a jejich uživatelského rozhraní. Tyto změny zahrnují: přidání či odebrání tlačítek, změny poměrů stran prvků UI, barevné rozlišování formulářů, možnosti aktualizace aktivit/fragmentů, atp.

4.1.1 Vynechané funkcionality

Kvůli rozsahu práce byly na základě dohody s vedoucím z implementace vynechány následující funkcionality s nízkou prioritou:

- Formuláře k přihlášení a odhlášení hráčů k tréninkům či turnajům
- Formulář s žádostí zástupci o změnu detailu účtu
- Nouzový kontakt
- Potvrzení registrace přes e-mail
- Externí registrace
- Možnosti obnovení dočasně smazaného účtu

Části některých z těchto funkcionalit jsou již implementovány, avšak ne v dostačujícím stavu. Návodů na jejich implementaci se budeme zabývat v kapitole Programátorská příručka. Další vynechanou částí implementace jsou typografické, rozměrové a stylistické úpravy u některých oken, zvláště v případech větších obrazovek s jiným rozlišením.

4.2 Instalační příručka

Zde se proberou možné způsoby instalace obou součástí aplikace a jejich detaily.

4.2.1 Serverová část

V následujícím návodu předpokládáme, že zatím nebyl spuštěn oficiální server se službou této aplikace. V opačném případě stačí nainstalovat pouze klienta, viz sekce Mobilní klient.

Tuto část lze sestavit z přiloženého zdrojového kódu ve složce `server`. Tento kód byl vytvořen v prostředí IntelliJ IDEA, které je silně doporučeno pro jeho úspěšné spuštění. Ideální verze pro instalaci je 2021.1 s libovolnou edicí (Community/Ultimate). Před instalací samotného projektu bude potřeba stáhnout Java vývojovou sadu JDK, v doporučené verzi 13. Poté stačí jen tento projekt v prostředí otevřít (File -> Open) a bude automaticky nakonfigurován. Server bude zapnut spuštěním funkce `main` v souboru `RugbyApplication.kt`.

4.2.2 Mobilní klient

V době psaní práce není aplikace na Google Play, což je oficiální a doporučená platforma pro instalaci Android aplikací. Z tohoto důvodu je nutno projekt buďto stáhnout a nainstalovat ve formě souboru `.apk`, nebo skrze přiložený zdrojový kód pod názvem `client`. Následující sekce popisují instalaci na mobilní zařízení s verzí Android 9.0 a výše, za funkčnost na jiných verzích či platformách autoři projektu neručí.

APK soubor

Pro instalaci bude zřejmě nutné povolit na svém zařízení instalaci neznámých aplikací. Toto se od verze 8.0 liší na zařízeních značky Samsung a ostatních. Na Samsung zařízení je potřeba následující navigace z hlavní obrazovky:

```
Nastavení (Settings) -> Biometrika a zabezpečení (Biometrics and security) -> Instalovat neznámé aplikace (Install unknown apps).
```

Zde se musí vybrat zvolený prohlížeč, kterým bude soubor stažen a stisknout možnost **Povolit z tohoto zdroje (Allow from this source)**. Na ostatních zařízeních by měla navigace vypadat takto:

```
Nastavení (Settings) -> Zabezpečení (Security) -> INSTALACE APLIKACÍ (APP INSTALLATION) -> Instalace z neznámých zdrojů (Unknown source installations)
```


a zde povolit prohlížeč. Poté stačí tímto prohlížečem stáhnout daný soubor a zařízení ho samo nainstaluje. Jakožto prohlížeče lze zvolit nejen ty internetové, ale také souborové, tedy lze spustit a instalovat .apk soubor vložený z jiného zařízení.

Kód v Android Studio IDE

K instalaci přes kód je potřeba nainstalovat Android Studio (verze 4.1.3) a otevřít v něm soubor s projektem. V tomto prostředí je možnost spuštění aplikace buďto na skutečném zařízení, nebo ve virtuálním.

Skutečné zařízení Ke spuštění aplikace na zařízení je zde nutnost aktivace vývojářského režimu. Od verze Android 9 to lze následovně: přechody

Nastavení (Settings) -> O telefonu (About Phone) ->
Informace o softwaru (Software information)

a zde je potřeba sedmkrát stisknout kolonku **Číslo sestavení (Build number)**. Nyní byl aktivován vývojářský režim, který zpět v nastavení zobrazí možnost **Vývojářské možnosti (Developer options)** ve které lze povolit **Ladění USB (USB debugging)**. Poté stačí zařízení připojit přes USB k přístroji s Android Studiem, které ho vyhledá a zobrazí jako možný cíl instalace zařízení. S vybráním tohoto zařízení stačí program spustit skrze funkcionalitu **Run** (Shift+F10 nebo zelená šipka).

Alternativou je připojení zařízení přes Wi-Fi, které je mírně složitější nastavit a lze ho vidět na oficiální stránce mezi návody [35].

Virtuální stroj Ke spuštění virtuálního stroje je nutná instalace Android Virtual Machine (AVD). Po jeho instalaci lze nastavit do seznamu zařízení nové AVD, jehož image je potřeba stáhnout. Aplikaci lze na vybraném stroji následně nainstalovat a spustit, opět, funkcí **Run**.

Vlastní APK Android Studio umožňuje export projektu do vlastního souboru .apk, který je možný navigací přes lišty

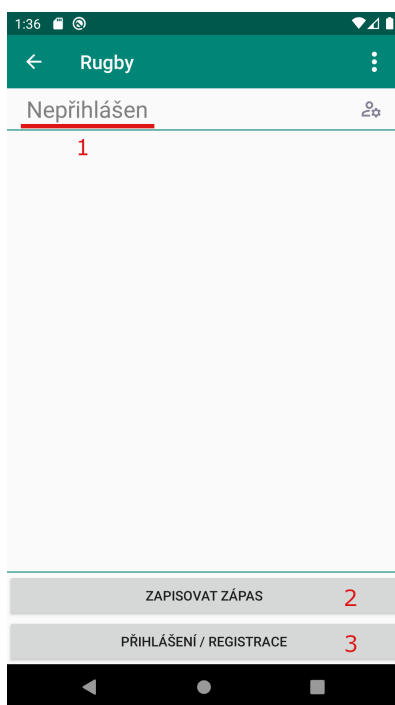
Build -> Build Bundle(s)/APK(s) -> Build APK(s).

Vytvořený soubor lze na zařízení nainstalovat dle sekce APK soubor.

4.3 Uživatelská příručka

Tato část se zabývá přiblížením funkcionalit aplikace a navigace k nim běžnému uživateli. Funkcionality zde zmíněné jsou omezeny pouze na ty, které byly realizovány či upraveny v rámci tohoto modulu.

Úvodní obrazovka

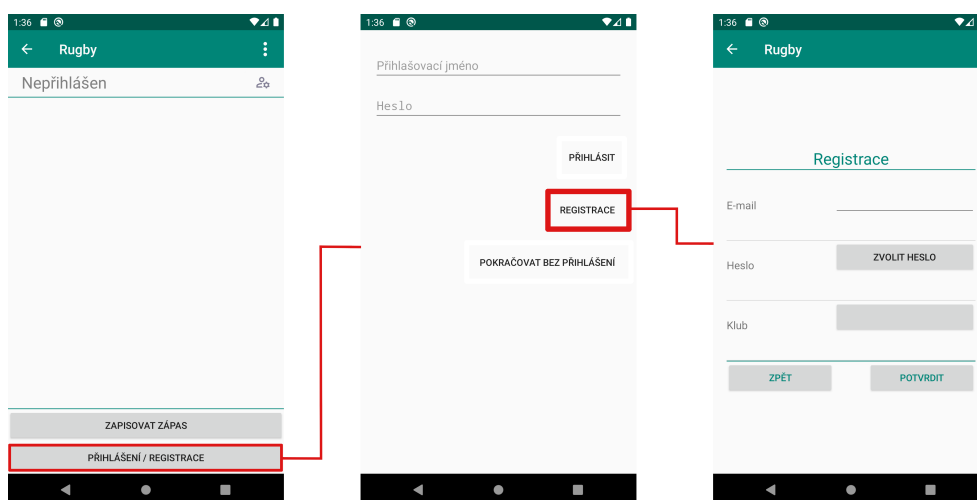


Obrázek 4.1: Úvodní obrazovka

Tuto obrazovku lze vidět na obrázku [Úvodní obrazovka]. Úvodní obrazovka se mění v závislosti na tom, zda je uživatel přihlášen, nebo ne. Pokud přihlášen není (1), má k dispozici pouze tlačítka „Zapisovat zápas“ (2) a „Přihlášení/Registrace“ (3). Chování této obrazovky s přihlášeným uživatelem je vysvětleno v kapitole Hlavní menu.

Registrace

K registraci se z úvodní obrazovky uživatel dostane přes tlačítka „Přihlášení/Registrace“. Následující obrazovka obsahuje dvě kolonky k vyplnění a možnost „Přihlásit“, „Registrace“ a „Pokračovat bez přihlášení“. Při kliknutí na možnost „Registrace“ bude uživatel přesunut na registraci účtu. Zde lze zadat do kolonky e-mail a přes tlačítka nastavit heslo a klub. Bude-li e-mail neplatný



Obrázek 4.2: Navigace k registraci a přihlášení

nebo již využitý, heslo nebude vyplněno, nebo nebude vybrán klub, aplikace na to upozorní chybovou hláškou. U hesla je nutno mít alespoň 8 znaků s minimálně dvěma čísly. Pokud si uživatel přeje dokončit registraci může stisknout tlačítko „Potvrdit“, jinak „Zpět“. Po potvrzení či návratu bude uživatel vrácen zpět na obrazovku přihlášení

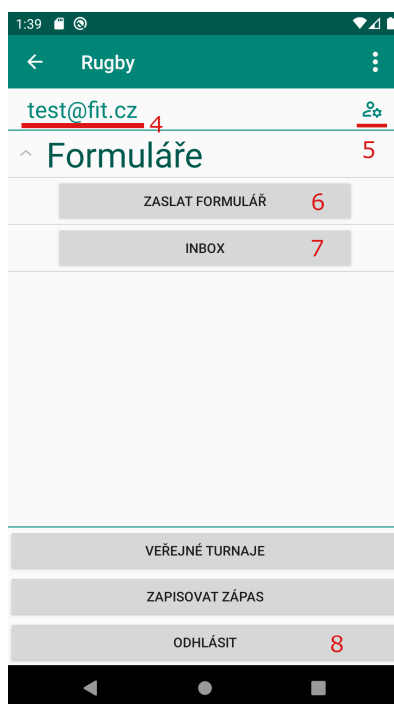
Přihlášení

Na přihlašovací obrazovce je možnost přihlášení úspěšně registrovaným účtem vyplněním korektního e-mailu a hesla, a stisknutím tlačítka „Přihlásit“. Přihlášení selže za předpokladu, že vyplněné hodnoty nebyly správné, účet nebyl zatím aktivován, nebo došlo ke zrušení účtu. Pokud nedojde k selhání, následuje přechod do hlavního menu.

Hlavní menu

V případě že byl uživatel úspěšně přihlášen, dojde k transformaci úvodní obrazovky do hlavního menu (obrázek [Hlavní menu]). V tomto menu se zobrazí e-mail (4), tlačítko k nastavení účtu (5), formuláře (6,7) a popř. další funkcionality spojené s přiřazenými rolemi. Také se lze kdykoliv z tohoto menu odhlásit (8).

4. IMPLEMENTACE



Obrázek 4.3: Hlavní menu

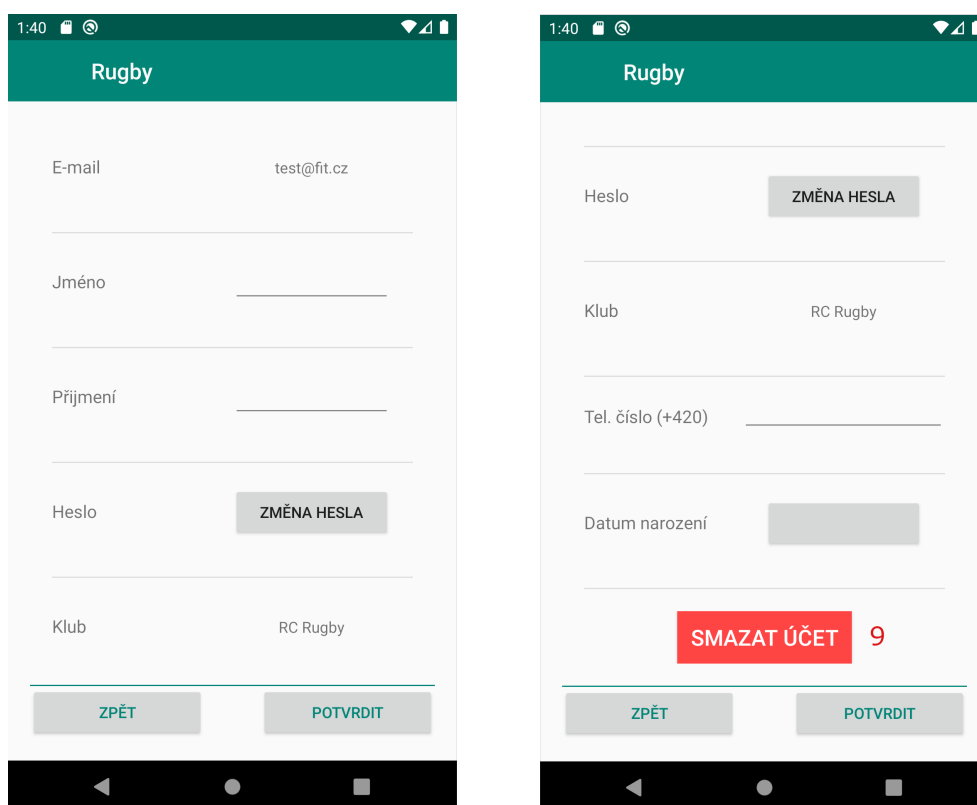
Nastavení účtu

Stiskem tlačítka v pravém horním rohu hlavního menu (5) bude majitel účtu schopen shlédnout a měnit detaily svého účtu. Nalézá se zde:

- E-mail
- Jméno
- Příjmení
- Heslo
- Klub
- Telefonní číslo
- Datum narození
- Smazání účtu (9)

Detaily ke smazání účtu viz sekce Smazání účtu. Některé tyto detaily nelze změnit, avšak další, jako je jméno a příjmení, lze měnit vždy. Ke změně hesla je nutno potvrzení heslem starým – pokud se nebude shodovat, změna nenastane.

Datum narození lze pouze jednorázově nastavit a momentálně není způsob jak ho změnit. V budoucnu budou možnosti úpravy těchto detailů realizovány přes formuláře zaslané administrátorovi. Pokud si jedinec přeje zadané změny uložit, musí stisknout tlačítko „Potvrdit“, pokud ne tak „Zpět“. Po stisku jednoho z těchto tlačítek dojde k navrácení do hlavního menu.



Obrázek 4.4: Nastavení

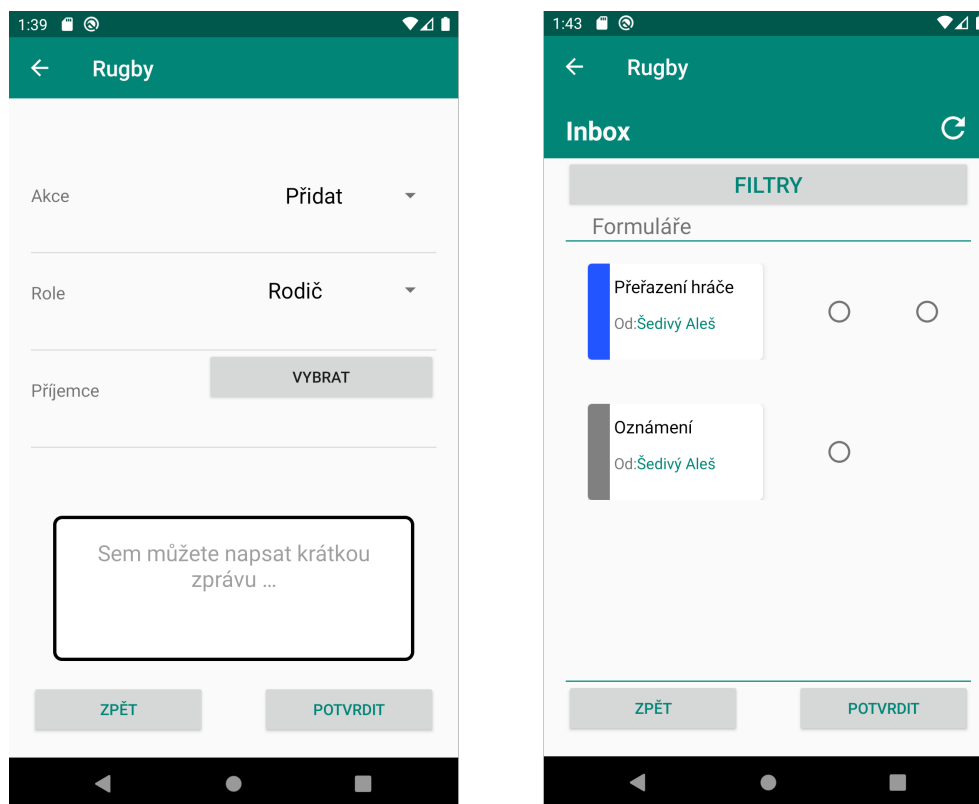
Zaslání formuláře

Vyplnění a zaslání formuláře uživatel docílí tím, že stiskne tlačítko „Zaslat formulář“ (6). Zde mu bude ukázán výběr všech jemu přístupných formulářů. Po výběru stiskem příslušného typu se dostane k možnostem vyplnění daného formuláře. Tyto možnosti se většinou liší formulář od formuláře, proto si zde ukážeme pouze plnění u „Přiřazení/Odebrání role“.

Jak lze vidět na obrázku [Zaslání formuláře a Inbox], uživatel má schopnost změnit akci „Přidat/Odebrat“, roli, na kterou chce danou akci aplikovat a příjemce formuláře. Dále je dole okénko s možností zapsání krátké správy

4. IMPLEMENTACE

tomuto příjemci. K odeslání vyplněného formuláře poté stačí jen stisknout „Potvrdit“.



Obrázek 4.5: Zaslání formuláře a Inbox

Příjem a reakce na formuláře

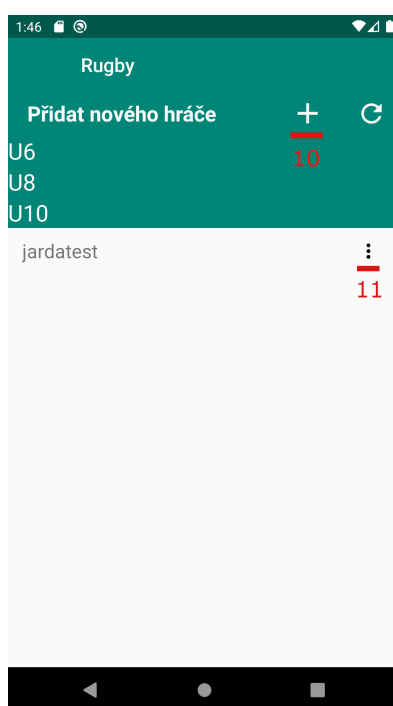
Ke shlednutí a reakci na příchozí formuláře je nutno z hlavního menu přejít do Inboxu. Zde jsou vidět formuláře, po jejichž rozkliknutí bude zobrazen detail jejich účelu a zpráva od odesílatele. K jejich potvrzení a zpracování je potřeba stisknout kruhový checkbox vlevo a pro odmítnutí ten vpravo. Při odmítnutí vyskočí možnost zapsání důvodu zamítnutí akce. Poté už jen stačí stisknout tlačítko „Potvrdit“ a uvedené reakce budou zpracovány.

Získání role rodič

K přiřazení role rodič (zákonný zástupce) stačí zaslat formulář (viz Zaslání formuláře) o přiřazení role rodič.

Registrace a správa hráče

Pokud byla role úspěšně přidána, lze nyní v hlavním menu spatřit lištu s nápisem „Rodič“. Po rozkliknutí budou pod touto lištou možnosti k hráčům, mezi nimi tlačítko „Správa hráčů“. Tímto tlačítkem se uživatel dostane ke správě hráčů (na obrázku [Správa hráčů]), kde je možné skrze tlačítko (10) registrovat nového hráče, nebo přes (11) upravit účet již existujícího. Daná úprava je téměř identická s tou v Nastavení účtu, rozdíly jsou ve změnitelnosti e-mailu/přezdívky a přidáním výběrem „Rodičovská kontrola“. Nutná poznámka, účet hráče lze zatím smazat pouze trvale.



Obrázek 4.6: Správa hráčů

Rodičovská kontrola

Úroveň kontroly lze změnit v nastavení hráče, viz sekce výše. Nyní existují tři typy kontroly:

- Žádná kontrola
- Částečná kontrola
- Plná kontrola

U „žádné kontroly“ má hráč úplnou kontrolu nad správou svého účtu a zástupce se může pouze dívat na jeho details. U „částečné kontroly“ má rodič zatím stejné pravomoci jako u plné, ale hráč se může dívat na details svého účtu. Při „plné kontrole“ může rodič kdykoliv měnit nastavení hráčova účtu, ke kterým hráč nemá přístup, stejně jako k formulářům.

Tyto úrovně lze kdykoliv změnit, kromě „žádné kontroly“, pro kterou je potřeba aby hráč byl starší 18 let a měl zadán platný e-mail. Po nastavení „žádné kontroly“ nemůže rodič již tuto kontrolu změnit.

Smazání účtu

Smazání účtu probíhá stiskem rudého tlačítka „Smazat účet“ (9). V případě, že uživatel má roli rodič a k ní přidělené hráče, bude požádán o jejich eliminaci skrze zbavení kontroly, přeřazení jinému zástupci, popř. jejich smazáním přes potvrzení tohoto okna. Pokud dojde k rozhodnutí smazat hráče, bude obrazovka obnovena a dojde k nutnosti stisknout tlačítko znovu.

Pokud účet hráče přiděleny nemá, dojde k výběru druhu výmazu, dočasného či trvalého. Dočasný výmaz deaktivuje účet, avšak bude zde možnost obnovy v budoucnu skrze e-mailový odkaz. Trvale smazaný účet již obnovit nepůjde. Po výběru vyskočí okénko s popisem výmazu, se kterým musí uživatel souhlasit, čímž dojde k odhlášení účtu, který bude následně vymazán.

4.4 Programátorská příručka

Tato sekce obsahuje návody, typy a triky, spolu s několika implementačními details, jejichž účel je seznámení programátora s rozšiřováním prvků vyvinutých či započatých v rámci tohoto modulu.

4.4.1 Formuláře

Zavedení formulářů vyžaduje specifické zásahy do serverové části i klienta. Zde si popíšeme, jak postupovat při tvorbě nového typu formuláře.

4.4.1.1 Aplikační vrstva

Pro vytvoření nového formuláře je potřeba zavést tyto objekty s ním související: entitu (entity), repozitář (repository), službu (service) a ovladač (controller).

Entita V kódu [Entita formuláře] můžete vidět „šablonu“ na tvorbu formulářové entity. Tuto třídu Spring se svým ORM přemění na databázovou entitu díky anotaci `@Entity`. `@DiscriminatorValue` udává (unikátní) hodnotu, dle které bude odlišena od zbytku podtříd třídy `Form`, abstraktní třídy, která reprezentuje všechny formuláře. Je-li daný formulář ve vztahu s jinou entitou, je

ji zde nutno uvést a označit její kardinalitou (1:1, 1:M, ...). Kompletní třídu vložte do složky /entities/forms.

```

1 /* Nahradte texty obalene
2  * znaky ## prislusnymi hodnotami
3  */
4
5 @Entity
6 @DiscriminatorValue("#IDENTIFIKATOR_FORMULARE#")
7 class #Typ formulare# : Form() {
8
9     @#Kardinalita (OneToMany, ManyToMany, ...)#
10     var #Nazev atributu# : #Typ atributu#? = null
11 .
12 .
13 .
14 }

```

Listing 4.1: Entita formuláře

Poté je nutno ve zmíněné třídě Form přidat nový řádek, do kterého vložíte příkaz `JsonSubTypes.Type` dle výňatku [JSON podtyp], řádku 7. Toto umožní výše popsané rozlišování.

```

1 @Entity
2 @JsonTypeInfo(use= JsonTypeInfo.Id.NAME, include = JsonTypeInfo.As.
3   PROPERTY, property = "className", visible = true)
4 @JsonSubTypes(
5     JsonSubTypes.Type(value = RoleForm::class, name = "FORM_ROLE"),
6     JsonSubTypes.Type(value = InfoForm::class, name = "FORM_INFO"),
7     JsonSubTypes.Type(value = PlayerTransferForm::class, name = "FORM_TRANSFER"),
8     JsonSubTypes.Type(value = #Typ formulare#::class, name = "#IDENTIFIKATOR_FORMULARE#")
9 )
10 @Inheritance(strategy = InheritanceType.SINGLE_TABLE)
11 @DiscriminatorColumn(name = "FORM_NAME", discriminatorType = DiscriminatorType.STRING)
12 @JsonIgnoreProperties(ignoreUnknown = true)
13 open class Form {
14 .
15 .

```

Listing 4.2: JSON podtyp

Repozitář Tato část je velice triviální, stačí pouze implementovat do složky /entities/repositories soubor obsahující následující rozhraní:

```

1 interface #Typ formulare#Repository : FormRepository< #Typ formulare#> {}

```

Listing 4.3: Repozitář

4. IMPLEMENTACE

Daná implementace deklaruje kontejner, ve kterém budou uskládány vaše formuláře.

Služba Služba slouží k definici transakcí nad určitými repositáři. Tyto repositáře jsou definovány jako proměnná s pozdní inicializací, spolu s anotací `@Autowired`. Jejich inicializaci zařizuje Spring. Transakce jsou na druhou stranu funkce s různými uživatelem definovatelnými vstupy a výstupy, označeny `@Transactional`. V následující ukázce lze vidět, jak takovou službu vytvořit. Funkce, kterou zde lze vidět, slouží k získání všech formulářů daného typu, které byly zaslány jednomu účtu daného identifikátorem v parametru. Je vhodné také vytvořit triviální funkce k ukládání a mazání formulářů z repositáře, ideálně za pomoci jeho vestavěných metod `save` a `deleteById`. Služby se nachází ve složce `/services`.

```
1 @Service
2 class #Typ formulare#Service {
3
4     @Autowired
5     lateinit var #Typ formulare#Repository: #Typ formulare#
      Repository
6
7     @Transactional
8     fun get#Typ formulare#s(receiverId : Long) : List< #Typ
      formulare# > {
9         return #Typ formulare#Repository.getFormByReceiverId(
      receiverId)
10    }
11 .
12 .
13 .
14 }
```

Listing 4.4: Služba

Ovladač Anotace `@Controller` slouží k označení třídy jako ovladače. Ovladač mapuje REST URI příkazy zasílané klientem na svoje funkce, ve kterých bývají použity výše definované služby. Je vhodné vytvořit metody pro POST, GET a DELETE za pomoci značek `@PostMapping`, `@GetMapping` a `@DeleteMapping`. Hodnota zapsaná do těchto příkazů se spojí s předponou v anotaci `@RequestMapping` a vytvoří tím ucelené URI. Dále `@ResponseBody` určí výstup funkce jako tělo odpovědi klientovy, `@RequestParam` před parametrem jej bude vyžadovat v těle žádosti a `@RequestBody` si vyžádá v těle nějakou komplexnější třídu (třeba celý formulář). Pokud funkce nic nevrací, je doporučeno ji předělat tak, aby vracela `EmptyResponse()`, kvůli zpětné vazbě klientovi (ten to může brát jako problém s připojením). Složka, kde jsou ovladače obsaženy má název `/controllers`.

```
1 @Controller
2 @RequestMapping("/forms", produces = ["application/json"])
3 class #Typ formulare#Controller {
4
5     @Autowired
6     lateinit var accountService: AccountService
7
8     @Autowired
9     lateinit var #typ formulare#Service: #Typ formulare#Service
10
11     @PostMapping("/#typ formulare#")
12     @ResponseBody
13     fun post#Typ formulare#(
14         authentication: Authentication,
15         @RequestBody form: #Typ formulare#,
16         response : HttpServletResponse
17     ): EmptyResponse {
18         form.timeStamp = Timestamp(
19             Calendar.getInstance().timeInMillis
20         )
21         #typ formulare#Service.add#Typ formulare#(form)
22
23         response.status = HttpServletResponse.SC_OK
24         return EmptyResponse()
25     }
26 .
27 .
28 .
29 }
```

Listing 4.5: Ovladač

Toto je zhruba vše, co je potřeba k vytvoření nového typu formuláře ze strany serveru. O zbytek se automaticky stará Spring a Hibernate ORM.

4.4.1.2 Klient

Implementace na straně klienta je poněkud složitější. Je nutno zavést nové podtypy tříd `Form` a `FormHandler`, fragment pro jejich tvorbu, vytvořit žádosti na server a tento nový podtyp registrovat do `GsonParser`.

Žádost na server Pro žádost na server je nutno přidat metodu do příslušného interface v `/backendInterfaces` a následně ji implementovat ve shodujícím se `/networking/backendRequests` souboru, podobně jako na příkladu pro typ `InfoForm` níže. Daný příkaz využívá adresu URI, kterou je nutno zadat do `/res/values/strings.xml` ve formátu uvedeného v [URI string]

4. IMPLEMENTACE

```
1 override fun getReceiverInfoForms(  
2     successCallback: (List<InfoForm>) -> Unit,  
3     useProgressDialog: Boolean,  
4     failCallback: ((error: RequestError) -> Unit)?,  
5     connectionErrorCallback: (() -> Unit)?  
6 ) {  
7     requestHandler.getRequest(  
8         context.resources.getString(R.string.get_info_forms_request)  
9         ,  
10        Array<InfoForm>::class.java,  
11        null,  
12        successCallback,  
13        useProgressDialog,  
14        failCallback,  
15        connectionErrorCallback  
16    )  
17 }
```

Listing 4.6: Žádost na server

```
1 <string name="nazev_prikazu">adresaURI</string>
```

Listing 4.7: URI string

Form Nový podtyp třídy `Form` by měl svým obsahem odpovídat jeho serverovému protějšku, tedy jsou v něm atributy, jejichž informace je potřeba udržet v databázi. Jsou tu ale možnosti tvorby pomocných funkcí, které nebudou na server zaslány. Pro tento účel, hodnoty které zaslat chceme, označíme skrze `@Expose`, u ostatních jej vynecháme. Všechny formuláře jsou uloženy ve složce `/entities/forms/formtypes`.

```
1 class #Typ formulare#: Form() {  
2     @Expose  
3     var #nazev atributu# : #Typ atributu#? = null  
4 }
```

Listing 4.8: Formulář

FormHandler Nejsložitější částí je implementace `FormHandler`, třídy, která zaobaluje formulář výše a slouží jako jeho adaptér s definicemi chování v případě přijetí (funkce `accept()`), či odmítnutí (`decline(reason: String)`), k tomu hlavičku a detailnější popis účelu formuláře. Také je zde možnost, jak změnit vzhled formuláře v inboxu, hlavně barevného pruhu pro rozlišení různých typů formulářů (`showForm(layout: View)`). Funkce pro přijetí a odmítnutí by v sobě měli mít zahrnuté výše definované žádosti pro provedení formulářového úkonu, smazání formuláře a následně zaslání `InfoForm` původnímu odesílateli. Tento `InfoForm` by měl ve své hlavičce zahrnovat zprávu o přijetí či odmítnutí, v případě odmítnutí ještě zahrnout uživatelem zapsaný důvod. Tato třída se nachází ve složce `/entities/forms/formhandlers`.

```

1 class #Typ formulare#Handler(
2     form: #Typ formulare#,
3     val ctx : Context,
4     val connection: RequestAggregator
5 ): FormHandler(form) {
6     override val header: String = "#hlavicka#"
7
8     override val description: String = "#popis#"
9
10    override fun accept() : Unit {
11        #telo funkce#
12    }
13    override fun decline(reason : String) : Unit {
14        #telo funkce#
15    }
16
17    override fun showForm(layout : View) : View {
18        return super.showForm(layout)
19    }
20 }

```

Listing 4.9: FormHandler

```

1 override fun showForm(layout : View) : View {
2     var lay = super.showForm(layout)
3     lay.findViewById<View>(R.id.layout_form_type_color)
4         .setBackgroundColor(
5         layout.context.resources
6             .getColor(R.color.#id_barvy#)
7     )
8     return lay
9 }

```

Listing 4.10: Změna barvy pruhu

```

1 val infoOut = InfoForm()
2 infoOut.id = 0
3 infoOut.sender = form.receiver
4 infoOut.receiver = form.sender
5 infoOut.infoDescription = declineHeader
6 infoOut.messageString = reason
7
8 connection.formRequests().postInfoForm(infoOut, {}, true, {})

```

Listing 4.11: Odmítnutí – zaslání InfoForm

Nakonec stačí přidat řádek do when výrazu, který ve funkci `/fragments/formFragments/FormHandlerMapping.kt` namapuje typ formuláře na odpovídající typ handleru, jak lze vidět na řádcích 10-14 v kódu [FormHandler mapa].

4. IMPLEMENTACE

```
1 fun FormHandlerMapping(ctx: Context, form: Form, connection:
  RequestAggregator) : FormHandler {
2   return when (form) {
3     is RoleForm -> RoleFormHandler(form,ctx,connection)
4     is InfoForm -> InfoFormHandler(form,ctx,connection)
5     is PlayerTransferForm -> PlayerTransferFormHandler(
6       form,
7       ctx,
8       connection
9     )
10    is #Typ formulare# -> #Typ handleru#(
11      form,
12      ctx,
13      connection
14    )
15    else -> throw Exception("Unrecognised form type")
16  }
17 }
```

Listing 4.12: FormHandler mapa

Fragment pro tvorbu Nejprve je nutno vytvořit fragment a jeho interface, které bude definovat jeho callback funkce z aktivity `FormBuilderActivity`. Do této aktivity poté naimplementujte skrze interface přidané funkce a vložte do `FormBuilderMenuInterface` funkci novou, která daný fragment spustí. Příklad takového spuštění lze spatřit u [Spuštění fragmentu]. U jakých rolí bude spustitelný je potřeba nastavit ve fragmentu `FormBuilderMenuFragment`.

```
1 override fun run#Typ fragmentu#() {
2   val fragment = #Typ fragmentu#()
3   singleFragmentFunc.pushFragment(fragment)
4 }
```

Listing 4.13: Spuštění fragmentu

Po nastavení spuštění je nutné implementovat fragment samotný. K tomu byl vytvořen `layout_form_creation_fragment`, jehož inicializaci lze vidět ve [Fragment layout].

```

1 override fun onCreateView(
2     inflater: LayoutInflater,
3     container: ViewGroup?,
4     savedInstanceState: Bundle?
5 ): View {
6     if (context == null) throw IllegalStateException()
7     val view = inflater.inflate(
8         R.layout.layout_form_creation_fragment, container,
9         false
10    )
11    .
12    .
13    .
14    return view
15 }

```

Listing 4.14: Fragment layout

Tento layout obsahuje „nafouknutelný“ seznam, do kterého lze vložit libovolné definované pohledy (Views). V rámci této práce byly implementovány pohledy, jako `layout_single_attribute_with_error_button`, který obsahuje programovatelné tlačítko a chybovou hlášku. Takovýchto pohledů bylo vytvořeno více a lze je vložit např. jako v kódu [Vložení pohledu].

```

1 layoutManager = view.findViewById(R.id.form_builder_layout_list)
2
3 var #navez pohledu# = parentActivity.layoutInflater.inflate(
4     R.layout.layout_single_attribute_with_error_button,
5     null
6 ) as LinearLayout

```

Listing 4.15: Vložení pohledu

GsonParser Poslední část je jednoduchá, stačí v souboru `/login/json/GsonParser` vložit do třídy `formFactory` jeden řádek, viz [Registrace typu do GsonParser], řádek 5. Zde viditelný identifikátor musí odpovídat tomu, který byl zadán na straně serveru. Tímto bude zařízeno, že daná třída bude korektně zaslána a serverem zpracována.

```

1 val formFactory = RuntimeAdapterFactory.of(Form::class.java, "
2     className")
3     .registerSubtype(InfoForm::class.java, "FORM_INFO")
4     .registerSubtype(RoleForm::class.java, "FORM_ROLE")
5     .registerSubtype(PlayerTransferForm::class.java, "FORM_TRANSFER")
6     .registerSubtype(#Typ formulare#::class.java,
7     "#IDENTIFIKATOR_FORMULARE#")

```

Listing 4.16: Registrace typu do GsonParser

4.4.2 Potvrzení přes e-mail

Momentálně je při registraci účet vytvořen již aktivovaný, nevyžadujíc e-mailové potvrzení. Ke změně tohoto stavu je nutno v ovladači `/controllers/RegistrationController` nahradit momentální stav účtu za ten zakomentovaný. Tím se zařídí, že účty budou tvořeny ve stavu `PENDING` a nebude možnost se k nim přihlásit.

```
1 @PostMapping("/regular")
2 @ResponseBody
3 fun postAccountRegular(@RequestBody account: Account, response:
4     HttpServletResponse) : EmptyResponse {
5     account.state = AccountState.ACTIVE // AccountState.PENDING
6     accountService.registerAccount(account)
7     registrationService.addRegistration(account)
8
9     response.status = HttpServletResponse.SC_OK
10    return EmptyResponse()
11 }
```

Listing 4.17: Registrace

Dále je zde nutnost nastavit databázového `Daemona` tak, aby kontroloval nové vstupy do tabulky `Registration` a zasílal jim maily s vygenerovanými odkazy na potvrzení. Tyto odkazy budou obsahovat URI příkaz na funkci, která danému účtu změní stav z `PENDING` na `ACTIVE`. Podobně je řešená funkce [Deaktivace účtu]. Poté by měla být daná registrace z tabulky smazána.

```
1 @PutMapping("/loginAccount/deactivate")
2 @ResponseBody
3 fun deactivateMyAccount(authentication: Authentication, response:
4     HttpServletResponse) : EmptyResponse {
5     val accountId = accountService.getMyAccount(authentication.name)
6     .id
7     accountService.setState(accountId, AccountState.INACTIVE)
8     response.status = HttpServletResponse.SC_OK
9     return EmptyResponse()
10 }
```

Listing 4.18: Deaktivace účtu

`Daemon` by měl také kontrolovat časové stopy registrací, aby po uplynutí určitého časového intervalu (30 dní) záznam a s ním asociovaný neaktivní účet smazal a umožnil opětovnou registraci s daným e-mailem.

4.4.3 Externí registrace

Externí registrace by měla být řešena obdobně, jako registrace výše. Hlavní trenér klubu zadá přes mail registrační formuláře, které rodiče vyplní a zašlou. Následně zmíněný trenér exportuje data z formulářů do tabulky v `Microsoft Excel`, kterou projede předem připraveným `insert` skriptem. Tento skript zatím

nebyl vytvořen. Účelem tohoto skriptu je vytvoření účtu ve stavu `EXTERNAL_PENDING` spolu se záznamem v tabulce `Registration`, ve které bude generováno náhodné prvotní heslo.

Ze strany klienta je potřeba implementovat v aktivitě `LoginActivity` změnu prvotního hesla na nové, v místě s `//TODO` v [Přihlášení externího účtu].

```

1 AccountState.EXTERNAL_PENDING -> {
2     connection.accountRequests().getLoginAccountRoles({ roles : List
3         <Role>->
4         for (i in roles.indices) {
5             accountManager.setUserData(
6                 account,
7                 resources.getString(
8                     R.string.account_role,
9                     i
10                ),
11                roles[i].roleIdentifier(this)
12            )
13        }
14        setAccountAuthenticatorResult(this.intent.getExtras())
15        setResult(
16            AccountAuthenticatorActivity.RESULT_OK,
17            this.intent
18        )
19        //TODO - password change
20        finish()
21    })
22    val int = Intent(this, FirebaseTokenRegistrationListener::class.
23        java)
24    int.action = "cz.cvut.fit.sp.rugby.SEND_FIREBASE_TOKEN"
25    sendBroadcast(int)
26 }

```

Listing 4.19: Přihlášení externího účtu

K tomu lze využít již existující fragment `AccountPasswordChangeSimpleDialog`, který lze nalézt ve složce:

```

/fragments/accountFragments/accountSettingsFragments/
accountSettingsPasswordChange.

```

Tento fragment byl využit k nastavení hráčova hesla v rodičovském nastavení, stačí pouze vložit jeho interface do dané aktivity a implementovat jeho funkce.

Po úspěšné změně hesla by měl být na server zaslán příkaz, který změní heslo i stav účtu na `ACTIVE`, následováno úspěšným přihlášením uživatele. K implementaci tohoto příkazu je možno jednoduše využít již existující žádost [Update účtu], stačí jen před jejím zasláním změnit stav na `ACTIVE`.

```
1 override fun putMyAccount(  
2     account: Account,  
3     successCallback: () -> Unit,  
4     useProgressDialog: Boolean,  
5     failCallback: ((error: RequestError) -> Unit)?,  
6     connectionErrorCallback: (() -> Unit)?  
7 ) {  
8     requestHandler.putRequest(  
9         context.resources.getString(  
10            R.string  
11            .account_settings_update_account_request  
12        ),  
13        account.also {  
14            it.password?.let {  
15                pw -> it.password = PasswordHash.sha256Hash(pw)  
16            }  
17        },  
18        null,  
19        successCallback,  
20        useProgressDialog,  
21        failCallback,  
22        connectionErrorCallback  
23    )  
24 }
```

Listing 4.20: Update účtu

4.4.4 Nouzový kontakt

K využití nouzového kontaktu je nutno, aby rodič měl nastaven v účtu telefonní číslo. U tohoto požadavku je do budoucna zamýšleno předělat číslo jako povinné pole u externí registrace a znemožnění zažádání o roli rodič účtům bez čísla.

Funkcionalita je zamýšlena jako rychlý a viditelný způsob kontaktu rodiče pro trenéry v rámci tréninků a turnajů.

Tréninky

U tréninků lze přidat tlačítko pro tuto funkcionalitu v `layout_player_detail`. Zde je hodně místa na velké a viditelné tlačítko. Tato obrazovka je dostupná z `PlayerDetailActivity`, ke které má přístup trenér v rámci jeho tréninků. Více o těchto aktivitách se dozvíte v práci kolegy Martina Paula [10]. Pro naše účely stačí tuto aktivitu obohatit o novou funkci, kterou v příslušném fragmentu využijeme jako callback namapovaný na námi přidané tlačítko. V dané funkci bude spuštěn fragment (viz [Spuštění fragmentu]), který bude zprostředkovávat požadované funkcionality nouzového kontaktu. Přidání tlačítka vypadá v XML následovně:

```

1 <Button
2     android:id="@+id/#identifikator#"
3     android:layout_width="#sirka#"
4     android:layout_height="#vyska#"
5     android:text="#text#"
6     android:textAlignment="#zarovnani#"
7     android:textColor="#barva textu#"
8 />

```

Listing 4.21: XML tlačítko

XML layouts lze nalézt v `/res/layout`.

Turnaje

Správa hráčů v rámci turnajů je řešena skrze týmy, bylo by proto vhodné zavést seznam hráčů v rámci daného týmu, ze kterého by vedla cesta k detailu hráče. K tomuto by mohla být upravena aktivita kolegy viz *Tréninky* výše.

Samotná funkcionalita

Do nově vytvořeného fragmentu vložíme skrze argumenty JSON zprávu, která obsahuje vytažené telefonní číslo zástupce. Níže jsou uvedeny příklady zaslání textového řetězce z aktivity do nově inicializovaného fragmentu a jeho následná extrakce z JSON uvnitř téhož fragmentu.

```

1 fun send(){
2     val fragment = #Typ fragmentu#()
3     val args = Bundle()
4     args.putString(
5         getString(R.string.MESSAGE_DIALOG__text_argument),
6         GsonParser.toJson(#text#)
7     )
8     fragment.arguments = args
9
10    singleFragmentFunc.pushFragment(fragment)
11 }

```

Listing 4.22: Zaslání textu

```

1 fun getText(bundle: Bundle){
2     val textJson =
3         bundle.getString(parentActivity.getString(R.string.
4 MESSAGE_DIALOG__text_argument))
5     if (textJson != null) {
6         text = GsonParser.fromJson(textJson)
7     }
8 }

```

Listing 4.23: Přijetí textu

Tyto funkce nám jsou užitečné, jelikož telefonní číslo je uloženo v textovém řetězci. Následující příkazy slouží k nastartování aktivity, která přesměruje

4. IMPLEMENTACE

uživatele z aplikace na hovor, nebo k zaslání SMS s danou zprávou. Zpráva SMS by měla obsahovat kontaktní informace na trenéra, případně klub.

```
1 Intent callIntent = new Intent(Intent.ACTION_CALL);
2 callIntent.setData(Uri.parse("tel:#cislo#"));
```

Listing 4.24: Volání

```
1 Intent sendIntent = new Intent(Intent.ACTION_VIEW);
2 sendIntent.putExtra("sms_message", "#obsah zpravy#");
3 sendIntent.setType("vnd.android-dir/mms-sms");
4 startActivity(sendIntent);
```

Listing 4.25: Zaslání SMS

Nutno doplnit, že pro funkčnost této části je nutno do hlavního manifestu (souboru `AndroidManifest.xml`) přidat tato povolení:

```
1 <uses-permission android:name="android.permission.CALL_PHONE" />
2 <uses-permission android:name="android.permission.SEND_SMS" />
```

Listing 4.26: Přijetí textu

Testování

Téma této kapitoly je popis a provedení testů na implementované části prototypu modulu.

Jednotkové testy

K zajištění základní funkcionality projektu byly ke třídám a funkcím vytvořeny jednotkové testy tam, kde mělo jejich zavedení smysl. K jejich tvorbě a aplikaci byl využit JUnit framework, který je součástí IntelliJ i Android Studio.

Zaměření těchto testů bylo převážně na funkce využitých při rozhodování typů formulářů a anonymizace účtů.

Integrační testy

Tyto testy se soustředí na ověření funkčnosti větších celku programu. Jejich implementace je umožněna tzv. „Mockery“ z knihovny Mockito, které napodobují provoz daných celků aplikace. Při testování částí uživatelského rozhraní klienta byl také použit testovací framework Espresso.

Uživatelské testy

Dne 6.5.2021 proběhly na katedře softwarového inženýrství uživatelské testy. Testování proběhlo na třech lidech, každý jednotlivě a na stejném zařízení. Jelikož dané zařízení nebylo na testy předem připraveno, z důvodu selhání v získání adekvátního testovacího nástroje, šlo zde nalézt chyby pravděpodobně způsobené nastaveními původního uživatele. Tyto chyby se týkaly zejména příjmu notifikací v testech kolegy Matěje Ulmana a neměly efekt na testy tohoto modulu. Testy proběhli dle scénářů (viz Přílohy – Scénář uživatelských testů a práce kolegů [10], [11]). Záznam testování byl pro účely práce pořízen, ale z důvodu ochrany osobních údajů nebude zveřejněn.

5. TESTOVÁNÍ

Testy byly moderovány každým autorem scénářů zvláště, s důrazem na minimalizaci komunikace s testovaným k ověření intuitivnosti funkcionalit a uživatelského rozhraní. Samozřejmě, každé z osob byl předem krátce vysvětlen kontext a účel aplikace.

Průběhy a výsledky testů týkajících se tohoto modulu jsou vypsány níže.

Testovaný 1

Až na drobné zmatky ve výmazu hráčů a rozhraní formulářů proběhl test bez problémů. Testovaný správně reagoval na úkony dané scénářem, aplikace fungovala konzistentně a bez znatelných chyb.

Testovaný uvedl, že aplikace je velice pěkná a snadno se mu v ní naviguje.

Testovaný 2

Tento test proběhl též bez problémů, osoba správně reagovala a neměla větší potíže v navigaci. Aplikace též fungovala správně.

Testovaný 3

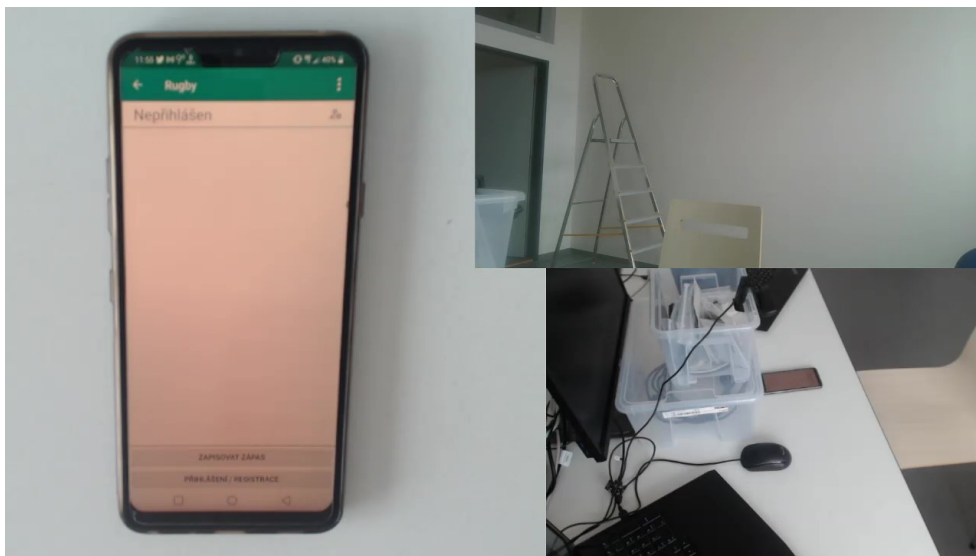
I zde byl průběh bez problému, avšak testovaný měl validní připomínku. U data narození v nastavení účtu a registraci hráče je vložitelné datum omezeno pouze momentálním datem. Specifičtější omezení data narození by měla být do budoucna předmětem průzkumu u žadatelů. I přes to testovaný uvedl že se mu zpracování aplikace líbilo.

Poznámka

Ve scénáři bylo chybně zadáno, že pokus o přihlášení smazaného účtu bude oznámen textem „Účet není aktivní“. Zde se jedná o omyl autora scénáře, text měl znít: „Přihlášení se nepodařilo“.

Závěr uživatelských testů

Testy tohoto modulu úspěšně proběhly s menšími výhradami, které budou brány v potaz v následujícím vývoji aplikace.



Obrázek 5.1: Záběr z testování

Akceptační test

Akceptační test byl proveden Ing. Jiřím Chludilem a bylo v něm předvedeno a hodnoceno splnění požadavků, obecná funkčnost aplikace a jejího uživatelského rozhraní. Níže budou popsány největší problémy, jež se jeho průběhu naskytly.

Ověření platnosti e-mailu

Kontrola platnosti e-mailu je momentálně řešena porovnáním s jednoduchým regulárním výrazem. Tento způsob může působit problémy, proto je nutné ho do budoucna zaměnit za zaslání zprávy na daný mail. V případě, že zpráva dorazí, tak daný mail existuje a je vhodný k použití při registraci.

Metrika síly hesla

V implementaci je dostatečnost hesla řešena zastaralým způsobem, kde je pouze kontrolována délka osmi a více znaků, z nichž nejméně dvě musí být čísla. Zmíněný způsob by měl být v následujících iteracích aplikace změněn na některý z algoritmů pro měření síly hesla.

Tlačítka u formulářů

Nezaškrtnuté checkboxy u formulářů vypadají identicky a nemají viditelné rozlišení mezi potvrzením a zamítnutím. Tento problém měl triviální řešení a

5. TESTOVÁNÍ

byl vyřešen obarvením ohraničení prázdných checkboxů odpovídajícími barvami.

Chyba přihlášení nově vytvořených hráčů

Za hráče registrované rodičem nebylo možno se přihlásit. Tento problém byl způsoben chybějící implementací hashe hesla při jeho zasílání u tvorby hráče. Problém měl triviální řešení a byl opraven.

Chybná kontrola validity e-mailu při změně kontroly

Když rodič zbaví svého hráče v nastavení rodičovské kontroly, mělo by dojít ke kontrole validity e-mailu. V testu k této kontrole nedošlo, řešení tohoto problému vyžaduje bližší prozkoumání podmínek kontroly.

Chybějící autorizace při výmazu účtu

Před trvalým výmazem (anonymizací) účtu chybí složitější potvrzení. Toto otevírá možnosti např. překliknutí, nebo výmazu účtu při odcizení zařízení. Do budoucna by proto měla být u možnosti trvalého výmazu vyžadována autorizace heslem.

Nadbytečné kategorie v seznamu hráčů

Ve správě hráčů u rodiče jsou automaticky vypsány všechny kategorie. Do budoucna by měl daný fragment být upraven tak, aby zobrazoval pouze ty, ve kterých má rodič přiděleného hráče.

Budoucí funkcionality

Níže jsou uvedeny funkcionality, které jsou očekávány jako budoucí rozšíření této aplikace.

Rozšíření formulářů

- Změna detailu účtu (Hráč -> Rodič)
- Přihlášení a odhlášení hráče z týmu/tréninku
- Žádost administrátorovi o změnu e-mailu / data narození

Rozšíření registrací

- Potvrzení registrace e-mailem
- Dokončení externí registrace
- Algoritmus pro ohodnocení bezpečnosti hesla
- Ověření platnosti e-mailu zasláním zprávy

Rozšíření správy účtů

- Ověření heslem před výmazem účtu
- Možnosti obnovy dočasně zrušeného účtu

Rozšíření zákonného zástupce

- Nouzový kontakt

6. BUDOUCÍ FUNKCIONALITY

- Změny menu správy hráčů

Závěr

Práce se zabývala tvorbou modulu pro jádro mobilní aplikace, která má sloužit k organizaci sportovních klubů, tréninků a turnajů zaměřených na dětské hráče.

V analýze byl popsán momentální stav jádra aplikace a technologií využitých k jeho vývoji. Dále byla vyhodnocena konkurenční řešení s podobnými tématy, v čemž byly porovnány jejich nedostatky oproti požadavkům od žadatelů. Poté byly analyzovány podobné odborné práce a došlo k detailnějšímu upřesnění rolí uživatelů. Z poznatků získaných během analýzy byly odvozeny požadavky, které by modul měl obsahovat a nakonec byla popsána komunikace se žadatelem a zhodnocena compliance s GDPR.

Na základě předešlé analýzy byly navrženy případy užití daných funkcionalit, jejich aktivity, třídy a uživatelské rozhraní. Tento návrh byl následně implementován a podroben integračním, akceptačním a uživatelským testům. Došlo také k sepsání detailní dokumentace k možnostem užití a rozšíření této implementace. Z výsledných informací byly poté určeny možné rozšiřující funkcionality.

Z pohledu autora došlo ke splnění všech cílů odvozených ze zadání práce.

Literatura

- [1] NVIDIA Corporation: Fixing Common Android Lifecycle Issues in Games. leden 2021, [online]. [obrázek převzat 2021-04-17]. Dostupné z: <https://developer.nvidia.com/fixing-common-android-lifecycle-issues-games>
- [2] Banff Minor Hockey Association: TeamSnap. září 2020, [online]. [obrázek převzat 2021-03-08]. Dostupné z: <https://www.banffminorhockey.com/teamsnap/>
- [3] TeamSnap: Show or Hide Teams in the Teams List. prosinec 2020, [online]. [obrázek převzat 2021-04-17]. Dostupné z: <https://helpme.teamsnap.com/article/225-show-or-hide-teams-in-the-teams-list>
- [4] Turistika.cz s.r.o: Spoluhráči.cz. květen 2021, [online]. [cit. 2021-04-07]. Dostupné z: <https://www.spoluhraci.cz/>
- [5] APKPure.com: TeamStuff. duben 2021, [online]. [cit. 2021-04-08]. Dostupné z: <https://www.apkpure.com/teamstuff/com.beyondsoftware.teamstuff>
- [6] AppAdvice LLC.: Týmuj. listopad 2020, [online]. [obrázek převzat 2021-04-04]. Dostupné z: <https://www.appadvice.com/app/t-c3-bdmuj/1218860308>
- [7] Team Mates: Create A Free App For Your Team. únor 2021, [online]. [cit. 2021-03-08]. Dostupné z: <https://www.teamapp.com/>
- [8] VMI Silesia s.r.o.: Co je Sportnect? prosinec 2020, [online]. [cit. 2021-03-11]. Dostupné z: <https://www.sportnect.com/cs/news/co-je-sportnect>

- [9] Broun, H. H.: I like to see things done with zest. *Ames Daily Tribune*, leden 1974.
- [10] Paul, M.: *Aplikace rugby – modul tréninky*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.
- [11] Ulman, M.: *Aplikace rugby - modul organizace turnajů a zápasů*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.
- [12] Špetl, J.: *Aplikace rugby – modul výzvy*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2021, (práce odevzdána později).
- [13] Android Developers: Meet Android Studio. únor 2021, [online]. [cit. 2021-03-08]. Dostupné z: <https://developer.android.com/studio/intro>
- [14] JetBrains: IntelliJ IDEA overview. únor 2021, [online]. [cit. 2021-03-09]. Dostupné z: <https://www.jetbrains.com/help/idea/discover-intellij-idea.html>
- [15] VMware, Inc.: Spring Framework. únor 2021, [online]. [cit. 2021-03-09]. Dostupné z: <https://spring.io/projects/spring-framework>
- [16] Hibernate: Spring Framework. květen 2021, [online]. [cit. 2021-03-09]. Dostupné z: <https://hibernate.org/orm/>
- [17] H2 Database Engine. únor 2021, [online]. [cit. 2021-03-09]. Dostupné z: <https://www.h2database.com/html/main.html>
- [18] The PostgreSQL Global Development Group: What is PostgreSQL? únor 2021, [online]. [cit. 2021-03-09]. Dostupné z: <https://www.postgresql.org/about/>
- [19] Heller, M.: What is Kotlin? The Java alternative explained. *InfoWorld*, květen 2020, [online]. [cit. 2021-03-09]. Dostupné z: <https://www.infoworld.com/article/3224868/what-is-kotlin-the-java-alternative-explained.html>
- [20] Android Developers: Espresso. březen 2021, [online]. [cit. 2021-03-09]. Dostupné z: <https://developer.android.com/training/testing/espresso>
- [21] ECMA-404 The JSON Data Interchange Standard. březen 2021, [online]. [cit. 2021-03-09]. Dostupné z: <https://www.json.org/json-en.html>
- [22] Red Hat, Inc.: What is a REST API? únor 2021, [online]. [cit. 2021-03-09]. Dostupné z: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>

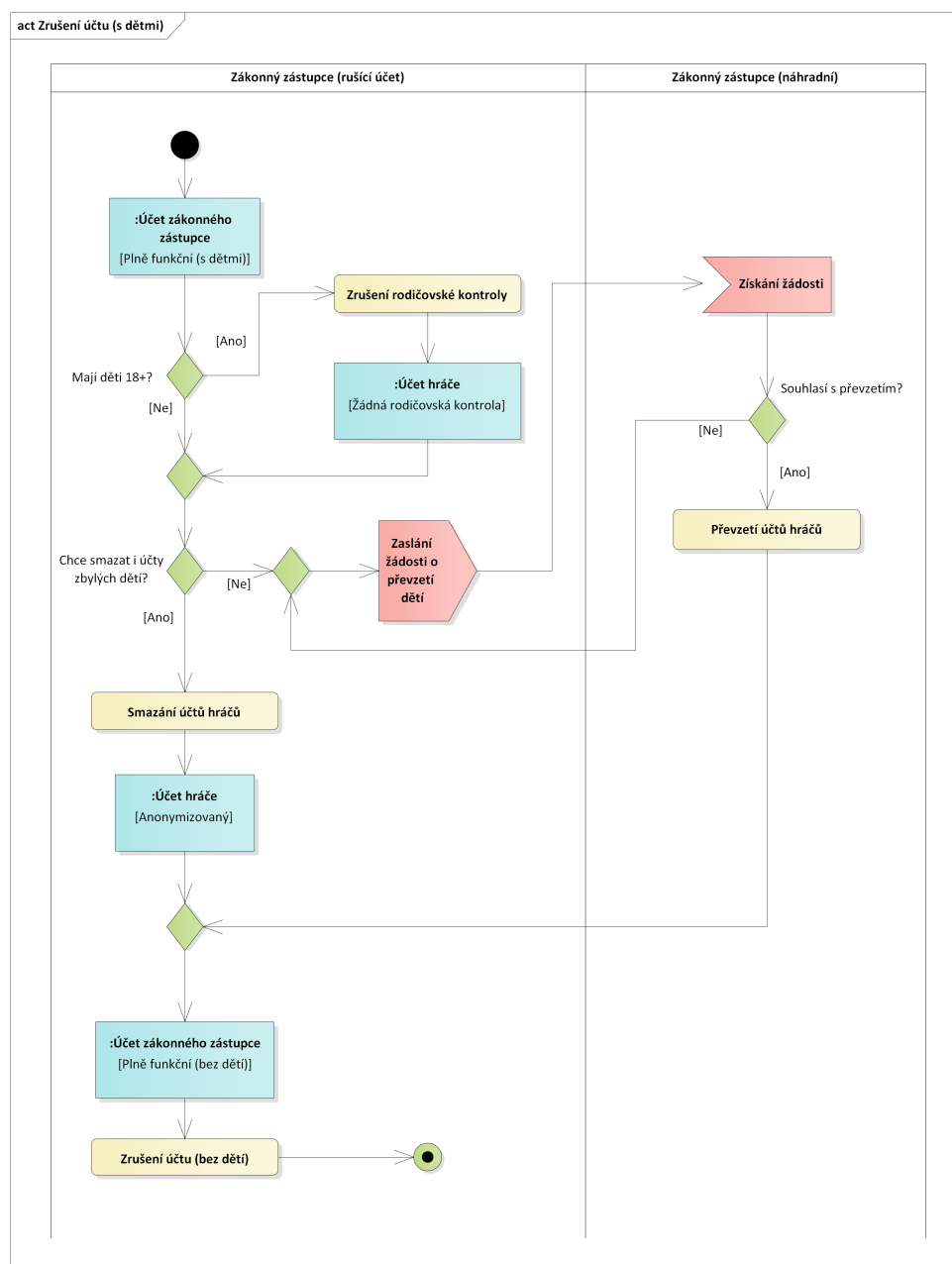
-
- [23] GitLab Inc.: About GitLab. březem 2021, [online]. [cit. 2021-03-12]. Dostupné z: <https://about.gitlab.com/company/>
- [24] Gradle Inc.: Gradle Features. leden 2021, [online]. [cit. 2021-03-12]. Dostupné z: <https://gradle.org/features/>
- [25] The Apache Software Foundation: What is Maven? květen 2021, [online]. [cit. 2021-03-12]. Dostupné z: <https://maven.apache.org/what-is-maven.html>
- [26] Android Developers: Introduction to Activities. únor 2021, [online]. [cit. 2021-03-12]. Dostupné z: <https://developer.android.com/guide/components/activities/intro-activities>
- [27] Android Developers: Fragments. únor 2021, [online]. [cit. 2021-03-12]. Dostupné z: <https://developer.android.com/guide/fragments>
- [28] TeamSnap: TeamSnap. únor 2021, [online]. [cit. 2021-03-08]. Dostupné z: <https://www.teamsnap.com/>
- [29] Teamstuff.com: TeamStuff. leden 2021, [online]. [cit. 2021-03-11]. Dostupné z: <https://blog.teamstuff.com/>
- [30] ByAllMeans: Organizuj sportovní tým online V mobilu i na webu. Zdarma. květen 2021, [online]. [cit. 2021-04-07]. Dostupné z: <https://www.tymuj.cz/>
- [31] Steklík, Š.: *FurryBall - správa licencí a uživatelů*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, červen 2015. Dostupné z: <https://dspace.cvut.cz/handle/10467/63195>
- [32] Vlčková, M.: *BigCloud - modul obchodníka*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, únor 2017. Dostupné z: <https://dspace.cvut.cz/handle/10467/69126>
- [33] Kovařovic, K.: *Výuková aplikace Dráček II - gamifikace a personalizace*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, červen 2016. Dostupné z: <https://dspace.cvut.cz/handle/10467/66216>
- [34] Evropský parlament a rada Evropské unie: NAŘÍZENÍ EVROPSKÉHO PARLAMENTU A RADY (EU) 2016/679 ze dne 27. dubna 2016 o ochraně fyzických osob v souvislosti se zpracováním osobních údajů a o volném pohybu těchto údajů a o zrušení směrnice 95/46/ES (obecné nařízení o ochraně osobních údajů). duben 2016, [online]. Dostupné z: <https://eur-lex.europa.eu/legal-content/CS/TXT/?uri=CELEX:32016R0679>

LITERATURA

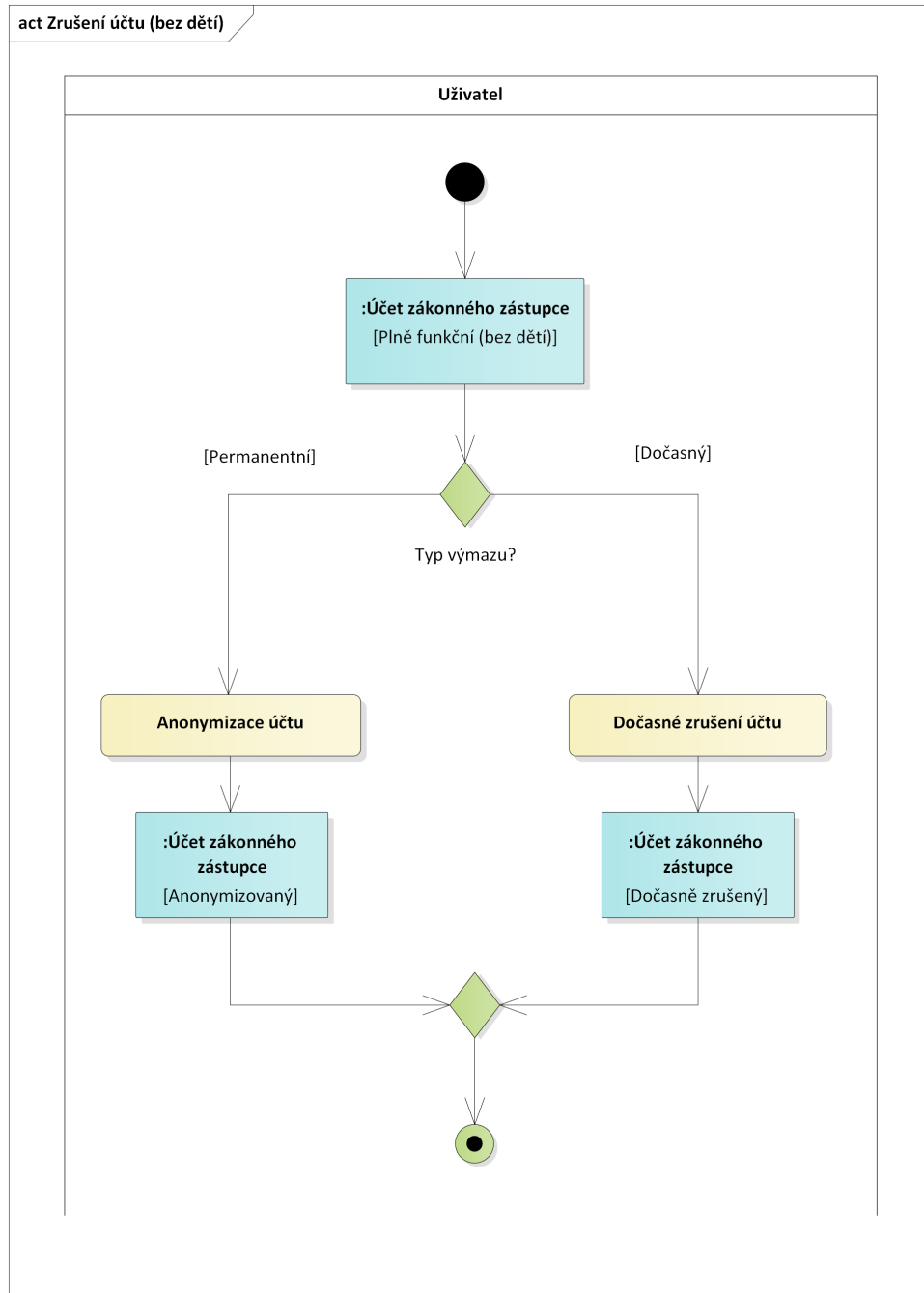
- [35] Android Developers: Run apps on a hardware device. duben 2021, [online]. [cit. 2021-04-28]. Dostupné z: <https://developer.android.com/studio/run/device#wireless>

Diagramy

A.1 Zrušení účtu



Obrázek A.1: Diagram aktivit – zrušení s dětmi



Obrázek A.2: Diagram aktivit – zrušení bez dětí

Scénář uživatelských testů

Modul zákonného zástupce - testovací scénář

Test registrací, přihlášení, formulářů, nastavení účtů, tvorby hráčů a mazání účtů

1. Uživatel spustí aplikaci a zaregistruje se s následujícími kredenciály:
 - E-mail: test@fit.cz
 - Heslo: password66
 - Klub: Újezd
2. Po úspěšné registraci se daným účtem přihlásí.
3. Po přihlášení přejde do nastavení účtu, kde si nastaví libovolné jméno, příjmení a datum narození.
4. Následně zašle formulář, ve kterém zažádá uživatele *Jindřich Kopanec* o přiřazení role Rodič s libovolnou zprávou. *Uživatel Jindřich Kopanec [hlavni_trener, password] (druhá osoba) přejde do inboxu formulářů a shlédne, zda zmíněný formulář dorazil. Pokud ano, zkontroluje zda odpovídá typ formuláře, jeho popis, jméno odesílatele a zpráva. Poté formulář přijme.*
5. Původní uživatel si zkontroluje inbox (může využít obnovení stránky), zda obdržel potvrzení o přiřazení role. V případě že ano, odhlásí se a znovu se přihlásí.
6. Po přihlášení by měl vidět v menu rozkliknutelný seznam s názvem Rodič. V tom případě přejde na správu hráčů a vytvoří nového hráče s libovolnými kredenciály.
7. Byl-li hráč vytvořen, shlédne jeho detaily ve správě hráčů.
8. Nakonec uživatel přejde do nastavení svého účtu a smaže si svůj účet. Měl by narazit na okno, které ho informuje o přiřazených hráčích, ve kterém stiskne "Potvrdit". Následně se opět pokusí o výmaz účtu, tentokrát se objeví jiné okno. Uživatel vybere tlačítka "Vymazat trvale" a "Potvrdit".
9. Uživatel by měl být automaticky odhlášen. Může zkusit, zda se lze na účet přihlásit, při čemž by se měla objevit chybová hláška "Účet není aktivní".

Konec testu

Dotazníky

Rugby aplikace - Rodič

V rámci bakalářských prací na Fakultě informačních technologií ČVUT vyvíjíme aplikaci pro rugbyové kluby, která by měla usnadnit chod klubu. Aplikace bude podporovat například správu klubu, organizaci turnajů a tréninků či sdílení fotek z akcí. Cílem tohoto dotazníku je získat od Vás takové informace, abychom mohli vývoj aplikace přizpůsobit Vaším požadavkům, požadavkům budoucích uživatelů.

V tomto dotazníku bychom rádi zjistili Váš názor na náš návrh, jaké funkcionality by měla podporovat rodičovská sekce aplikace, tedy část aplikace určená pro rodiče.

K většině otázek bychom rádi získali podrobnější vyjádření, které můžete napsat do textového pole přímo pod otázkou.

Tato zpětná vazba je pro nás velmi důležitá, abychom mohli aplikaci přizpůsobit Vaším požadavkům. Proto předem děkujeme za Váš čas, který tomuto dotazníku věnujete. Věříme, že Vám konečná aplikace pomůže a ulehčí komunikaci s klubem.

Dotazník je zcela anonymní. Data budou využita pouze pro vývoj této aplikace a nebudou nikde veřejně sdílena.

* Required

Základní funkcionality

Tato část se soustředí na registraci a správu účtů

Registrace

Registrovat se uživatel může buďto přes formulář mimo aplikaci za pomoci hlavního trenéra klubu, který následně vytvoří daný účet a zpět pošle e-mailem náhodně vygenerované heslo k prvnímu přihlášení, nebo si uživatel může v aplikaci sám účet vytvořit a poslat v ní žádost o přidělení pravomocí zákonného zástupce/rodiče.

1. Které z těchto možností registrace byste v aplikaci uvítali? *

Mark only one oval.

Přes e-mail

Přes aplikaci

Oboje

Other: _____

2. Pokud máte k registraci nějaké další poznámky, náměty či připomínky, nebo byste měli zájem o nějakou další notifikaci, prosím, napište je níže:

Účet hráče

Jakožto rodič/zákonný zástupce budete schopni v aplikaci vytvořit a spravovat účet vašeho dítěte/svěřence. Pod správu účtu spadá např. změna hesla, přihlašování na turnaje, tréninky, atp.

3. Využili byste možnost vytvoření účtu hráče? *

Mark only one oval.

Ano

Ne

4. Pokud máte k účtu hráče nějaké další poznámky, náměty či připomínky, nebo byste měli zájem o nějakou další notifikaci, prosím, napište je níže:

Pravomoc nad účtem hráče

Rodič/zákonný zástupce bude mít možnost určit, jakou moc má nad účtem svého dítěte/svěřence. Na výběr bude mít při nastavování účtů 3 možnosti: úplná kontrola, částečná kontrola a žádná kontrola (pouze hráči 18+)

5. Využili byste tyto možnosti kontroly? *

Mark only one oval.

Ano

Ne

6. Pokud máte k pravomocem nad účty hráčů nějaké další poznámky, náměty či připomínky, nebo byste měli zájem o nějakou další notifikaci, prosím, napište je níže:

Specializované formuláře

Jakožto uživatel budete moci přímo v aplikaci zasílat formuláře správcům a trenérům k záležitostem jako: změna role, přihlášení hráče na turnaje a obdobné akce.

7. Měli byste zájem o zavedení těchto formulářů v rámci aplikace? *

Mark only one oval.

Ano

Ne

8. Pokud máte k formulářům nějaké další poznámky, náměty či připomínky, nebo byste měli zájem o nějakou další notifikaci, prosím, napište je níže:

SMS varování

Pokud bude vaše dítě zraněno nebo ztraceno během klubové akce, bude odpovědný trenér mít možnost vám skrze aplikaci zaslat na vámi předem zadané číslo SMS o stavu vašeho dítěte a potřebné kontaktní informace.

9. Měli byste zájem o zavedení SMS varování v rámci aplikace?

Mark only one oval.

Ano

Ne

10. Pokud máte k SMS varování nějaké další poznámky, náměty či připomínky, nebo byste měli zájem o nějakou další notifikaci, prosím, napište je níže:

Tréninky

Tato část dotazníku se týká tréninků dítěte a dalších věcí spojených s tréninky. Trenér bude v aplikaci vytvářet časový plán tréninků, které uvidí rodič/hráč.

Potvrzování účasti na tréninku

Rodič, respektive dítě bude mít možnost prostřednictvím aplikace potvrdit svou účast na tréninku, případně se omluvit. Následně budou mít trenéři možnost s očekávaným počtem dětí na tréninku pracovat (například počet trenérů potřebných na trénink).

11. Využili byste možnost potvrzování účasti dítěte na tréninku? *

Mark only one oval.

Ano

Ne

Other: _____

12. Pokud máte k potvrzení účasti na tréninku nějaké další poznámky, náměty či připomínky, prosím, napište je níže:

Oznámení o pozdním příchodu

Pokud dítě nebude stíhat trénink, bude mít možnost (hráč nebo rodič) prostřednictvím aplikace informovat trenéry, že dorazí se zpožděním.

Stejně tak se může stát, že i trenér dorazí později na trénink. Ti budou mít taktéž možnost informovat hráče a rodiče o svém pozdním příchodu.

13. Využili byste možnost oznámení trenérům o pozdním příchodu hráče? *

Mark only one oval.

Ano

Ne

Other: _____

14. Přejde Vám funkcionalita oznamování pozdního příchodu trenéra užitečná? *

Mark only one oval.

Ano

Ne

Other: _____

15. Pokud máte k oznámení o pozdním příchodu nějaké další poznámky, náměty či připomínky, prosím, napište je níže:

Potvrzování příchodu hráče na trénink

Je zde možnost, že by aplikace umožňovala automaticky potvrzovat příchod hráče na trénink pomocí polohy telefonu. Rodič by měl následně možnost například dostávat notifikace o příchodu hráče na trénink.

Tato funkce se však může zdát dosti kontroverzní, proto je zde pro nás Váš názor velmi důležitý.

16. Měli byste zájem o funkcionalitu potvrzování příchodu hráče na trénink? *

Mark only one oval.

Ano

Ne

17. Pokud máte k potvrzování příchodu hráče na trénink nějaké další poznámky, náměty či připomínky, prosím, napište je níže:

Notifikace

Aplikace bude umožňovat zasílat o různých událostech notifikace (upozornění v mobilním telefonu, které formou krátké zprávy oznamuje nějakou aktualitu). Níže prosím vyberte, o které notifikace byste měli zájem ve spojitosti s tréninkem.

18. O které notifikace byste měli zájem?

Check all that apply.

- Informace o nově přidaném tréninku
- Informace o zrušeném/změněném tréninku
- Informace o konci tréninku
- Příchod dítěte na trénink

19. Pokud máte k notifikacím nějaké další poznámky, náměty či připomínky, nebo byste měli zájem o nějakou další notifikaci, prosím, napište je níže:

Vizualizace tréninků

Rodič i hráč bude mít možnost vidět přehled všech tréninků, včetně jeho docházky. Aplikace zobrazí seznam tréninků, včetně různých informací, jako například kde se trénink koná.

V aplikaci bude také možnost exportovat kalendář tréninků do elektronických kalendářů, jako je například aplikace Google Calendar. Zde by probíhala automatická synchronizace s daty v aplikaci.

20. Chtěli byste vidět přehled tréninků v aplikaci? *

Mark only one oval.

- Ano
- Ne
- Other: _____

21. Měli byste zájem o možnost exportu kalendáře do aplikace Google Calendar? *

Mark only one oval.

Ano

Ne

22. Pokud máte k vizualizaci tréninků a exportu kalendáře nějaké další poznámky, náměty či připomínky, prosím, napište je níže:

23. Máte ještě nějaké tipy či připomínky k sekci tréninků? Podělte se o ně s námi, prosím:

Výzvy

Výzvy jsou okruh obsahující otázky, které se týkají motivace dětí a zpracování jejich výsledků ve výzvách.

Mezi výzvy patří např. plnění úkolů zadaných trenérem, dosahování milníků a pravidelné nabídky od aplikace k plnění úkolů.

24. Líbí se vám možnost, že byste potvrzovali, zda dítě úkol opravdu splnilo?

Mark only one oval.

Ano

Ne

25. Líbí se vám nápad, že by děti měly na splnění výzvy den nebo týden?

Mark only one oval.

Ano

Ne

26. Chcete, aby aplikace zaznamenávala výsledky dětí na základě splněných výzev a aktivitě na trénincích?

Mark only one oval.

Ano

Ne

27. Chcete, aby aplikace na základě vložených dat trenérem vyhodnotila aktivitu hráče, a na základě aktivity vytvořila grafy, na kterých bude snadno vidět vývoj hráče v čase?

Mark only one oval.

Ano

Ne

28. Chcete, aby aplikace motivovala hráče pravidelnými výzvami ke cvičení / zlepšování se?

Mark only one oval.

Ano

Ne

29. Chcete, aby aplikace zaznamenávala řadu, tj. kolik dní v kuse dítě plnilo výzvy?

Mark only one oval.

Ano

Ne

30. Chcete, pokud potvrdíte, že děti plní výzvy spolu (např. ve dvojicích) a ne po jednom, aby byla spolupráce odměněna?

Mark only one oval.

Ano

Ne

31. Chcete, aby za výzvy a aktivitu na trénincích bylo možné od trenéra získávat body, které by sloužily k zakoupení různých vylepšení v aplikaci a k obecnému bodovacímu systému, díky kterému je aplikace bude schopna vytvářet žebříčky hráčů?

Mark only one oval.

Ano

Ne

32. Líbí se vám možnost, že by každý uživatel měl možnost vytvořit si svého avatara, který by fungoval jen jako vizuální reprezentace dané osoby (např. Maruška – bohyně hněvu, Fanda – zabiják medvědů)?

Mark only one oval.

Ano

Ne

33. Chcete, aby si své avatary mohli rodiče a trenéři vylepšit dle svého uvážení, zatímco hráči si změny museli zasloužit a mohli si je pak koupit za body získané za aktivitu?

Mark only one oval.

Ano

Ne

34. Chcete, aby trenéři, rodiče i děti měli možnost vytvořit si svůj profil, přidat svou fotografii a něco o sobě napsat do svého popisku?

Mark only one oval.

Ano

Ne

35. Chcete, aby hráč měl přístup ke zdi milníků, které se hráči odemykají na základě výsledků ve výzvěch (např. gym destroyer -> dítě překonalo milník 200 kliků)?

Mark only one oval.

Ano

Ne

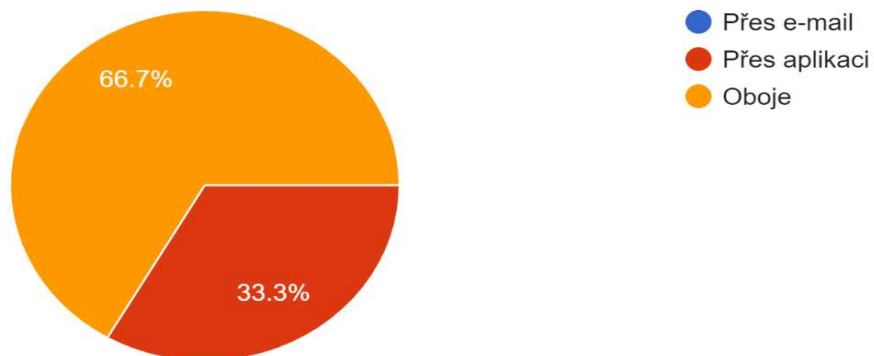
36. Máte k nějaké z otázek výhrady či připomínky? Budeme rádi za jakoukoli zpětnou vazbu.

This content is neither created nor endorsed by Google.

Google Forms

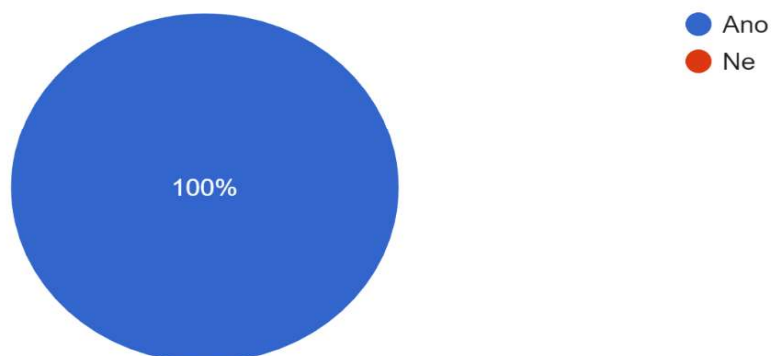
Které z těchto možností registrace byste v aplikaci uvítali?

6 responses



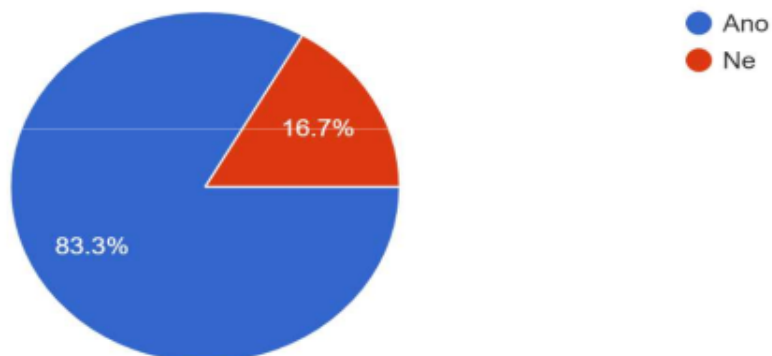
Využili byste možnost vytvoření účtu hráče?

6 responses



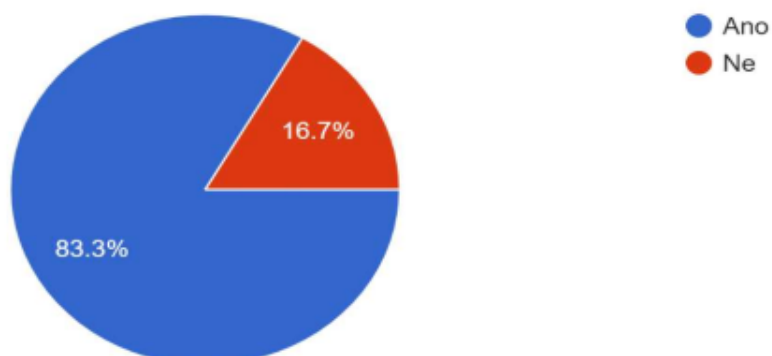
Využili byste tyto možnosti kontroly?

6 responses



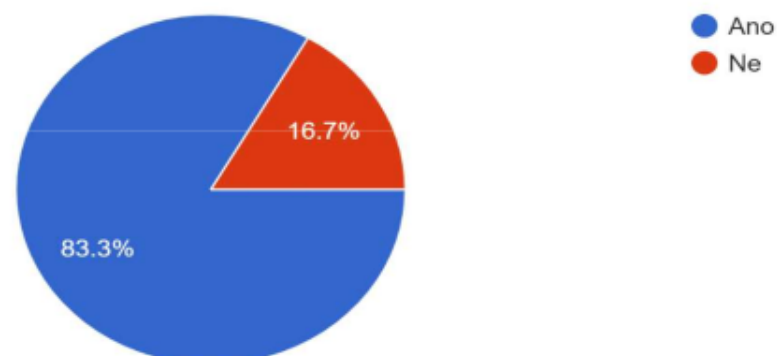
Měli byste zájem o zavedení těchto formulářů v rámci aplikace?

6 responses



Měli byste zájem o zavedení SMS varování v rámci aplikace?

6 responses



Seznam použitých zkratk

- AES** Advanced Encryption Standard
- API** Application Programming Interface
- APK** Android Application Package
- AVD** Android Virtual Device
- CD** Continuous Deployment
- CI** Continuous Integration
- CRUD** Create, Read, Update, Delete
- DSL** Domain-Specific Language
- EE** Enterprise Edition
- FP** Functional Programming
- GDPR** General Data Protection Regulation
- GSSAPI** Generic Security Services Application Program Interface
- GUI** Graphical User Interface
- HTTP** Hypertext Transfer Protocol
- IoC** Inversion of Control
- JDBC** Java Database Connectivity
- JDK** Java Development Kit
- JPA** Java Persistence API

D. SEZNAM POUŽITÝCH ZKRATEK

JVM Java Virtual Machine

MD5 Message-Digest Algorithm

ODBC Open Database Connectivity

OOP Object-oriented programming

ORM Object-Relational Mapping

REST Representational State Transfer

RC Rugby Club

SCRAM Salted Challenge Response Authentication Mechanism

SE Standard Edition

SHA Secure Hash Algorithm

SQL Structured Query Language

SSPI Security Support Provider Interface

UI User Interface

URI Uniform Resource Identifier

XML Extensible Markup Language

ZZ Zákonný Zástupce

Obsah přiloženého média

karloda_BP	
├── BP.pdftext práce ve formátu pdf
├── readme.txtstručný popis obsahu média
├── srcsoubor se zdrojovými kódy
│ ├── latexzdrojový kód textu práce ve formátu L ^A T _E X
│ ├── serverzdrojový kód lokální serverové části
│ └── clientzdrojový kód mobilního klienta
└── APKclient.apkAPK soubor k instalaci mobilního klienta