

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Interaktivní aplikace pro dětské tábory
Bakalářská práce

Autor: David Kozel
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Daniel Vondra
Pracoviště: Katedra informatiky a kvantitativních metod

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 20. 4. 2024

.....
David Kozel

Poděkování:

Chtěl bych především poděkovat panu Ing. Danielu Vondrovi za cenné rady, připomínky a vstřícnost při konzultacích a za pomoc a rady při zpracování této práce. Také děkuji svým přátelům, rodině a akademickému sboru, kteří mě při zpracování mé bakalářské práce velmi podporovali.

Anotace

Cílem bakalářské práce byl vývoj nové edukativní mobilní aplikace pro využití v oblastech neformálního vzdělávání na dětských táborech. Aplikace seznamuje mladší generace s novými technologiemi a funkcionalitami, které mohou organizátoři a účastníci dětských táborů využívat na dětských táborech. Mobilní aplikace byla vytvořena pomocí programovacího jazyka Java s využitím na mobilních telefonech s operačním systémem Android, grafický vzhled byl vyřešen za pomoci značkovacího jazyka XML. Mobilní aplikace využívá různých knihoven, na základě kterých byly integrovány funkcionality kompas, QR, RFID čtečka, mapy a Morseův kód.

Přínosy práce nalezneme v praktickém využití na dětských táborech s možností dalšího rozšíření v rámci existujících sportovních spolků. V neposlední řadě aplikace nabízí lehkou správu dat v databázi za použití různých typů formulářů.

Annotation

Title: Educational application for children's camps

The aim of the bachelor's thesis was the development of a new educational mobile application for use in the areas of informal education at children's camps. The application introduces younger generations to innovative technologies and functionalities that organizers and participants of children's camps can utilize. The mobile application was created using the Java programming language for mobile phones with the Android operating system, and its graphical interface was designed using the XML markup language. The application employs various libraries to integrate functionalities such as a compass, QR, RFID reader, maps, and Morse code.

The benefits of the work can be found in its practical application at children's camps, with the possibility of further expansion within existing sports associations. Lastly, the application offers easy data management in databases using various forms.

Obsah

1	Úvod	1
2	Cíl a metodika práce.....	3
2.1	Cíle práce	3
2.2	Metodologický postup	4
3	Stávající softwarová řešení a jejich význam	6
3.1	Význam mobilních a počítačových aplikací v kontextu dětských táborů.....	6
3.2	Edukace prostřednictvím mobilních a počítačových aplikací na dětských táborech.....	7
3.2.1	Neformální vzdělávání.....	7
3.3	Budoucí trendy	8
4	Návrh požadavků pro pracovní prostředí nové mobilní aplikace	9
4.1	Návrh základního rámce.....	9
4.1.1	Business požadavky.....	9
4.1.2	Technické požadavky.....	11
4.2	Implementace	11
4.3	Testování a odstraňování chyb	11
4.4	Nasazení, údržba a další rozvoj.....	12
5	Uživatelské prostředí	14
6	Použité technologie.....	16
6.1	Mobilní aplikace.....	16
6.1.1	Java	17
6.1.2	Android Studio.....	18
6.1.3	Aktivita.....	19
6.1.4	Fragment.....	20
6.1.5	XML	21

6.2	Knihovny	22
6.2.1	Knihovna pro čtení QR kódu	23
6.2.2	Knihovna pro práci s mapou	23
6.2.3	Volley	24
6.3	Databáze MySQL	24
6.4	PHP	25
7	Vývoj	27
7.1	Mobilní aplikace	27
7.1.1	Organizace projektu	27
7.1.2	Layouty	28
7.1.3	Navigační menu	30
7.1.4	Přihlášení	31
7.1.5	Řešení práv	32
7.2	Funkcionality	32
7.2.1	Rozpoznání QR kódu	33
7.2.2	Kompas	34
7.2.3	Mapa	36
7.2.4	Čtení a zápis pomocí technologie RFID	37
7.2.5	Morseova šifra	38
7.3	Serverová část	39
7.4	Integrace a správa dat v aplikaci	41
8	Závěr	42
9	Seznam použité literatury	44
10	Přílohy	47

Seznam obrázků

Obrázek 1: Porovnání Gradle proti Maven	19
Obrázek 2: Dvě rozdílné verze rozlišení s využitím fragmentů a aktivit	21
Obrázek 3: Organizace dat	27
Obrázek 4: Příklad layoutu	29
Obrázek 5: Ukázka hlavního menu	30
Obrázek 6: Ukázka přihlášení	31
Obrázek 7: Ukázka skenovacího okna	33
Obrázek 8: Ukázka kompasu	34
Obrázek 9: Ukázka recalibrace	35
Obrázek 10: Ukázka mapy	36
Obrázek 11: Ukázka RFID	37
Obrázek 12: Ukázka zápisu RFID	38
Obrázek 13: Ukázka překladu Morseova kódu	39
Obrázek 14: Ukázka databáze	40

Úvod

V dnešní digitální době se inovativní informační technologie prolínají životem všech osob, čímž se naskýtají nespočetné možnosti v oblasti vzdělávání. Jednou z těchto oblastí, kde mohou být moderní technologie využívány, je edukační proces v oblastech neformálního vzdělávání na dětských táborech. Dětské tábory představují nejenom prostor, kde mládež tráví čas volnočasovými aktivitami, a to pro nezapomenutelné zážitky, ale taktéž nové příležitosti, aby se děti naučily novým dovednostem a rozšířily si své oblasti vědění. Problematika sebevzdělávání v rámci participace mládeže při skautingu byla řešena i v rámci kvalitativního výzkumu ze strany Polského svazu skautingu a vůdcovství [1]. Volnočasové aktivity a další vzdělávání na akcích pořádaných pro děti v rámci různých táborů či spolků je tedy možné zařadit do oblasti neformálního vzdělávání.

V době před vznikem chytrých telefonů byly veškeré aktivity řešeny pomocí papírových map, kompasů a dalších pomůcek. Instruktoři učili mládež, jak tyto nástroje správně používat prostřednictvím různých typů her. Práce s těmito pomůckami je v dnešní době nepraktická a velká část dětí by si kompas domů nejspíš ani nepořídila.

Dětské tábory, které pamatují naši rodiče, jsou spojené se vzpomínkami na venkovní aktivity plné různých sportovních her a samotných sportů, jako je plavání, ale také na táboření či stanování. Rodiče či prarodiče na tyto aktivity velmi rádi vzpomínají, protože to byla doba bez digitálních technologií. Pobyt venku na čerstvém vzduchu byl mnohdy vyčerpávající, avšak plný adrenalinu a radosti. V minulém století se na organizaci outdoorových her, venkovních sportů a volnočasových aktivit spojených s dětskými tábory podílel TJ Sokol či Junák – český skaut, které fungují dodnes [2].

Postupným rozšiřováním digitálních technologií však docházelo k útlumu venkovních aktivit i na dětských táborech, jež začaly být spojovány s využíváním mobilních telefonů s odpovídajícími aplikacemi. Tyto aplikace mohou dětem nabídnout zlepšení zážitků při různých volnočasových aktivitách, přičemž současně jsou využívány také pro celkovou efektivní organizaci různých aktivit v elektronické podobě.

Digitální technologie tak vnáší do oblasti pořádání dětských táborů inovativní prvky, které jsou využívány pro zvýšení efektivity, neformální vzdělávání a současně i pro zábavu na dětských táborech.

Proto je také existence aplikací, jež dětem ukážou využití těchto chytrých zařízení rovněž v oblasti tábornictví, žádoucí. Hlavní je pak jejich zapojení v rámci táborových aktivit, čímž táboroví vedoucí získají za účelem tvorby nových her a aktivit více možností.

Cílem této bakalářské práce je návrh a vývoj mobilní aplikace pro využití v oblasti neformálního vzdělávání mládeže na dětských táborech s implementací funkcionalit, jako jsou např. kompas či práce s mapovými podklady.

V první části je rozpracován teoretický rámec, který zahrnuje analýzu existence již podobných projektů, následný postup při zpracovávání aplikace a v neposlední řadě potřebné technologie k vytvoření aplikace.

V praktické části jsou charakterizovány dílčí části vytvořené aplikace včetně popisné charakteristiky všech modulů s jejich funkcionalitami. V závěru práce je pak zhodnocen celkový přínos práce a využitelnost pro praktické využití a nastíněn další možný vývoj v téže oblasti.

Dalším dílčím cílem je zefektivnění práce vedoucích táborů z hlediska organizačního pořádání táborů a jejich aktivit. Přínosy této kvalifikační práce mohou být využitelné pro organizátory dětských táborů z hlediska zvýšení efektivity při organizování táborů a všech souvisejících aktivit včetně volnočasových. Aplikace může nalézt uplatnění i v rámci jiných spolků, jež se podílejí na neformálním vzdělávání, především u sportovních spolků TJ Sokol, Junák – český skaut či jiných neziskových organizací, jež se v městech a obcích podílejí na organizování volnočasových aktivit pro mládež. Celkové přínosy práce jsou využitelné v rámci praktického uplatnění s možností dalšího rozvoje na základě individuálních požadavků.

Motivací pro zpracování tohoto tématu byla možnost vytvořit inovační vzdělávací aplikaci, která by mohla být využita nejen na táboře Podkova, ale také na ostatních táborech, které by měly o tuto aplikaci zájem.

1 Cíl a metodika práce

V této kapitole jsou rozpracovány cíle práce a metodologický postup pro řešení stanovených cílů a přínosů práce.

1.1 Cíle práce

Hlavním cílem této bakalářské práce je vývoj interaktivní mobilní aplikace určené pro mládež a instruktory, kterou bude možné využívat na dětských táborech. Tato aplikace bude využívána nejen jako výukový nástroj, jenž pomůže zvládnout práci s funkcionalitami typu kompas, RFID technologie, skener QR kódu, mapa, ale také může být využívána jako prostředek pro tvorbu nových interaktivních her. Samotná aplikace bude dostupná na zařízeních s operačním systémem Android.

Vedlejším cíle práce je možnost využívání této aplikace pro efektivní pořádání a celkovou organizaci související s pořádáním dětských táborů.

Pro splnění hlavního cíle byly stanoveny dílčí cíle vedlejší:

1. Výukové technologie

Důležitou součástí projektu budou funkce, které mládeži umožní vypořádat se s venkovní orientací. Prvním aspektem bude implementace mapy a kompasu v rámci mobilní aplikace. Tato funkcionalita umožní uživatelům zobrazit svou aktuální polohu na mapě, přidávat nové body na mapu a navigovat se pomocí kompasu.

Další klíčovou funkcionalitou bude práce s RFID technologiemi. Tato funkce bude podporovat dva režimy: režim čtení, který bude využíván jak instruktory, tak mládeží, a režim zápisu, který bude dostupný pouze instruktorům. Tato technologie umožní táborníkům vytvářet nové aktivity a hry, jako je např. hledání nových předmětů.

Jednou z dalších klíčových funkcionalit aplikace bude schopnost skenovat QR a čárové kódy. Tato funkce bude sloužit k poskytování různých informací mládeži, které mohou být součástí daných úkolů nebo her. Tyto informace budou dostupné prostřednictvím vytisknutých QR kódů umístěných v táboře či v oblasti, kde daná aktivita probíhá.

V neposlední řadě bude aplikace zahrnovat funkci, která umožňuje mládeži využívat Morseovu abecedu a tím se i naučit její základní znaky a dovednosti spojené s touto formou komunikace.

2. Interaktivní výukové aktivity

Dalším cílem je vytvoření různých interaktivních výukových aktivit a her, jež budou využívat všechny výše zmíněné technologie. Tyto aktivity pomohou mládeži lépe porozumět použití daných technologií a tomu, co práce s nimi obnáší.

3. Správa dat a uživatelské účty

Aplikace bude rovněž zahrnovat mechanismy pro správu uživatelských účtů a ukládání dat o postupu mládeže v rámci táborových her, které budou zaznamenávány do bodového žebříčku. Samotné uživatelské účty budou rozděleny na základě práv, a to na administrátorský a uživatelský účet, kdy uživatelé budou mít převážně práva pro čtení, zatímco administrátoři budou mít možnost zápisu do databáze.

1.2 Metodologický postup

Pro zpracování této bakalářské práce byl navrhnout metodologický postup, který je rozdělen do dvou hlavních částí a dílčích podčástí:

1. Teoretická část zahrnovala tyto dílčí podčásti:
 - analýza a význam stávajících softwarových řešení využívaných na dětských táborech;
 - analýza a komparace edukačních aktivit v oblasti neformálního vzdělávání na dětských táborech;
 - návrh požadavků pro pracovní prostředí nové mobilní aplikace s využitím na dětských táborech;
 - návrh struktury uživatelského prostředí a oprávnění včetně vhodného výběru technologií, které budou využity pro vývoj aplikace.
2. Praktická část zahrnovala následující dílčí podčásti:
 - vývoj mobilní aplikace na základě navržených požadavků a funkcionalit v teoretické části;
 - testování vytvořené aplikace v rámci testovacího provozu.

Metody

Za účelem zjištění relevantních informací pro zpracování této práce byly využity především metody analýzy, generalizace, indukce, dedukce a komparace.

Pro zpracování této kvalifikační práce byla využita kombinace uvedených metod tak, aby byly splněny stanovené cíle a přínosy práce. V práci pak byly využity i některé anglicismy, jež není možné nahradit českým synonymem.

2 Stávající softwarová řešení a jejich význam

V dnešní době existuje mnoho způsobů, jak zaujmout velké množství mládeže na dětských táborech, a to především využitím mobilních zařízení v rámci táborových aktivit. Většina těchto zařízení je již vybavena řadou senzorů, které by se daly potencionálně využít v konkrétních hrách jako zpestření. Jejich další možnost využití je i zároveň jako edukace právě s těmito technologiemi. Sensory a funkce, kterými běžně telefony disponují, jsou např. GPS lokátor, fotoaparát, RFID senzor, magnetometr a akcelerometr. Často však také děti využívají mobilní zařízení pro sdílení různých souborů a fotografií, a hlavně pak pro komunikaci mezi sebou za použití různých sociálních sítí.

Využití výše zmíněných funkcionalit se již v praxi na některých táborech využívá. Jedná se o skupinu aplikací třetích stran, jakými mohou být např. Mapy.cz nebo Google maps, QR scanner či kompas, které se dají stáhnout zcela zdarma z obchodu Google play nebo App store od Apple [3].

V zahraničí se již na některých dětských táborech přímo zaměřují na edukaci s novými technologiemi, a to převážně s aplikacemi App Inventor [4] a LEGO mindstorm [5]. Využití těchto aplikací v rámci letních táborů mělo za cíl naučit mládež základy programování v jazyce Java. Výstupem pak nebyla jen aplikace pro operační systém Android, ale také základy robotiky díky propojení fyzické stavebnice LEGO a její aplikace. Dalším účelem bylo zvýšit počet kvalifikovaných lidí v oboru softwarového inženýrství a robotiky, a sice tím, že poskytl tyto aplikace mladistvým, čímž je třeba mohli nasměrovat na příslušný typ studií.

2.1 Význam mobilních a počítačových aplikací v kontextu dětských táborů

Na letních táborech jsou spíše využívány fyzické pomůcky ve formě různých nástrojů, jakými jsou např. papírové mapy, kompas a další. S ohledem na mládež, která však využívá mobilní telefony od raného věku, je vhodné využívání těchto zařízení v rámci volnočasových aktivit, a to např. i v oblasti související s dětskými tábory.

Na dětských táborech jsou využívány různé aplikace, které jsou volně dostupné na platformě Google play. Jedna z významných aplikací, jež byla využívána, se jmenuje *Stopovaná*. Celkový princip aplikace je založen na QR kódech, prostřednictvím kterých děti hledají na vyznačené trase vytisknuté QR kódy. Uvnitř QR kódů jsou schované

souřadnice pro další stanoviště. Díky těmto souřadnicím pak táborník věděl, jakým směrem se vydat. Další aplikací mohou být i mapy, díky nimž se děti odvedou do místa A následně se musí podle mapy za dozoru vedoucích dostat do místa B [3].

Počítačové aplikace mohou být využívány také pro tábory, které se specializují na typ výuky se specializovaným softwarem za účelem zvýšení povědomí o některých odvětvích informačních technologií. Příkladem může být využití LEGO mindstorm, které rozvíjí vědomosti o robotice u dětí a její využití v praxi [5].

2.2 Edukace prostřednictvím mobilních a počítačových aplikací na dětských táborech

Samotnou edukaci pro oblast letních táborů lze rozdělit na dvě části. První oblast vzdělávání je zaměřena na tvorbu aplikací a práci s roboty. Naopak druhá oblast je zaměřena na výuku dětí s technologiemi principiálně již staršími, převedenými akorát do digitální podoby. Tato starší oblast je nazývána jako oblast tábornická.

V první uvedené oblasti můžeme zmínit počítačové aplikace, jako jsou LEGO mindstorm či App Inventor, které mají za úkol zábavnou formou naučit děti nutný základ k programování a práci s roboty. Tato informace se jim poté může hodit např. v budoucím studiu na střední anebo vysoké škole. Tábory patřící do této oblasti bývají časově delší a převážně městské.

Ve druhém odvětví se mládež učí využívat mobilní aplikace, jako jsou mapy, kompas a QR kódy. Tyto znalosti je mohou naučit, jak řešit běžné situace, které mohou nastat kdykoliv v jejich budoucím životě. Např. dostat se z bodu A do bodu B pomocí mapy nebo dojít za pomoci kompasu na předem dohodnuté místo. Případně se také dozvědí, jakým způsobem lze např. využít QR čtení pro poslání platby za elektřinu. Edukační činnost spojená s volnočasovými aktivitami na dětských táborech je řazena do oblasti neformálního vzdělávání.

2.2.1 Neformální vzdělávání

Neformální vzdělávání se uskutečňuje mimo formální vzdělávací systém, ve kterém je možné po dokončení veškerých požadavků získat určitý typ certifikátu, např. vysvědčení nebo diplom.

Hlavní rozdíl mezi formálním a neformálním vzděláváním je tedy takový, že neformální je zcela dobrovolné. Toto vzdělávání je založeno na různých výchovně vzdělávacích aktivitách mimo rámec oficiálního postupu vzdělávání mládeže

zavedeného příslušným státem (předškolní, základní, střední a vysokoškolské vzdělávání). Tento postup nabízí současně rozvoj nových životních zkušeností, dovedností a postojů [6].

Samotnými organizátory potom bývají různá uskupení lidí, kteří mají zájem předávat své zkušenosti a vzdělání i jiným způsobem než učitelskou profesí. Příkladem těchto uskupení mohou být např. školní družiny, kluby nebo soukromé jazykové školy. Taktéž to mohou být různé domy dětí a mládeže, které nabízejí volnočasové kroužky a aktivity pro děti. Nelze však opomenout ani organizace, které se zabývají tábornictvím, a taktéž tak spadají do nějakého sdružení [7].

Účelem tábornických organizací může být rovněž zvyšování fyzické kondice dětí a nauka určená k získání rozmanitých dovedností, např. vázání různých typů uzlů, využívání kompasu za účelem hledání různorodých předmětů, stavba tábornických ohnišť či zacházení s ohněm a řeznými nástroji na úpravu dřeva.

2.3 Budoucí trendy

Trendem blízké budoucnosti mohou být i inovace v návaznosti na propojení dalších technologií, jako je např. umělá inteligence. Ta bude v budoucnu využívána jako chytrý rádce pro mladistvé nebo pro zjednodušení práce vedoucích. Toto rozšíření by mohlo být žádoucí také pro tábory specializující se na oblast sportu, kde by umělá inteligence mohla vyhodnocovat různé míčové hry. Těmi mohou být např. varianty volejbalu, vybíjená či další sporty [8].

Princip těchto variant volejbalu, vybíjené a dalších sportů je založen na systému kamer sledujících hrací prostor a hráče. Po uplynutí času umělá inteligence vyhodnotí výsledky a předá je instruktorům pro následné zpracování. Tato práce zjednoduší psaní a vypočítávání konečného skóre míčové hry [9].

Další budoucí trendy můžeme nalézt i v oblastech jiných sportů, které mohou být realizovány na dětských táborech, a to např. v rámci specializovaných ozdravných pobytů či jiných obdobných akcí organizovaných za účelem zlepšení zdravotního stavu se zaměřením na sportovní aktivity v přírodě.

3 Návrh požadavků pro pracovní prostředí nové mobilní aplikace

V této kapitole jsou rozpracovány dílčí části pro vývoj nové softwarové aplikace. Tyto dílčí části, dále označované jako fáze, byly rozděleny do čtyř částí:

- návrh základního rámce (analýza požadavků a prostředí);
- implementace;
- testování;
- nasazení, údržba a další rozvoj.

3.1 Návrh základního rámce

V počáteční fázi je nutné navrhnout základní rámec budoucí softwarové aplikace, přičemž je třeba především definovat požadavky. Na základě stanovených požadavků bude následně navržen procesní model pro novou aplikaci včetně konkrétních funkcionalit.

Požadavky na novou aplikaci byly rozděleny na dvě části:

- business požadavky – účel aplikace a cíle aplikace, přístupová oprávnění a využití v praxi;
- technické požadavky – analýza a definice nároků na hardware a software včetně požadavků na výkonnost [10].

Tato fáze se pro mnohé může zdát být nepotřebnou, avšak pro efektivní vývoj aplikací je tento postup zcela klíčový. Hlavně proto, že v této fázi jsme schopni odstínit, a dokonce i přímo odstranit chyby, které by nám poté vznikly v průběhu implementace samotného kódu.

Tyto chyby, jež by mohly potencionálně nastat, by v následném vývoji dovedly způsobit zbytečnou ztrátu času a navýšení složitosti samotného programování. Vzhledem ke schopnosti analyzovat vše již na počátku jsme poté schopni se tomuto problému vyhnout.

3.1.1 Business požadavky

Tvorba a odpovídající analýza těchto požadavků je nezbytnou částí pro pochopení veškerých funkcionalit, které bude nová aplikace vykonávat a podporovat. V této fázi byly nejprve identifikovány skupiny uživatelů, které budou novou aplikaci

využívat, a následně byla těmto uživatelům přiřazena odpovídající role, přístupová oprávnění a odpovědnost. Tento proces bývá velice náročný pro všechny zúčastněné strany, a to jak pro stranu samotných tvůrců aplikací, tak i pro zadavatele zakázky.

Z podstaty věci lze říci, že se jedná o sběr dat, které potřebujeme pro vývoj dané aplikace. Tento sběr informací probíhá v rámci komunikace se zákazníkem, který má nějakou vlastní představu o budoucím softwaru. Ta však může být ovlivněna mnohými faktory, jako je např. nedostatek vzdělání v odvětví tvorby aplikací, na základě čehož je pak nutné, aby zákazník co nejlépe popsal a vysvětlil veškeré své požadavky a funkcionality na novou aplikaci. Především je nezbytné, abychom získali detailní popis i těch aspektů, které jsou pro zákazníka relevantní. Bez těchto informací by totiž mohlo dojít k situaci, kde by byl programátor nucen v implementační fázi rozhodovat na základě svého uvážení, což by mohlo vést k neshodě s požadavky zákazníka. Je tedy důležité, aby před zahájením vlastního vývoje nové aplikace vývojový tým disponoval všemi důležitými informacemi tak, aby byl zajištěn vývoj nové aplikace v souladu s požadavky a očekáváními objednavatele služby.

Dalším problémem, který může nastat, je ten, kdy se zákazník zaměřuje na určitý detail, nebo trvá na postupu řešení, který by však šel vyřešit mnohem efektivnějším a snadnějším způsobem. Důsledkem toho bývá zbytečně složitá funkce, která pak komplikuje fungování jinak jednoduché aplikace, a pro uživatele může být i toto řešení v budoucnu složité na pochopení.

Z předchozích problematik lze vyvodit, že projektový tým nebo tvůrce aplikace by měli svými dotazy získat co možná největší množství potřebných informací. Tyto informace je pak nutné zařadit si do kontextu budoucí aplikace tak, aby bylo možné mít už jasnou představu o budoucím řešení [10].

V rámci analýzy a komunikace se zadavatelem byly identifikovány potřebné role a požadavky, které by měly být pro naši aplikaci základním stavebním kamenem, a sice pro následnou implementaci:

- Role: Účastníci tábora. Vedoucí tábora. Správce aplikace. Správce databáze.
- Požadavky: Dostupnost na mobilních zařízeních se systémem Android. Možnost práce s mapou a kompasem. Evidence a rozřazení účastníků do týmů. Tvorba různých úkolů a napojení fyzického prostoru na mobilní v rámci různorodých typů her.

Detailní technické rozpracování je uvedeno v kapitole č. 4 Uživatelské prostředí.

3.1.2 Technické požadavky

Fáze technických požadavků je na rozdíl od business požadavků značně komplikovanější, obzvláště ve chvíli, kdy má zákazník malou odbornost v oblasti vývoje aplikací. Pakliže tato situace nastane, většina odpovědnosti připadne na osobu, která bude daný software vytvářet. Tato osoba musí s ohledem na nedostatek detailů uvážit, jaká řešení klientovi doporučit z pohledu jak cenového, tak i technického [10].

V případě této práce bude potřeba zvolit řešení formou mobilní aplikace, hlavně pro možnost využívat aplikaci kdekoli a kdykoli. Dalším důležitým benefitem je vcelku intuitivní rozhraní, které se dá v rámci chytrých telefonů vytvořit. S ohledem na finanční rozdíly u těchto typů zařízení se bude aplikace vyvíjet hlavně pro chytré telefony s operačním systémem Android, které jsou cenově dostupnější a zároveň výkonově dostačující pro tento typ softwaru.

Nezbytným technickým požadavkem na uvedené mobilní telefony je, aby disponovaly senzory, jako jsou magnetometr, akcelerometr, GPS a čtečka RFID.

3.2 Implementace

V implementační fázi byly požadavky, jež byly definovány v předchozí kapitole, převedeny pomocí programovacího jazyka do zdrojového kódu aplikace. Podstatnou roli při návrhu a následném vývoji nové softwarové aplikace zaujímá vhodný výběr programovacího jazyka, ve kterém bude aplikace vyvíjena. V našem případě byl využit programovací jazyk Java a vývojové prostředí Android Studio, které je pro tvorbu aplikací s tímto operačním systémem nejvhodnější [11]. Podrobná specifikace programovacího jazyka Java a vývojového prostředí Android Studio je rozpracována v kapitole č. 5 Použité technologie.

3.3 Testování a odstraňování chyb

Testovací fáze a následné odstraňování chyb se v časovém horizontu částečně prolíná s implementační fází. K tomuto prolínání dochází především z důvodu průběžné kontroly stávajícího kódu a zároveň toho, zda jeho část, jež byla nově přidána, funguje podle definovaných požadavků a nijak neomezuje či nerozsbíjí již stávající kód. Nejedná se nicméně o komplexní testování. Vývojář testuje a analyzuje jen tu část kódu, kterou aktuálně naprogramoval a která by mohla mít dopad na nějakou další část již vytvořené aplikace.

Za odpovědné komplexní testování jsou zodpovědné tyto dvě fáze [12]:

- interní – FAT (Factory acceptance test);
- externí – UAT (User acceptance test).

Interní testování je prováděno před samotným nasazením produktu do zákaznických zařízení. Samotné testování probíhá tak, že je kontrolováno, zda je aplikace kompletní z pohledu domluvených funkcionalit, jež byly stanoveny při prvotním návrhu nové aplikace. V ideálním stavu je toto testování prováděno dalším testerem, který na aplikaci nepracoval, ale disponuje odpovídajícími znalostmi ohledně vyvíjené aplikace včetně znalostí programovacího jazyka Java a Android Studio. Výhoda testování dalšími osobami spočívá v tom, že nezávislý tester je schopen získat další úhel pohledu na testovanou aplikaci.

Po interním testování chronologicky následuje testování externí, během něhož je již aplikace přímo testovaná zákazníkem v rámci praktického využití. Při tomto testování je proces obdobný jako u interního testování ze strany nezávislého testera, avšak již je realizováno v reálném prostředí.

Po ukončení vývoje této aplikace proběhlo testování interní i externí ze strany autora a nezávislého testera. Důvodem pro tuto dvoustrannou kontrolu bylo plánované skutečné využití aplikace v praktickém provozu na dětském táboře, na němž autor této práce současně i působí jako odborný instruktor.

3.4 Nasazení, údržba a další rozvoj

Tato fáze spočívala v nasazení vytvořené aplikace do provozu po provedení interního a externího testování. Pro přímé využívání nové mobilní aplikace je nutno si stáhnout a nainstalovat aplikaci do mobilních zařízení, mobilní zařízení však musí splňovat stanovené technické požadavky na operační systém.

Další etapou v této fázi bylo vytvoření a spuštění databáze. Pro vytvoření, konfiguraci a naplnění databáze bylo nutné vhodně zvolit cílový server, na kterém bude aplikace provozována. Na tomto serveru bude provozován hosting, jenž bude vhodně nakonfigurován včetně veřejné IP adresy. Hostingový server lze pořídit dvěma způsoby, a sice pronájemem v rámci placené služby nebo prostřednictvím instalace hostingových služeb na vlastním serveru. Pokud bychom si zvolili variantu vlastního serveru, bylo by nutné pořízení vlastního hardwaru a zřízení veřejné IP od providera. Vlastní serverové řešení je však finančně náročné, vhodnějším způsobem je tak pronájem hostingového serveru. Pronájem hostingového serveru je finančně méně nákladný, a současně výhodný, protože dílčí konfigurace jsou řešeny ze strany

poskytovatele hostingových služeb. Mnohdy poskytovatel nabízí také službu instalace a individuální konfigurace provozovaných aplikací ze strany nájemce, čímž je celý proces instalace a spuštění nové služby (softwaru) zjednodušen.

Další fáze, jako je údržba a další rozvoj oproti předešlým etapám vývoje, jsou již nepovinné. Avšak mohou být v rámci dodávané aplikace zákazníkem požadované, čímž dochází ke zvyšování úrovně spokojenosti u všech osob, které budou se softwarem pracovat na denní bázi [13].

Výše uvedené části se dají dále rozdělit do dvou rozdílných sémantických oblastí. Jedna z nich se váže na sledování technické části, kde je řešeno, zda je komunikace mezi aplikací a databází stabilní, bezproblémová a s co nejmenší reakční odezvou. Druhá se zabývá chybovými hlášeními přímo od uživatelů aplikace. Tyto zprávy mohou být různého typu, a sice od řady nečekaných pádů aplikace až po nedokonalé specifikace té části požadavků, jež nebyly odhaleny v první fázi vývoje. Poslední typ chyby může vzniknout v případě, kdy zákazník zapomněl na některý svůj požadavek, protože mu např. přišlo zbytečné jej vůbec zmiňovat, neměl na něj dostatečný rozpočet, popřípadě z důvodu blížící se ho konečného termínu dodání aplikace [13].

Případný další budoucí rozvoj aplikace se již může zakládat na vytvořené předloze, nicméně bude potřeba znovu vést komunikaci se zadavatelem a analyzovat nové požadavky na řešení budoucích problémů a rozšíření.

4 Uživatelské prostředí

V této kapitole je rozpracován návrh uživatelského prostředí a jednotlivých funkcionalit vyvíjené aplikace. Uživatelské prostředí je následně analyzováno v kapitole č. 6.1.3 Navigační menu. Účelem této kapitoly bylo podrobněji popsat funkční požadavky uživatelského prostředí, které vznikly během fáze analýzy (návrhu). Uživatelské prostředí vyvíjené aplikace vytváří a zobrazuje hlavní myšlenku toho, jak by budoucí aplikace měla vypadat a fungovat.

Z hlediska uživatelského oprávnění (úctů) bylo nutné definovat dílčí aktéry, kteří budou přistupovat k mobilní aplikaci nebo databázi.

Druhy uživatelských účtů:

- Účastník tábora:
 - čtení různých formátů QR kódu a čárového kódu, také čtení RFID;
 - geolokační prvky – kompas a mapa, kde bude možné přidávat dočasné body pro různé hry;
 - tvorba a poslech Morseova kódu;
 - zobrazení bonusových úkolů, které mohou posunout jeho tým k vítězství;
 - zobrazení průběžného žebříčku týmů.
- Vedoucí tábora:
 - stejné funkce jako uživatel „účastník tábora“ a další rozšíření;
 - zápis RFID;
 - správa bodů na mapě přímo v databázi pro delší uchování;
 - správa bonusových úkolů, her a jejich výsledků přímo v databázi.
- Správce aplikace a databáze:
 - správa účtů a týmů.

Grafická úprava a celková struktura uživatelského rozhraní a oprávnění je základním požadavkem, přičemž hlavní důraz je kladen na přehlednost a jednoduchost celého rozhraní. Tento požadavek vzešel z problematiky, která se váže na využívání aplikace různými věkovými kategoriemi. Víme, že aplikaci budou využívat také děti, které právě dokončily první rok povinné školní docházky na základních školách, tudíž i tento klíčový aspekt musí být při návrhu uživatelského rozhraní a oprávnění zohledněn již v samotném návrhu aplikace.

Taktéž nelze opomenout požadavek na operační systém, na němž bude aplikace fungovat. Z důvodu finanční náročnosti pro tábory i rodiče byl zvolena varianta mobilních telefonů se systémem Android. Jedná se totiž o nejlevnější variantu, která však obsahuje veškeré potřebné prvky pro tuto aplikaci.

Dalším požadavkem, který vyplývá z předešlého seznamu, je rozdělení pravidel pro přístup k aplikaci. Jinými slovy aplikace by měla rozlišit dva hlavní typy účtů:

- klasický uživatelský účet pro účastníky;
- administrátorský účet pro vedoucího.

Hlavní rozdíl v těchto typech uživatelských účtů bude spočívat v možnosti zápisu a čtení z databáze. Administrátor má všechna práva, zatímco účastníci mají převážně práva jen pro čtení.

Jednotlivé funkce, jako jsou RFID, mapy, skener QR a čárového kódu a Morseova šifra, budou rozděleny do jednotlivých účtů a budou zde definována i práva, která budou jednotlivým účtům náležet.

Samotné rozhraní bude rozděleno na uskupení tlačítek tak, aby bylo možné intuitivně rozeznat, co má které tlačítko dělat a co za funkci pod sebou skrývá. Díky tomu by ani ty nejmenší děti neměly mít problém s orientací v aplikaci.

5 Použité technologie

V této kapitole a dílčích podkapitolách je zpracován podrobný popis využitých technologií a prvků. Nejprve je charakterizována struktura vyvíjené mobilní aplikace a její významné komponenty, následně jsou popsány všechny použité knihovny, jejich význam a aplikace v kontextu vývoje a dále je zde rozpracována implementace samotné databáze SQL.

5.1 Mobilní aplikace

Na základě definovaných stanovených cílů a přínosů práce byl zvolen vývoj aplikace určené pro mobilní zařízení. Mobilní aplikace byla zvolena na základě možnosti využívání všech funkcionalit v rámci sportovních aktivit v přírodě, kde jsou dětské tábory obvykle realizovány. Mobilní aplikace není pro účastníky nikterak složitá a bude založena na programovacím jazyku Java, který při vývoji aplikace využívá existující zdrojové balíčky, jež lze při vývoji individuálně upravit na základě vlastních požadavků.

Než se přesuneme k samotným technologiím, je vhodné si nejprve obecně vysvětlit vývoj aplikací pro mobilní operační systém. Během tohoto vývoje mohou totiž nastat problémy, které je důležité neopomenout. V první řadě si musíme uvědomit, že tato zařízení mají opravdu malé obrazovky na rozdíl třeba od tabletů nebo notebooků. Dalším problémem může být rozdílná velikost displeje. V současnosti se vyrábí telefony s velikostí od tří do šesti palců, přičemž nejvíce využívanou velikostí je ta mezi pěti až šesti palci. V důsledku toho může docházet např. k tomu, že se nám třeba tlačítko, které nastavíme na určitou velikost, bude na každém telefonu chovat jinak. Je tedy nutné při vývoji aplikace volit responzivitu pro obvykle využívané typy velikosti displeje [14].

Rovněž musíme vzít v potaz to, jakým způsobem držíme telefon, a následně z toho také odvodit tu nejpohodlnější polohu pro tlačítka. Oproti počítačům se totiž jedná o interaktivní prostředí, se kterým je nutné komunikovat dotykem. Nejlepší přístup v případě, že máme uskupení tlačítek, je situovat je do jednoho sloupce, který je vycentrovaný na střed. Tímto způsobem usnadníme koncovému zákazníkovi držení telefonu a přístup k příslušným tlačítkům [14].

V poslední řadě musíme zanalyzovat vybavení telefonů a zjistit, jaká oprávnění a přístupy vlastně budeme k vývoji aplikace potřebovat. Dnešní chytré telefony totiž mohou nabízet spoustu senzorů a prvků, které slouží např. lokalizaci GPS, nebo dokonce pohybu samotného telefonu jako např. gyroskop či akcelerometr, anebo fotoaparátu. Každý zmíněný prvek bude potřebovat jistá oprávnění, a proto musíme vyfiltrovat ta, jež v naší aplikaci potřeba nebudou. Kdybychom tak neučinili, mohla by nám vzniknout bezpečnostní rizika, kterým se chceme v první řadě vyhnout. Pokud bychom např. vyvíjeli aplikaci na QR kód, tak určitě nebudeme potřebovat přístupy k mikrofonu na rozdíl od přístupu ke kameře, který bude naopak potřeba [14].

V následující podkapitole je dále popsán zvolený programovací jazyk, který je nedílnou součástí tvorby aplikací, a je tedy nejdůležitějším prvkem v rámci této práce. V našem případě je využit jazyk Java.

5.1.1 Java

Jedná se o jeden z mnoha programovacích jazyků, který nám nabízí širokou škálu výhod. Jednou z jeho velkých předností je skutečnost, že se jedná o multiplatformní jazyk. Multiplatformní jazyk můžeme definovat jako jazyk, prostřednictvím kterého lze programovat různé typy softwarových aplikací, a to jak pro telefony, tak pro serverová zařízení nebo stolní počítače [15].

Jeho další výhodou je to, že Java, oproti procedurálním jazykům, které rovnou pracují s vytvořenými metodami nebo procedurami, přistupuje ke kódu objektově. To znamená, že tento jazyk vytváří objekty, které obsahují jak data, tak metody. Mezi další výhody patří jednoduché spouštění kódu a neomezené využívání volání různých naprogramovaných metod. Důležitou předností je rovněž plné využití již hotových aplikací za účelem tvorby jiné aplikace, na základě čehož lze část námi vytvořeného kódu aplikace zkopírovat a použít v novém projektu při vývoji jiné aplikace [16].

Popularita tohoto jazyku mezi developery vzrostla především pro jeho možnost využívat kvalitně zpracovaných výukových zdrojů (balíčků), které lze nalézt na internetu. Nejvyužívanějšími zdroji (balíčky) jsou především W3schools, nebo česká varianta IT network. Taktéž nesmíme opomenout kvalitně zpracovanou dokumentaci v rámci Java Oracle. Další možností je pak využívání vestavěných funkcí a knihoven. Díky tomu není nutné vyvíjet každý projekt úplně od nuly, ale lze částečně využít kódu, který již existuje. Problematiku případného rozšiřování existujících balíčků lze řešit přes různá fóra a blogy s dalšími vývojáři. V neposlední řadě je nutno zdůraznit

kvalitně zpracované vývojářské nástroje, které nám šetří čas – např. nástroje pro ladění a vychytávání chyb, testovací pomůcky či různé skripty pro automatizované úpravy [15].

Z hlediska bezpečnosti je možné využívat předdefinované konfigurace, které lze individuálně upravit dle dílčích požadavků. Ve výsledku to znamená, že je možné stažení jakýchkoliv nedůvěryhodných kódů a jejich spuštění v zabezpečeném prostředí, aniž by stažený nedůvěryhodný kód narušil kybernetickou bezpečnost konkrétní aplikace [15].

Pro využití programovacího jazyku Java je také nutno zvolit vhodné vývojové prostředí, jež bylo stanoveno v předchozích kapitolách. V následující dílčí podkapitole je tedy více popsáno zvolené vývojové prostředí.

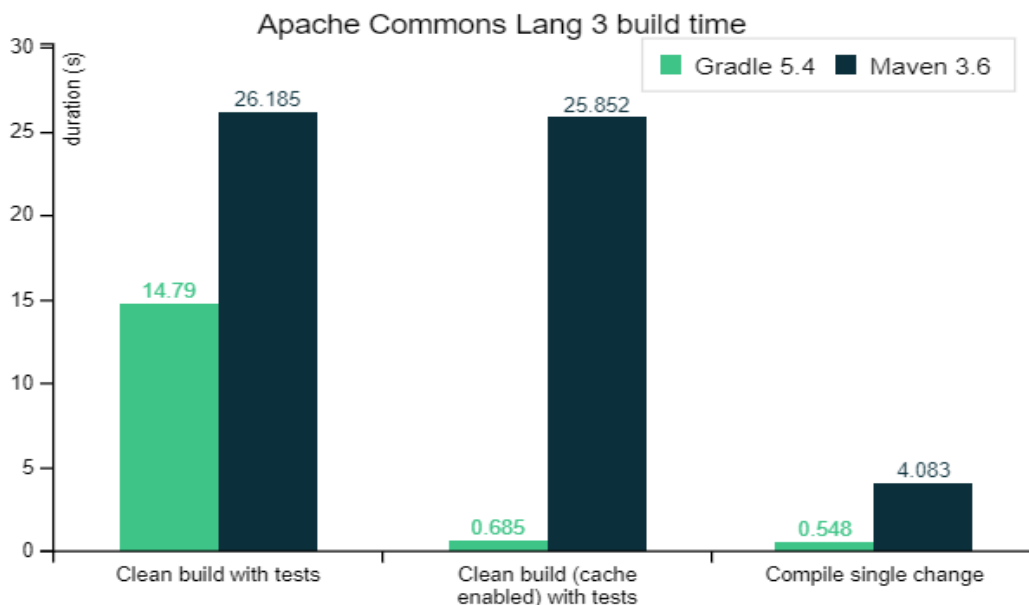
5.1.2 Android Studio

Android Studio představuje oficiální integrované vývojové prostředí (IDE) navržené speciálně pro vytváření aplikací určených pro stejnojmenný operační systém. Toto vývojové prostředí vychází ze silného a zároveň velmi oblíbeného editoru IntelliJ IDEA. Samotné Android Studio k němu přidává další prvky, jejichž účelem je zvýšit vývojářskou produktivitu při programování Android aplikací [17].

Jedním z řady vylepšení je flexibilní sestavovací systém založený na Gradle, který je oproti Mavenu rychlejší, flexibilnější a pro uživatele přívětivější. Samotná flexibilita je postavená na oprávněném a odpovědném uživateli. Ten tak má možnost si sestavy upravovat a vylepšovat. Další výhodou je podpora IDE od Gradle, která se neustále vylepšuje a tím přidává spoustu nových aktualizací a úprav, jež mohou být u koncových uživatelů žádané [18].

Gradle obsahuje moderní příkazovou řádku, která poskytuje zjišťovací funkce, jako jsou např. „gradle tasks“, a současně obsahuje také interaktivní webové uživatelské rozhraní pro ladění a optimalizaci sestavení „Build Scan“. Samotné rozhraní pak může být hostované na místě organizace za účelem shromáždění sestav pro získání analýzy trendů, porovnání sestavení pro ladění nebo optimalizace časů sestavení [18].

Na obrázku č. 1 je zobrazeno „Porovnání Gradle proti Maven“ výkonu ve vteřinách u obou zmiňovaných zavaděčů [18].



Obrázek 1: Porovnání Gradle proti Maven

Zdroj: Gradle.org

Dalším významným prvkem Android Studia je jeho rychlý a zároveň velmi dobře zpracovaný emulátor. Přes něj lze simulovat jakékoliv mobilní zařízení s operačním systémem Android. Tuto simulaci můžeme provést virtuálně na počítači nebo notebooku, na němž probíhá samotná tvorba aplikace. Danou službu můžeme také využít na našem fyzickém zařízení, kde lze naši aplikaci prostřednictvím pár kliknutí nainstalovat a tím si zjednodušit práci s hledáním a laděním různých chyb [17].

Také je nutné zmínit univerzálnost tohoto vývojového prostředí s ohledem na rozmanitost systémů Android. Díky tomuto softwaru jsme schopni vyvinout aplikaci pro jakoukoliv verzi, a to bez ohledu na to, zda je stále aktuální a má aktivní podporu. Současně nám umožňuje integraci GitHubu a využití různých šablon pro zlepšení výkonu, v neposlední řadě pak nabízí i vestavěnou podporu pro Google Cloud Platform, jejíž využití může být značnou výhodou při vývoji aplikace [17].

5.1.3 Aktivita

Aktivita je třída, která je nejdůležitější komponentou pro tvorbu Android aplikace. Způsob, jímž jsou aktivity spouštěny a kompletovány, je základní součástí tohoto aplikačního modelu platformy. Oproti klasickému programování, kde je využívána primárně metoda *main*, se aplikace v Android Studiu inicializuje za pomoci aktivity. Ta nám po zavolání vrátí metody určené pro zobrazení právě jedné obrazovky.

Těchto aktivit můžeme samozřejmě vytvořit i více, a to za účelem rozdělení aplikace na více oken, jako je např. přihlašovací obrazovka, menu aplikace a další okna [19].

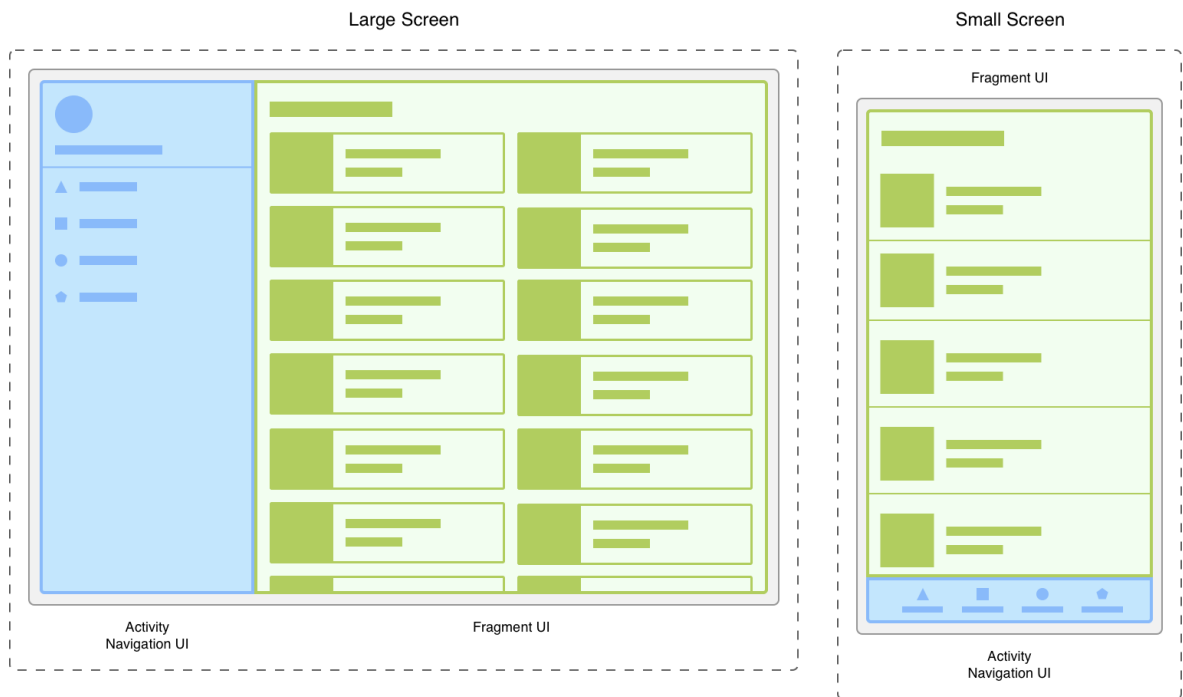
Každá aktivita, která byla vytvořena, musí mít i nadefinovaný layout, což je rozvržení různých prvků na obrazovce, mezi něž řadíme tlačítka, textové pole a další. Existují také společné metody, které jsou součástí takzvaného životního cyklu aktivity. Jedním z nich je např. *onCreate()*, jež se při tvorbě dané aktivity spustí vždy. Kromě spuštění a inicializace dané aktivity a jejího layoutu se nám zde nacházejí také různé operace s příslušným layoutem. Konkrétním příkladem může být třeba odposlouchávání tlačítek nebo toho, co člověk napíše do textových polí, a dalších úkonů. Po ukončení této metody to pokračuje dále, a sice metodou *onStart()*, a je možné využít i dalších služeb, jako je např. *onDestroy()* nebo *onPause()* [19].

5.1.4 Fragment

Fragment je částí uživatelského rozhraní, které můžeme opakovaně využívat v rámci naší vyvíjené aplikace. Prostřednictvím fragmentu dochází k vlastnímu rozvržení a stejně jako aktivita má svůj definovaný životní cyklus. Na rozdíl od aktivity nemůže být spuštěn sám za sebe, a je tedy závislý na jiném fragmentu nebo na námi definované aktivitě [20].

Výhodou fragmentů je hlavně modulárnost a znouvupoužitelnost uživatelského rozhraní v rámci jedné aktivity, a sice tím, že můžeme toto rozhraní rozdělit na několik různých částí. Využívání fragmentů v rámci aktivit nám umožňuje vhodně umístit globální prvky do celého rozhraní naší aplikace. Jednou z aktivit, kterou lze takto použít, je aktivita *drawer*, jež je příkladem menu, v němž pro získání dalších odkazů táhneme prstem směrem dolů. Naopak pakliže bychom chtěli využít uživatelského rozhraní v rámci jedné obrazovky nebo její části, je vhodnější vytvořit samostatné fragmenty [20].

Praktické využití fragmentů je demonstrováno na příkladu s responzivitou na obrázku č. 2, kde obrazovka nabývá různých velikostí. Na obrázku je znázorněné, jakým způsobem by se mohly objekty na různých obrazovkách zpracovat. Na větších displejích by mohl být navigační panel umístěn v levé části obrazovky a oblast s textem např. na pravé straně. Na obrazovkách menších by pak mohlo být navigační menu vložené pro lepší přehlednost pod oblast s textem [20].



Obrázek 2: Dvě rozdílné verze rozlišení s využitím fragmentů a aktivit

Zdroj: developer.android.com

Celkově lze proces rozdělení usnadnit tím, že oddělíme navigační prvky od obsahu. Tímto způsobem svěříme veškerou odpovědnost za zobrazení správného navigačního uživatelského rozhraní aktivitě. Fragment pak bude sloužit pouze k zobrazení textové oblasti. Díky tomu nám vznikne možnost upravovat vzhled za chodu naší aktivity. Musíme však splnit podmínku, abychom v životním cyklu byli na úrovni `STARTED` nebo vyšší [20].

Stav `STARTED` v životním cyklu fragmentu nastává pod dvěma podmínkami: buď je ho dosaženo hned po volání metody `onStart()`, nebo těsně před voláním metody `onPause()` [21].

5.1.5 XML

Zkratka XML vychází z názvu Extensible Markup Language, což znamená, že stejně jako HTML spadá do kategorie značkovacích jazyků. HTML i XML mají mnoho společných rysů, např. se oba skládají z tagů, z nichž lze libovolné prvky definovat a skládat. Jednotlivé tagy nám pak slouží k ukládání a práci s daty, které nám vývoj Android aplikace zjednoduší a poskytne lehkou škálovatelnost [22].

V rámci vývoje naší aplikace nám XML poskytne prostor pro tvorbu grafického uživatelského rozhraní, kde budou do souboru XML definovány tagy. Dané tagy budou souviset se vzhledem naší aplikace, jako je např. uskupení tlačítek, umístění mapy a další [22].

V Android Studiu se můžeme setkat s různými typy XML souborů:

1. Soubory s rozložením UI – soubory s rozložením UI neboli layoutem nám udávají, jak bude naše budoucí aplikace vypadat s ohledem na uživatelské rozhraní.
2. Soubor *AndroidManifest.xml* – poskytuje nám prostor pro popis základních informací o aplikaci, jako je např. název balíčku a další komponenty, mezi něž řadíme aktivity a služby. Nelze opomenout práva k používání různých prvků telefonu, které se taktéž nacházejí v tomto souboru.
3. Soubor *strings.xml* – obsahuje texty pro všechny TextView widgety. To nám umožní znovupoužitelnost kódu a také nám to pomůže s lokalizací aplikace při použití různých jazyků.
4. Soubor *themes.xml* – definuje nám základní motiv, taktéž se využívá k definování nových stylů a hledá uživatelské rozhraní aplikace.
5. Soubor Drawable XML – může poskytnout grafiku prvkům, např. změnit pozadí u tlačítek anebo vytvořit různé přechody.
6. Soubor *colors.xml* – je možné do něj ukládat barvy různého typu požadovaných v naší aplikaci.
7. Soubor *dimens.xml* – je zodpovědný za uchovávání velikostí a rozměrů u pohledů. Může se jednat třeba o výšku tlačítek [22].

5.2 Knihovny

Využití knihoven při vývoji nových aplikací lze přirovnat k běžnému životu tím, že kdykoliv potřebujeme zjistit nové informace, tak navštívíme příslušný objekt, v němž jsou tyto nové poznatky k nalezení. V běžném životě se tyto informace nalézají právě v knihovně, které jsou rozděleny podle jednotlivých kategorií (žánrů), např. sci-fi, akademické, historické a další [23].

V Javě existuje podobná, leč virtuální knihovna, která obsahuje hotové frameworky pro řešení mnoha problémů, jež programátoři v průběhu procesu programování musí řešit. To znamená, že tyto kódy můžeme využívat také k vývoji naší mobilní aplikace určené pro dětské tábory.

Nejznámějšími knihovnami, které jsou součástí Javy, jsou např.:

- Java.lang.math,
 - .sqrt
 - .sin
 - .cos
 - .pow
 - .pi
- Java.util,
- Java.io,
- Java.net.

5.2.1 Knihovna pro čtení QR kódu

Za účelem překladu čárového nebo QR kódu na text bylo zvoleno knihovny s názvem „Zebra Crossing“ neboli ZXING. Tato knihovna je jednou z nejpoblárnějších open-source API právě pro čtení těchto prvků. Ke správné funkčnosti dané části aplikace je nejprve nutné tyto knihovny předat Gradle pro další zpracování a stáhnutí potřebných tříd [24].

Do Gradle skriptu musíme přidat tyto dva řádky, v nichž definujeme knihovny:

- 'com.google.zxing:core:3.4.1',
- 'com.journeyapps:zxing-android-embedded:4.3.0'.

První z výše uvedených knihoven je jádrem samotného frameworku a je nedílnou součástí pro čtení QR a čárových kódů. Druhou zmíněnou knihovnou je dobrovolná knihovna sloužící jednak ke zjednodušené integraci čárových kódů, jednak nám umožní tento QR kód vložit do aktivity nebo fragmentu, jenž je přizpůsoben uživatelskému rozhraní. Taktéž nabízí podporu pro skenování na šířku i výšku [25].

5.2.2 Knihovna pro práci s mapou

Pro práci s mapou byly zvoleny knihovny, které jsou k poskytnutí zcela zdarma, a to přímo od Google. K tomuto kroku je potřeba do našeho Gradle skriptu implementovat tyto tři řádky:

- 'com.google.android.gms:play-services-maps:18.1.0',
- 'com.google.android.gms:play-services-location:21.0.1',

- 'com.google.android.material:material:1.9.0'.

První knihovna se stará specificky o samotnou implementaci mapy do námi zvolené knihovny a dále nám nabízí možnosti přidání různých tlačítek, gest a mnoha dalších funkcí pro práci s mapou. Těmito funkcemi může být např. přidání značek nebo tlačítek pro ovládání přibližování a oddalování mapy [26].

Druhá knihovna nám poskytuje různé informace o lokaci. Především umožňuje aplikacím získat co možná nejpřesnější polohu, čímž se dá např. zobrazit aktuální pozice, anebo můžeme přímo sledovat pohyb vlastního zařízení. Poslední knihovna je grafickým rozšířením první knihovny, kde je možné měnit a přidávat tlačítka, textová pole, seznamy, navigační prvky a mnoho dalšího [27].

5.2.3 Volley

Vzhledem k nutnosti komunikace aplikace s databází bylo potřeba zvolit takovou knihovnu, která nám umožní vést dialog se serverem v rámci protokolu HTTP. S touto problematikou nám dokáže pomoci komponenta Volley, jež nám sama o sobě přidá spoustu nových možností, jak tuto komunikaci vést [28].

Využití tohoto modulu bude vhodné pro řízenou komunikaci s aplikačním rozhraním na straně serveru. Zmíněné rozhraní, o kterém si více prozradíme v kapitole č. 5.4 PHP, nám bude sloužit jako prostředník mezi databází a mobilní aplikací.

Jedna z mnoha výhod této knihovny spočívá v integraci s různými protokoly. Včetně následné práce s nezpracovaným řetězcem znaků, obrázky nebo s pro nás důležitým JSON. Javascript object notation je formát pro přenos objektových dat, který může být využit např. pro přenos z jednoho počítače na druhý. Taktéž je u tohoto dokumentu nutné vyzdvihnout jeho jazykovou nezávislost, tudíž je možné posílat i soubory s českou diakritikou [29].

Dalšími benefity knihovny Volley je poté rozdělení komunikace se serverem na takzvané požadavky, které se automaticky plánují, a těchto připojení může běžet současně až několik. Také je potřeba vyzdvihnout možnost prioritizace požadavků a možnost rušení celé části či jednotlivých bloků [30].

5.3 Databáze MySQL

S ohledem na vývoj aplikace bylo potřeba zvolit také vhodný prostředek k uchování dat. V našem případě byla zvolena databáze MySQL, která se jednoduchým způsobem propojuje na náš zvolený vývojový jazyk Java. Abychom však

mohli toto propojení uskutečnit, byl využit standard pro databázové připojení s názvem JDBC (Java Database Connectivity). Díky tomuto propojení bude komunikace s naším databázovým serverem snadná [31].

Na začátku je ještě nutné si popsat význam samotné databáze MySQL. Jedná se o otevřený systém řízení báze dat, který je jednak multiplatformní, jednak funguje na bázi relačního modelu. Z názvu této služby můžeme odvodit, jakým jazykem bude komunikace mezi klientem (aplikací) a serverem probíhat. Ke komunikaci je třeba využít jazyku SQL, přes nějž je možné skládat různé typy dotazů na server, a získávat tak potřebná data [32].

Výhody využití samotného serveru MySQL pak spočívá v jeho použitelnosti na různých zařízeních, jako je notebook či osobní počítač. Z toho vyplývá, že je tento typ databáze nenáročný a výkon jednotlivých komponent je možné nastavovat. Dalšími výhodami je pak neustálý vývoj této služby a nabídka různých užitečných funkcí pro správu databází. Nejvíce je kladen důraz na konektivitu, rychlost a zabezpečení, které nám MySQL poskytne [32].

Abychom mohli snáze provést testování na straně aplikace, je možné využít lokálního serveru XAMPP. Jedná se o jednu z nejpopulárnějších aplikací, která je zcela zdarma a poskytuje i další služby, jako jsou PHP, APACHE a MariaDB. V našem případě nám bude poskytnut prostor, v němž bude možné si vytvořit databázový server pro lokální potřeby [33, 34].

V rámci realizace našeho projektu bylo využito externího hostingu databáze, na kterém již je MySQL nainstalované. Pro následnou komunikaci s databází bylo nutno vytvořit sérii aplikačních programových rozhraní. Toto rozhraní je napsané v jazyce PHP, podrobněji viz další podkapitola.

5.4 PHP

Pro komunikaci s databází bylo třeba navrhnout prostředníka, který bude získávat instrukce z mobilní aplikace a ty bude následně přetlumočovat databázovému serveru a získávat z něj potřebná data. K tomu nám poslouží aplikační programové vybavení postavené na jazyku PHP.

PHP je široce využívaným skriptovacím jazykem, kde je tento kód spouštěn hlavně na straně serveru. Obecně se tyto skripty hojně využívají pro generování dynamických webových stránek a pro různou práci se soubory, jako je např. vytváření,

mazání, čtení a další. Nejdůležitější částí, kterou budeme v rámci této práce využívat, je práce s databází [35].

Výhodou tohoto prostředníka je zvýšená bezpečnost pro práci s daty. Většina dat se nejprve zpracovává na straně serveru a až poté jsou přeměrována v objektu typu JSON do mobilní aplikace, kde následně probíhá další zpracování. Díky tomuto typu zpracování jsme např. schopni heslo jdoucí z aplikace na rozhraní ověřit přímo na serveru a zpátky vrátit jen hodnotu, zda se přihlášení podařilo, anebo bylo neúspěšné. Pro samotnou komunikaci je pak použita metoda post [35].

6 Vývoj

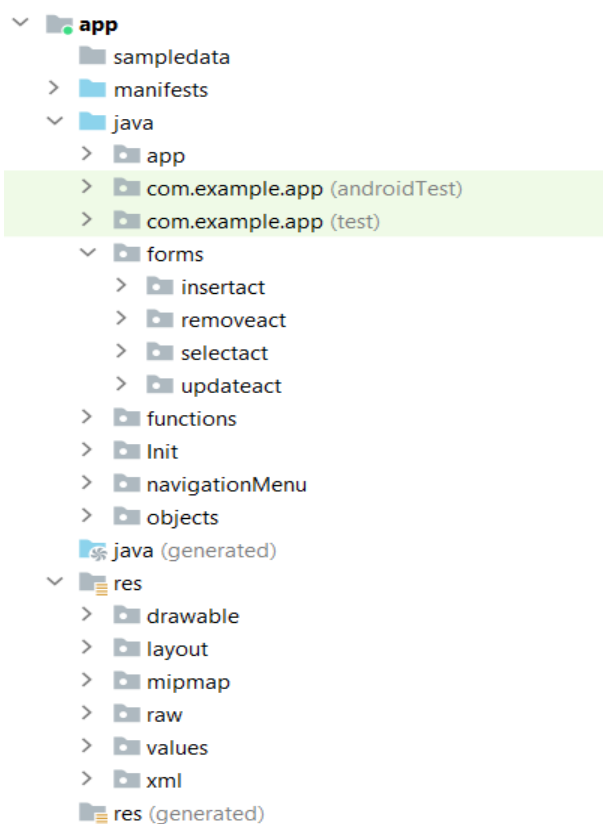
V této kapitole je popsán vývoj backendu i frontendu mobilní aplikace pro využití na dětských táborech. V dílčích podkapitolách jsou následně popsány jednotlivé moduly a funkcionality včetně implementace potřebných podpůrných prvků na straně serveru. Nacházejí se zde různé informace, jak byly samotné technologie implementovány a jak vypadá samotná aplikace na reálném zařízení.

6.1 Mobilní aplikace

V této podkapitole je popsán a rozpracován vývoj mobilní aplikace včetně jejích základních funkcionalit. Složitější a podstatnější prvky mobilní aplikace jsou popsány v dalších kapitolách.

6.1.1 Organizace projektu

Nejprve je nutné si popsat samotnou organizaci balíčků a souborů. Vývoj projektu, a sice nové mobilní aplikace, byl rozdělen do několika podstatných složek, přičemž seznam složek je zobrazen na obrázku č. 3.



Obrázek 3: Organizace dat

Zdroj: Vlastní zpracování

Ve složce *manifests* nalezneme jediný soubor, ve kterém byla nastavena různá povolení, jako je např. použití kamery a další.

Ve složce *java* lze nalézt veškeré naprogramované třídy a aktivity rozdělené podle tematiky, s níž jednotlivé aktivity a třídy souvisí. V balíčku *app* nalezneme nejdůležitější aktivitu, která se volá při zapnutí aplikace. V této aktivitě je vytvořeno menu první úrovně a nachází se zde základní ošetření pro zjištění, zda má telefon jednotlivé funkce implementované i hardwarově. Ve zdroji dat *forms* nalezneme aktivity obsahující formulářové prvky. Ty jsou rozdělené do kategorií podle toho, jaký typ dotazů bude databáze využívat. V samotných souborech se pracuje již se surovými daty ve formuláři, jež si server zpracovává sám. Dalším balíčkem je *functions*, v němž nalezneme všechny důležité funkce, jako je kompas, mapa, čtečka RFID apod. Dalším balíčkem, který je nutné zmínit, je ve zdroji dat *navigationMenu*, přičemž zde se nachází ostatní úrovně menu.

V poslední složce *res* se nacházejí soubory, které úzce souvisí s veškerými zdroji, s nimiž bylo při vývoji aplikace pracováno. V této složce je možné najít samotné layouty, které nám definují, jak budou jednotlivé aktivity vypadat. Také se zde nachází balíček *drawable*, do kterého se ukládají veškeré obrázky. Ve zdroji dat *raw* se nacházejí soubory nutné pro práci s Morseovým kódem. Poslední balíček, který je nutné zmínit, je pak *values*. Zde se nacházejí XML soubory ulehčující definici např. barev nebo znakových řetězců.

6.1.2 Layouty

Využití rozložení různých prvků je nejdůležitější částí této aplikace. Je proto nutné přizpůsobit telefon tak, aby bylo příjemné se všemi funkcemi manipulovat, a co nejlépe využít omezený prostor mobilních zařízení. K tomu všemu nám poslouží takzvané layouty.

Layouty se tvoří v souborech XML. Důležitost těchto souborů spočívá v tom, že je zde možné graficky vytvořit téměř cokoliv. Vytváří se zde např. vzhled každé aktivity, kde do souboru XML můžeme libovolně přidávat rozdílné formulářové komponenty. Těmito prvky mohou být jednotlivé fragmenty, různá tlačítka, jiné interaktivní elementy nebo obalující prvky typu rozložení. Pro správné umístění komponent je důležité, abychom v naší aplikaci zajistili jednak vhodnou responzivitu, jednak

jednoduchou manipulovatelnost. Např. tlačítka musí být vycentrována nejlépe ve střední části mobilního zařízení. Příklad layoutu je zobrazen na obrázku č. 4.

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/imageView"
    tools:ignore="NotSibling"
    android:gravity="center_horizontal"
    android:layout_centerInParent="true"
    android:orientation="vertical">
    <Button
        android:id="@+id/showLeaderboard"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="5dp"
        android:text="Zobrazit žebříček"
    />
    <Button
        android:id="@+id/showCampGames"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:layout_marginBottom="5dp"
        android:text="Zobrazit táborové hry"
    />
    <Button
        android:id="@+id/showCompetitions"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="5dp"
        android:layout_marginTop="5dp"
        android:text="Zobrazit soutěže proti týmům"
    />
    <Button
        android:id="@+id/showChildren"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:text="Zobrazit ostatní děti"
    />
</LinearLayout>
```

Obrázek 4: Příklad layoutu

Zdroj: Vlastní zpracování

Na výše uvedeném obrázku je zobrazen příklad layoutu využitého v rámci této práce, na němž jsou vidět všechny podstatné prvky pro vycentrování obsahu a způsob, jakým lze řešit responzivitu. Abychom měli naši aplikaci responzivní, je důležité u jednotlivých značek využívat atributů *layout_width* a *layout_height* s hodnotami „*match_parent*“ nebo „*wrap_content*“. Tím zajistíme, že si telefon sám vypočítá, jak by měly být tyto prvky veliké.

Pro responzivitu je také vhodné u atributů využívat hodnoty s jednotkou dp, anglicky *density-independent pixels*, neboli hustotně-nezávislé pixely. Ty jsou navrženy tak, aby odpovídaly konzistentní fyzické velikosti UI prvků na různých zařízeních. Jeden dp je zhruba 1/160 palce, což odpovídá přibližně 160 dpi u telefonů. Řešení posunu obsahu na střed se řeší atributy *gravity* a *layout_centerInParent*.

6.1.3 Navigační menu

Samotná navigace je řešena takovým způsobem, aby byla k uživateli co nejprívětivější. V důsledku toho byla všechna tlačítka vycentrovaná na střed a rozdělena do více úrovní. Samotný vzhled se nastavil v souboru XML, který je k dispozici ve složce *res* a v balíčku *layout*. Samotný kód lze nalézt v souboru *main*, který se nachází v balíčku *app*. Ukázka hlavního menu vytvořené aplikace je zobrazena na následujícím obrázku:



Obrázek 5: Ukázka hlavního menu

Zdroj: Vlastní zpracování

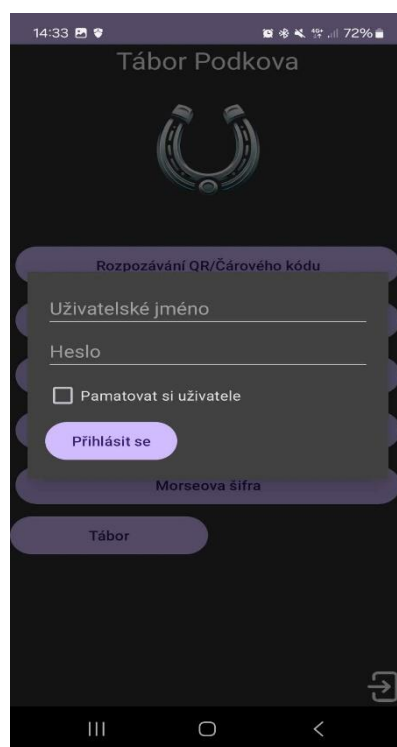
V prvotní fázi bylo potřeba propojit tlačítka s uvedeným ekvivalentem v layout souboru. To je řešeno pomocí této části kódu: `findViewById(R.id.qr)`. Zde probíhá nalezení prvku pomocí identifikačního názvu, který se musí shodovat s názvem v již zmíněném XML souboru.

Pro každé tlačítko je pak řešen způsob ošetření a zjištění, zda telefon uživatele obsahuje jednotlivý hardware a zda je zapnutý. Např. u tlačítka čtení RFID je zjišťováno, zda mobilní zařízení disponuje čipem čtoucím NFC, a zároveň se zjišťuje, zda je tato funkce zapnuta. U mapy se zjišťuje, zda existuje GPS lokátor a zda je ve stavu zapnutém. Poslední tlačítko, kde je toto ošetření využito, je u funkce kompasu. Zde se zjišťuje, zda telefon nabízí senzory typu akcelerometr, senzor měřící magnetické pole a gyroskop. Všechny tyto prvky jsou u kompasu důležité pro výpočet správného úhlu.

Tlačítka Tábor a Tábor – úpravy nám nabízí další úroveň menu, které nás intuitivně nasměruje pro práci s databází. Poslední tlačítko v pravém spodním rohu je vytvořeno pro uživatele, kteří se budou k systému této aplikace přihlašovat. Tím, že se přihlásí, získají přístup ke všem prvkům, které tato aplikace nabízí. Např. tlačítko Tábor – úpravy je pro nepřihlášené uživatele neviditelné. K tomu nám slouží přepis již známých hodnot v souboru XML. Můžeme to zajistit tím, že zavoláme na tlačítko `setVisibility(View.INVISIBLE)`. Jakmile je uživatel přihlášen, je potřeba pouze změnit hodnotu z neviditelného na `View.VISIBLE`, čímž bude tlačítko opět viditelné.

6.1.4 Přihlášení

Tlačítko přihlášení, které se nachází v pravém spodním rohu, je ošetřeno dvěma ikonami, jež se mění v důsledku toho, zda uživatel je, anebo není přihlášen. Pakliže se chce uživatel přihlásit, musí nejprve kliknout na tlačítko. Následně bude vyzván k zadání dat pro kontrolu, zda se v systému nachází. Ukázka menu pro přihlášení je zobrazena na obrázku níže:



Obrázek 6: Ukázka přihlášení

Zdroj: Vlastní zpracování

Samotné přihlášení je ošetřeno pro zadání prázdných hodnot, kdy je při jejich zaslání uživatel upozorněn, že musí zadat všechna data. Poté, co uživatel zadá všechny údaje, jsou data z formuláře poslána na server za pomoci knihovny Volley. Ta nám umožní vytvořit takzvaný `StringRequest`, který pošleme pro další zpracování na API.

Rozhraní API, anglicky *application programming interface*, je funkcionalita, která nám umožňuje výměnu informací mezi různými softwarovými aplikacemi.

Toto rozhraní je využito v rámci vývoje aplikace, avšak s ohledem na rozsah práce a standardní využívání při vývoji aplikací není v této práci podrobně popsáno. Následně už jen pracujeme s výsledkem, který nám byl zaslán, a zjistíme, zda byl uživatel v databázi nalezen, nebo ne. Pokud byl výsledek úspěšný, pak vytvoříme proměnnou token, který byl součástí zprávy doručené ze serveru, a vložíme ho do *SharedPreferences*. Toto umístění nám slouží primárně k ukládání a načítání malého množství dat, které jsou poté uloženy do interního úložiště zařízení ve formě XML. Jestliže měl uživatel zaškrtnuto pamatovat si uživatele, pak se do tohoto úložiště ještě přidá atribut *remember* s hodnotou *true*.

6.1.5 Řešení práv

V aplikaci jsou práva rozdělena podle toho, zda se uživatel přihlásil nebo zda byl uložen token se správnou hodnotou atributu *remember*. Přihlášená osoba poté získá možnost využívat veškerých funkcí v rámci aplikace. Naopak osoby, které touto formou autentizace neprošly, budou mít některé funkce vypnuté.

Nepřihlášené osoby pak budou mít tato omezení:

- Tlačítko, které přesměrovává na práci s databází, bude neviditelné. Nebude tedy možné mazat, přidávat a aktualizovat data v databázi.
- Tlačítko, které umožňuje rovnou z mapy přidat tlačítko přímo do databáze, bude v aktivitě nepřístupné.
- Funkce RFID bude poskytovat jen čtení, zatímco zápis a zobrazení velikosti NFC čipu budou možné jen pro přihlášené osoby.

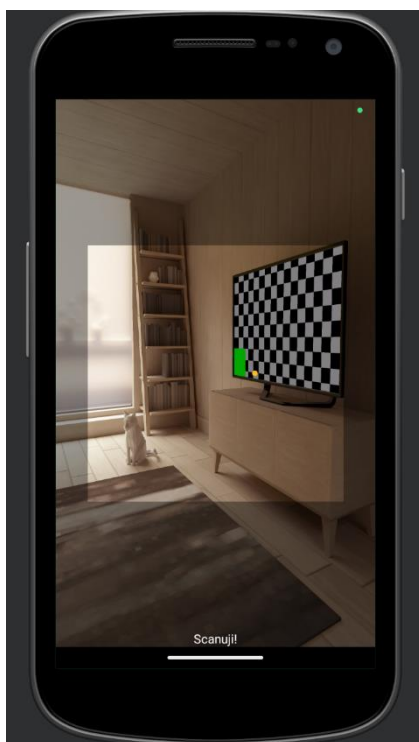
Aby bylo možné v ostatních aktivitách rozpoznat, zda je uživatel přihlášen, bylo využito zpětného získávání již zmíněného *SharedPreferences*.

6.2 Funkcionalita

V této kapitole jsou popsány všechny funkcionality vyvíjené mobilní aplikace včetně způsobu řešení a implementace podstatných prvků. Současně jsou u dílčích funkcionalit přiloženy grafické obrázky, které byly vytvořeny prostřednictvím mobilního zařízení po dokončení vývoje aplikace.

6.2.1 Rozpoznání QR kódu

Samotná funkcionality začíná v bodě, kdy proběhne přesměrování z tlačítka, které se nachází na hlavním menu. Čtení tohoto kódu probíhá zadní kamerou, kde aktivita vyčkává na zaregistrování čárového nebo QR symbolu. K tomuto kroku jsme schopni se dopracovat díky knihovně *ZXING*, která nám tuto funkci poskytuje. Na obrázku č. 7 je zobrazena ukázka skenovacího okna.



Obrázek 7: Ukázka skenovacího okna

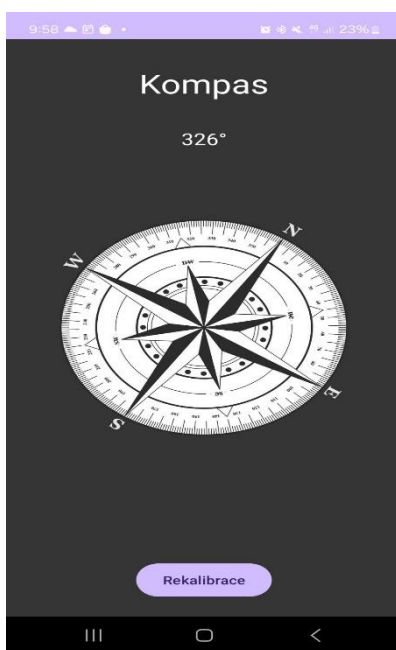
Zdroj: Vlastní zpracování

Po naskenování jedné z vybraných značek je uživateli umožněno výsledek po načtení uložit do vlastního zařízení nebo tento výsledek odkliknout druhým tlačítkem. Díky tomu mohou uživatelé v aplikaci uchovávat soubory s výsledky, na jejichž základě je možné vytvořit některé hry nebo na jiné navázat. Jako příklad lze uvést *šifrovanou*.

Doposud bylo podstatou této hry pomocí různých papírků rozluštit šifru, kterou tam někdo zanechal. Na těchto částech papírků byla slova s pořadím ve větě, přičemž bylo třeba šifru poskládat v co nejlepším čase. Díky této aplikaci je možné slova nahradit různými značkami, které budou muset soutěžící nejprve načíst a až poté skládat. V rámci této funkcionality je tedy možné využívat stávající hry, které jsou praktikovány na dětských táborech, a to v digitální formě prostřednictvím nové mobilní aplikace.

6.2.2 Kompas

V této kapitole je rozpracován popis funkcionality a samotná implementace kompasu v naší aplikaci za využití moderních technologických principů, jež nám poskytují co nejpřesnější hodnoty výpočtu pro otáčení kompasu. K dosažení této přesnosti bylo nezbytné využít všechny dostupné senzory, které mobilní telefon nabízí, včetně senzoru měřícího magnetické pole, akcelerometru a gyroskopu. Specificky, aplikace využívá `TYPE.ROTATION_VECTOR` pro zjednodušení výpočtu správného azimutu a otočení obrázku kompasu o potřebný úhel, což umožňuje efektivnější a přesnější orientaci uživatele. Ukázka funkcionality pro využívání kompasu je zobrazena na obrázku níže:



Obrázek 8: Ukázka kompasu

Zdroj: Vlastní zpracování

Taktéž nesmíme opomenout skutečnost možného ošetření chybovosti těchto senzorů – takovému procesu se říká recalibrace. Tento proces zahrnuje pohyb telefonu ve vzduchu, kde je nutné tímto zařízením opisovat tvar čísla osm po dobu alespoň třiceti vteřin. Ačkoliv samotné ošetření chybovosti není nezbytné řešit přímo v kódu aplikace, je zásadní uživateli alespoň poskytnout informace o potřebě recalibrovat. Tato potřeba je vyřešena tlačítkem *Recalibrace*.

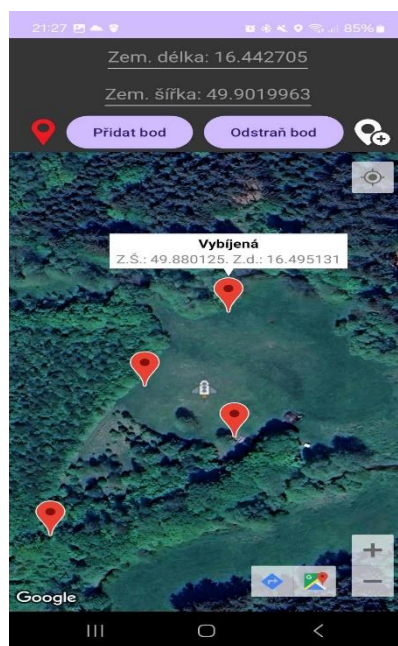


Obrázek 9: Ukázka recalibrace
Zdroj: Vlastní zpracování

Na obrázku č. 9 je jasně ilustrováno, kdy a za jakých podmínek by měl uživatel provést seřízení senzorů. Přesnost každého ze senzorů může dosahovat jedné ze tří úrovní: vysoké, střední nebo nízké. Samotná přesnost je nám k dispozici v rámci metody *OnAccuracyChanged()*, která umožňuje přiřadit numerickým hodnotám přesnosti jejich slovní ekvivalenty. Na základě těchto úrovní přesnosti se mění text, který je posazen hned pod tyto hodnoty. Uživateli je recalibrace doporučena již v případě, kdy jedna z hodnot vykáže odchylku v měření. Pakliže bude hodnota některého z prvků nízká, je potřebné provést recalibraci, čímž opravíme špatné hodnoty kompasu.

6.2.3 Mapa

Podstata využití interaktivní mapy v této práci spočívá ve využití bodů, které na ní mohou být umístěné a sloužit tak k navigaci nebo zábavným účelům. Tyto body mohou např. ukazovat dětem místa, na nichž se budou konat budoucí hry či akce. Další aplikace těchto bodů se nabízí v rámci hry hledání pokladu, kde budou mladí účastníci postupně získávat souřadnice a cestovat mezi nimi, dokud nedosáhnou cílového místa. Funkcionalita pro mapy je zobrazena na obrázku č. 10:



Obrázek 10: Ukázka mapy

Zdroj: Vlastní zpracování

Samotná mapa obsahuje hned několik zásadních funkcionalit. První z nich je zobrazení podstatných bodů, které se získávají z databáze. Abychom tyto body zobrazili, je nutné kliknout na ikonku v levé části telefonu, která se mění podle toho, zda body jsou, nebo nejsou zobrazeny.

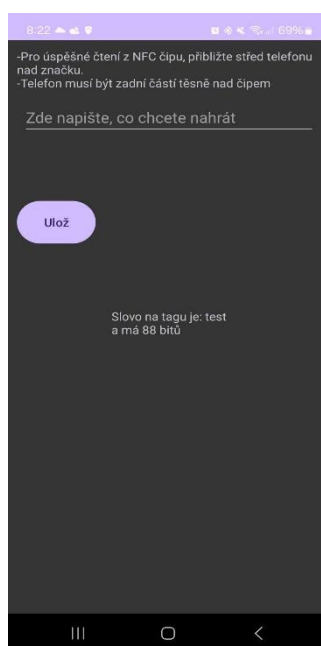
Pro dočasnou práci s body, např. u výše uvedené hry, je vytvořen základní formulář pro zadání zeměpisné šířky a délky s nápovědou, získanou z pozice mobilního zařízení. Při stisknutí tlačítka *Přidat bod* nejprve proběhne kontrola, zda uživatel zadal neprázdnou hodnotu. Pakliže byly tyto souřadnice zadány správně, přidá se bod nejprve na mapu a poté do proměnné *markersStack*, která nám tyto body ukládá do LIFO fronty za účelem budoucího mazání. Toto smazání probíhá při kliknutí na tlačítko „*Odstraň bod*“, kde bude smazán právě ten bod, který byl přidán jako poslední.

Poslední tlačítko, které se nachází v pravé části rozhraní, je dostupné jen pro přihlášené osoby. Jeho hlavní funkcí je při kliknutí přenos zadaných souřadnic do

databáze za účelem dalšího zpracování. Abychom ulehčili práci uživateli se zadáváním souřadnic, byla přidána funkcionality, kde již není nutné tyto souřadnice psát, ale je možné kliknout přímo do mapy a získat tak souřadnice tímto způsobem.

6.2.4 Čtení a zápis pomocí technologie RFID

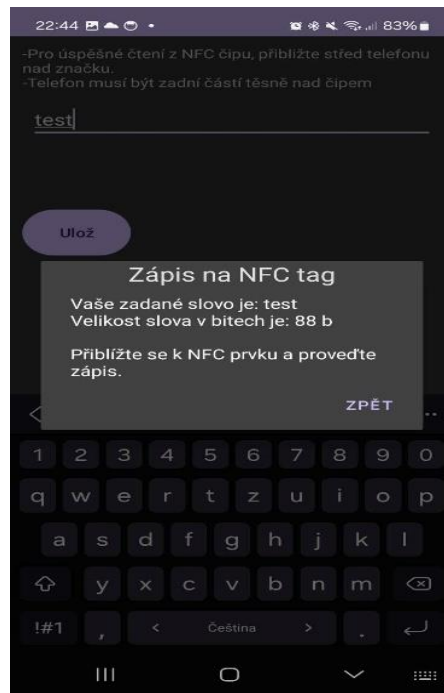
V této sekci je popsána implementace a vzhled funkcionality pro práci s NFC čipy. Samotný kód se nachází v aktivitě *RFIDActivity*, kde je na začátku nejprve nutné ověřit typ uživatelského účtu. Ve chvíli, kdy je uživatel přihlášen, je mu zobrazeno tlačítko „Ulož“. Aby však mohl provést uložení na čip NFC, je nezbytné, aby zadal text, který by rád uložil na tuto technologii. Samotná ukázka grafického řešení okna s RFID je zobrazena na obrázku č. 11.



Obrázek 11: Ukázka RFID

Zdroj: Vlastní zpracování

Další důležitou funkcionalitou je informování přihlášeného uživatele o tom, kolik místa bude požadované slovo na ukládaném zařízení zabírat. Tím je ošetřen případ, kdy budou např. vedoucí tábora na již zmíněné čipy zapisovat data, přičemž čipy budou mít omezené uložení. V takovou chvíli si uživatel může za předpokladu, že zná velikost čipu, ověřit, zda se tato data na dané uložení skutečně nahrají.

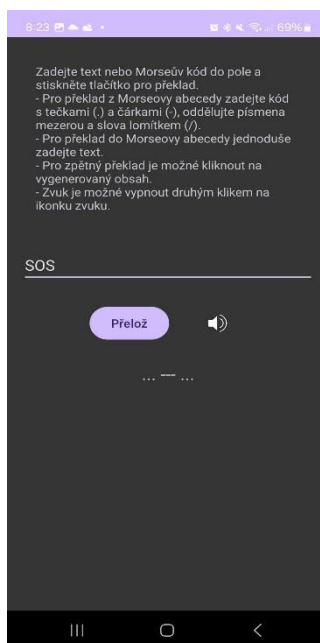


Obrázek 12: Ukázka zápisu RFID
Zdroj: Vlastní zpracování

Na obrázku č. 12 si lze všimnout výsledku po kliknutí na tlačítko „Ulož“. Nejprve se slovo nahraje do proměnné v aplikaci a poté na uživatele vyskočí okno s dalšími instrukcemi. Po přiblížení k NFC čipu se soubory nahrají a uživatel bude informován o tom, zda byl tento přenos úspěšný, či nikoliv.

6.2.5 Morseova šifra

Poslední podstatnou funkcionalitou je Morseova šifra, která se nachází v aktivitě *MorseCode*, přičemž tato funkce funguje na principu načítání dat ze souboru *morsecode.txt*, kde se nachází celá abeceda včetně písemného ekvivalentu. Samotné načítání se poté ukládá do dvou proměnných typu *HashMap*, která funguje na principu klíče a hodnoty. Pro správný překlad pro oba směry je potřeba tyto proměnné využít dvě. Jedna z nich uchová data, kde klíčem bude písmeno a hodnotou Morseův ekvivalent, zatímco druhá má tento princip opačný. Na obrázku č. 13 je zobrazena funkcionalita Morseova kódu.



Obrázek 13: Ukázka překladu Morseova kódu

Zdroj: Vlastní zpracování

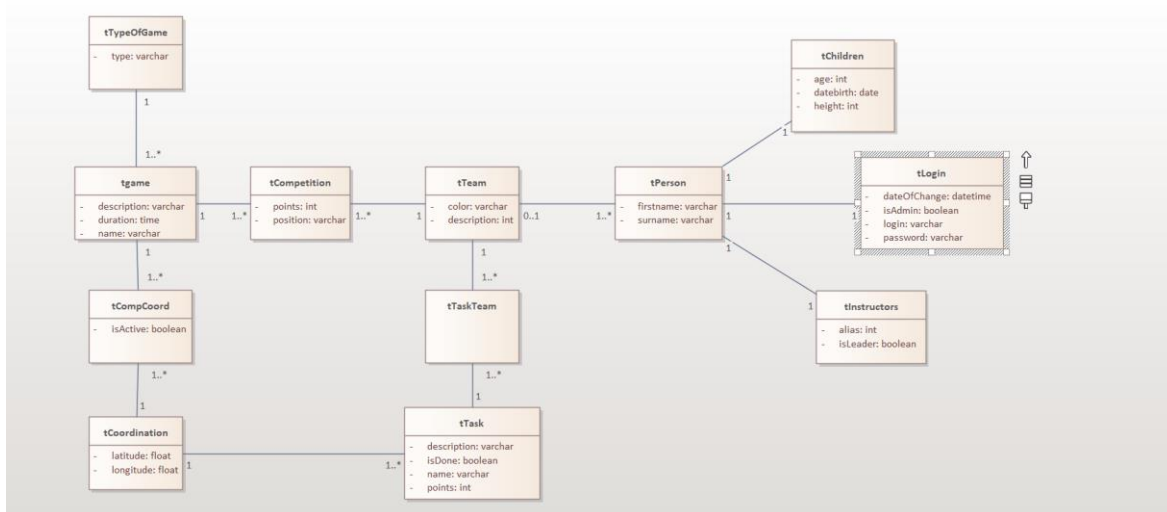
Překlad pak funguje již jednoduše, kdy se při kliknutí na tlačítko „Přelož“ nejprve zjistí, zda je něco zapsané, a poté o jaký text se jedná. Jestliže se jedná o slovo, pak bude přeloženo na morseovku. Pokud uživatel zadal morseovku, bude výsledkem přeložené slovo. Pro zpětnou kontrolu je možné kliknout na výslednou hodnotu, která se nám automaticky propíše do okna se zadávaným textem, a můžeme znovu provést překlad.

Jestliže je výsledkem Morseův kód, je možné provést přehrání zvuku. To provedeme kliknutím na ikonku zvuku, která se tímto aktem změní na zelenou barvu a začne s přehráváním. Tato problematika byla vyřešena pomocí vlákna, jež obsahuje cyklus pro zpracování tohoto řetězce, který v sobě ověřuje, zda se jedná o tečku či čárku. Jednotlivé výsledky se poté předávají metodě *playDotSound* nebo *playDashSound* pro další zpracování zvuků. Tyto zvuky jsou následně vyřešeny pomocí třídy *SoundPool*, která je pro krátké zvukové stopy nejvhodnější.

6.3 Serverová část

V této kapitole si v první části představíme návrh databáze, která nám slouží jako podpůrný prvek pro plnou funkčnost naší aplikace. V části druhé je popsáno možné řešení problematiky komunikace s databází tak, aby se velká část zpracování přesunula na stranu serveru. V poslední části je poté znázorněn princip, který se zabývá obalením dat a přesunem těchto dat ze serveru do mobilní aplikace.

Databáze je vhodným řešením pro různá vylepšení a zkvalitnění práce s tímto typem softwaru. Děti si např. mohou v reálném čase zobrazovat výsledky soutěží, nebo jim lze naopak nabídnout i dobrovolné úkoly za účelem získání nějaké odměny. Ukázkou návrhu databáze zobrazuje obrázek č. 14.



Obrázek 14: Ukázkou databáze

Zdroj: Vlastní zpracování

Samotný návrh databáze lze rozdělit do několika částí:

1. Část správy uživatele – v této části se nachází tabulka *tPerson*, která se váže na tabulky *tChildren*, *tLogin* a *tInstructors*, kde jedna osoba může mít právě jeden záznam v těchto tabulkách. Osoba může být buďto dítětem, nebo instruktorem a poté mít svůj přihlašovací účet.
2. Část správy soutěží – v této části se nachází tabulky s týmy a hrami, přičemž je nutné, aby mezi nimi vznikla tabulka kontingenční, která vzniká pomocí vztahu M:N.
3. Část spravující hry – v této části je u tabulky *tGame* vhodné kategorizovat jednotlivé hry, a sice pro budoucí účely možného rozšíření aplikace. U her jsou dále definována pomocná tabulka *tCompCoord*, která využívá dat z tabulky *TCoordination*, kde se nachází souřadnice. To uživateli ulehčí hledání vhodných míst pro jednotlivé hry v budoucnu. Zde taktéž musí vzniknout kontingenční tabulka.
4. Část spravující bonusové úkoly – tato část je možným rozšířením, kterým by šla aplikace směřovat, a současně je možné v této části vytvářet úkoly typu uklidit prostor kolem stanů, naškrábat brambory a tak dále. Děti by pak mohly tyto úkoly vidět a být za splnění příslušného úkolu odměněny.

Aby bylo možné komunikovat s databází bezpečně a současně bez nutnosti zvyšování výkonnostních požadavků na straně klienta, je vhodné využít řešení zpracování dat na straně serveru. V této práci je daný problém řešen pomocí API, které je zmíněno i v rámci kapitoly PHP. Primárním úkolem tohoto rozhraní je zpracovat požadavek klienta na straně serveru a zpátky mu zaslat data, o které si zažádal. Vezmeme-li v úvahu např. přihlášení, pak data s heslem cestují jen jedním směrem, poněvadž nám zpátky stačí poslat jen hodnotu, zda bylo přihlášení úspěšné.

Principem těchto rozhraní je tedy zpracovat nějaký typ dotazu, který je pro klienta neznámý. Jediné, co klient na server posílá, jsou data získaná z formuláře. Poté, co jsou data přítomna na rozhraní, provede se potřebný dotaz. Výsledek tohoto dotazu se následně obalí do objektu JSON a posílá se zpět na klienta, kde proběhne zpracování tohoto objektu.

6.4 Integrace a správa dat v aplikaci

V rámci této aplikace je rozšiřující možností práce s databází přímo v mobilním zařízení. Nejenomže je možné si zobrazit některá předdefinovaná data z databáze, např. táborový žebříček týmu, nebo výsledky her, ale také je možné používat ostatní operace, jakými jsou např. přidávání, odebírání a aktualizování dat v databázi.

Samotné zobrazování je možné pro všechny uživatele využívající tuto aplikaci. Proces zobrazení probíhá tak, že jsou nejprve sesbírána potřebná data, nebo je odeslán požadavek přímo na rozhraní našeho serveru. Zde proběhne zpracování a následné odeslání do naší aplikace v podobě JSON. V aplikaci je pak tento JSON zpracován v odpovědi metody *StringRequest*, kde se data odbavují pomocí cyklu a posílají na následné zpracování do třídy *Table*. Tato třída obsahuje jednu přetěžovanou metodu *createTable*, do níž vstupují buďto tři, nebo čtyři parametry neboli sloupce a další důležité atributy, jako je barva či kontext. Výstupem této metody je poté pohled, který se předává jako řádek tabulce.

Přidávání, mazání a aktualizování dat v databázi je povoleno pouze přihlášeným uživatelům. Aplikace nabízí čtyři formuláře pro přidání: účastníka tábora, hry, výsledků her a souřadnic ke hře. Pro mazání jsou k dispozici formuláře umožňující odstranění dětí, výsledků her, her samotných a souřadnic. Pro aktualizace je k dispozici jeden formulář sloužící k aktualizaci stavu aktivní hry, kde rozhodujeme, zda se hra zobrazí na mapě, či nikoliv.

7 Závěr

Prvotním cílem této bakalářské práce bylo vytvoření mobilní aplikace pro dětské tábory, která by byla využívána mládeží do 15 let věku a jejími instruktory na dětských táborech nebo v rámci existujících sportovních spolků TJ Sokol aj. Před zahájením vývoje nové mobilní aplikace byla stanovena základní kritéria a funkcionality, které byly při vývoji integrovány.

Samotná aplikace byla v počáteční fázi vývoje otestována na dětském táboře Podkova, kde byly v rámci zkušebního testování vytvořeny tři funkcionality: mapa, kompas a čtečka QR kódu. Při testovacím provozu byl ze strany dětí a vedoucích tábora sledován pozitivní ohlas.

První z testovaných funkcí byla QR čtečka, která má děti naučit využívat právě tuto technologii. Samotné QR kódy se v dnešní době vyskytují v různých oblastech, které člověk využívá v rámci běžného života, a sice např. při placení nebo při rychlé instalaci nového softwaru skrze Google Play. Využívání QR kódů se však neustále posouvá, a tato funkce byla také využita v rámci táborové hry *stopovaná*, kde děti hledaly různá stanoviště, na nichž plnily rozličné úkoly.

Druhou funkcí, která byla testována v rámci zkušebního provozu, byl kompas, který měl za úkol dětem ukázat, že i digitalizací lze vytvořit fyzický nástroj, který lze využít např. během orientačního běhu. Tato funkce byla také stejně jako ta předchozí otestována v rámci již zmíněné táborové hry *stopovaná*, kde z jedné zprávy děti zjistily, kolik kroků musí ujít určeným směrem, aby se dostaly z jednoho stanoviště na druhé. Tam bylo jejich úkolem donést vajíčko na lžici zhruba 50 m, aniž by se rozbilo.

Třetí funkcí byla mapa, která měla dětem ukázat, jakým způsobem se lze orientovat nejen v přírodě, ale i v městské zástavbě. Díky této funkci a táborové hře *cesta za pokladem* se děti nejenom naučily práci se souřadnicovým systémem, ale také si na konci cesty vyzkoušely, jakým způsobem funguje detektor kovu, a zkusily daný poklad samy najít a vykopat.

Čtvrtou funkcí, která již nebyla otestována, byla funkce čtení RFID. Ta má za úkol přiblížit dětem práci s NFC čipem, který je dnes už téměř v každém mobilním zařízení. V rámci známých táborových her by šla tato funkcionality použít ve hře *šifrovaná*, kde by bylo úkolem zjištění jednotlivých slov nahraných na NFC čip. Poté by hra pokračovala až do té doby, než by děti rozluštily šifru a poskládaly slova do správného pořadí podle indicií.

Poslední funkcí, jež dosud nebyla v praxi využita, je Morseův kód, jehož účelem je seznámit děti s alternativním způsobem komunikace, který se liší od jejich rodného jazyka. Taktéž by tato funkcionalita mohla být součástí nějaké hry, např. výše uvedené *šifrované*.

V rámci této bakalářské práce byla tedy vytvořena inovativní mobilní aplikace se zajímavými funkcionalitami, které lze využívat na dětských táborech ze strany dětí a vedoucích. Neméně důležitou částí je atraktivnost této aplikace a její lehká manipulovatelnost, která by měla být přizpůsobena právě dětem. Na základě toho bylo nutné situovat tlačítka ve spodní polovině telefonu a taktéž učinit aplikaci co nejméně složitou, aby bylo lehce pochopitelné, co kde zmáčknout nebo co kam napsat. Rovněž byla aplikace doplněna o funkcionalitu práce s databází, kde jsou uživatelé schopni práce s těmito daty, a sice přes nadefinované formuláře.

Aplikace je vytvořena tak, aby mohla být dále vylepšována a rozšiřována o další funkcionality, např. v oblasti mapových podkladů. Dalším možným vylepšením může být vytvoření tlačítka a správa úkolů nebo rozšíření, jež by se nabízelo u Morseova kódu, a sice implementovat funkci, která by umožňovala odposlech zvuků a jejich převod do textové podoby v Morseově kódu.

Tato bakalářská práce a její cíle a přínosy představily inovativní přístup ke vzdělávání na dětských táborech, obohacující tradiční aktivity o technologický rozměr. Integrací inovativních funkcionalit v podobě čtečky QR kódů, kompasu, map, čtečky RFID a Morseovy šifry umožňuje dětem objevovat nové oblasti učení, ale také nabízí vedoucím širší spektrum možností při plánování her. Zpětná vazba získaná během prvotní fáze testování na táboře Podkova nám současně ukázala vysoký potenciál využití v praxi, neboť aplikace získala pozitivní reakce jak ze strany dětí, tak ze strany vedoucích.

Pro budoucí vývoj se nabízí rozšíření v oblasti nových typů dětských táborů, které již působí v zahraničí a na kterých se např. děti učí programovat roboty nebo samotné programování, což by mohlo dále zvýšit jejich vzdělávací a zábavní hodnotu. Vytvořená mobilní aplikace, jež vznikla v rámci bakalářské práce, nabízí novou cestu inovace vzdělávacích aktivit v oblasti neformálního vzdělávání, a to za využití nových technologií, čímž současně zvyšuje digitální gramotnost mládeže a dospělých.

8 Seznam použité literatury

- [1] MROCZEK-ŻULICKA, Aleksandra. The Young Educator in Tourism – a Case Study of Teenage Instructors of the Polish Scouting and Guiding Association. *Folia Turistica* [online]. 2018, 47, 101–135. ISSN 0867-3888, 2353-5962. Dostupné z: doi:10.5604/01.3001.0012.6215
- [2] MARTIN, Andrew, Geoff WATSON, Jan NEUMAN, Ivana TURČOVÁ a Lucie KALKUSOVÁ. Czech education in nature traditions. *History of Education Review* [online]. 2020, 49(1), 31–43. ISSN 0819-8691. Dostupné z: doi:10.1108/HER-04-2019-0008
- [3] ŠINDELÁŘ, Jan. *Odborný článek: Využití mobilních zařízení na táboře* [online]. [vid. 2024-01-31]. Dostupné z: <https://clanky.rvp.cz/clanek/c/Z/21833/vyuziti-mobilnich-zarizeni-na-tabore.html>
- [4] WAGNER, Amber, Jeff GRAY, Jonathan CORLEY a David WOLBER. Using app inventor in a K-12 summer camp. In: *Proceeding of the 44th ACM technical symposium on Computer science education* [online]. New York, NY, USA: Association for Computing Machinery, 2013, s. 621–626 [vid. 2024-01-31]. SIGCSE '13. ISBN 978-1-4503-1868-6. Dostupné z: doi:10.1145/2445196.2445377
- [5] ALHUMOUD, Sarah, Hend S. AL-KHALIFA, Muna AL-RAZGAN a Auhood ALFARIES. Using App Inventor and LEGO mindstorm NXT in a summer camp to attract high school girls to computing fields. In: *2014 IEEE Global Engineering Education Conference (EDUCON): 2014 IEEE Global Engineering Education Conference (EDUCON)* [online]. 2014, s. 173–177 [vid. 2024-01-31]. ISSN 2165-9567. Dostupné z: doi:10.1109/EDUCON.2014.6826086
- [6] *Neformální vzdělávání, MŠMT ČR* [online]. [vid. 2024-02-20]. Dostupné z: <https://www.msmt.cz/mladez/neformalni-vzdelavani-1>
- [7] *Neformální vzdělávání - gov.cz* [online]. [vid. 2024-02-20]. Dostupné z: <https://portal.gov.cz/informace/neformalni-vzdelavani-INF-103>
- [8] XU, Weiqi a Fan OUYANG. The application of AI technologies in STEM education: a systematic review from 2011 to 2021. *International Journal of STEM Education* [online]. 2022, 9(1), 59. ISSN 2196-7822. Dostupné z: doi:10.1186/s40594-022-00377-5
- [9] ARAUJO, Duarte, Micael COUCEIRO, Ludovic SEIFERT, Hugo SARMENTO a Keith DAVIDS. *Artificial Intelligence in Sport Performance Analysis* [online]. 2021. ISBN 978-1-00-316358-9. Dostupné z: doi:10.4324/9781003163589
- [10] WIEGERS, Karl E. a Joy BEATTY. *Software Requirements*. B.m.: Pearson Education, 2013. ISBN 978-0-7356-7962-7.
- [11] DAWID LIJEWSKI. *Best IDEs for Android Development in 2023* [online]. [vid. 2024-01-25]. Dostupné z: <https://www.netguru.com/blog/best-ides-for-android-development>

- [12] *Acceptance testing* [online]. 2024 [vid. 2024-01-25]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Acceptance_testing&oldid=1194211720#cite_note-9
- [13] *Software maintenance* [online]. 2023 [vid. 2024-02-05]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Software_maintenance&oldid=1189285206
- [14] FRANK, Jiří. *Lekce 1 - Android programování - Úvod do Android programování* [online]. [vid. 2024-02-05]. Dostupné z: <https://www.itnetwork.cz/tutorial-programovani-pro-android-v-jave-uvod>
- [15] What is Java? - Java Programming Language Explained - AWS. *Amazon Web Services, Inc.* [online]. [vid. 2024-02-08]. Dostupné z: <https://aws.amazon.com/what-is/java/>
- [16] *Java OOP (Object-Oriented Programming)* [online]. [vid. 2024-02-08]. Dostupné z: https://www.w3schools.com/java/java_oop.asp
- [17] Meet Android Studio. *Android Developers* [online]. [vid. 2024-02-12]. Dostupné z: <https://developer.android.com/studio/intro>
- [18] Gradle | Gradle vs Maven Comparison. *Gradle* [online]. 5. únor 2024 [vid. 2024-02-12]. Dostupné z: <https://gradle.org/maven-vs-gradle/>
- [19] Introduction to activities. *Android Developers* [online]. [vid. 2024-02-21]. Dostupné z: <https://developer.android.com/guide/components/activities/intro-activities>
- [20] Fragments. *Android Developers* [online]. [vid. 2024-02-21]. Dostupné z: <https://developer.android.com/guide/fragments>
- [21] Lifecycle.State. *Android Developers* [online]. [vid. 2024-03-01]. Dostupné z: <https://developer.android.com/reference/androidx/lifecycle/Lifecycle.State>
- [22] A Complete Guide to Learn XML For Android App Development. *GeeksforGeeks* [online]. 23. červenec 2021 [vid. 2024-02-24]. Dostupné z: <https://www.geeksforgeeks.org/a-complete-guide-to-learn-xml-for-android-app-development/>
- [23] What is the Java Library? • Vertex Academy. *Vertex Academy* [online]. 5. srpen 2016 [vid. 2024-03-26]. Dostupné z: <https://vertex-academy.com/tutorials/en/what-is-java-library/>
- [24] How to Read QR Code using Zxing Library in Android? *GeeksforGeeks* [online]. 13. leden 2021 [vid. 2024-03-26]. Dostupné z: <https://www.geeksforgeeks.org/how-to-read-qr-code-using-zxing-library-in-android/>
- [25] *journeyapps/zxing-android-embedded* [online]. Java. B.m.: JourneyApps. 22. března 2024 [vid. 2024-03-26]. Dostupné z: <https://github.com/journeyapps/zxing-android-embedded>

- [26] *Google Maps Platform Documentation | Maps SDK for Android | Google for Developers* [online]. [vid. 2024-03-26]. Dostupné z: <https://developers.google.com/maps/documentation/android-sdk>
- [27] Build location-aware apps | Sensors and location. *Android Developers* [online]. [vid. 2024-03-26]. Dostupné z: <https://developer.android.com/develop/sensors-and-location/location>
- [28] Volley Library in Android. *GeeksforGeeks* [online]. 15. listopad 2018 [vid. 2024-04-07]. Dostupné z: <https://www.geeksforgeeks.org/volley-library-in-android/>
- [29] *JSON Introduction* [online]. [vid. 2024-04-07]. Dostupné z: https://www.w3schools.com/js/js_json_intro.asp
- [30] Volley overview. *Volley* [online]. [vid. 2024-04-07]. Dostupné z: <https://google.github.io/volley/>
- [31] Java Database Connectivity with MySQL. *GeeksforGeeks* [online]. 1. leden 2022 [vid. 2024-02-25]. Dostupné z: <https://www.geeksforgeeks.org/java-database-connectivity-with-mysql/>
- [32] *MySQL :: MySQL 8.0 Reference Manual :: 1.2.1 What is MySQL?* [online]. [vid. 2024-02-25]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>
- [33] *XAMPP Installers and Downloads for Apache Friends* [online]. [vid. 2024-02-25]. Dostupné z: <https://www.apachefriends.org/>
- [34] IGNITIONMEDIA. XAMPP: A Popular PHP Development Environment. *Ignition Media* [online]. 22. listopad 2014 [vid. 2024-02-25]. Dostupné z: <https://ignitionmedia.com.au/programming/xampp-a-popular-php-development-environment/>
- [35] *PHP Introduction* [online]. [vid. 2024-04-07]. Dostupné z: https://www.w3schools.com/php/php_intro.asp

9 Přílohy

Data na přiloženém médiu

1. Databáze – obsahuje vzhled i skripty pro naplnění testovacími daty
2. API – obsahuje zdrojový kód v PHP
3. Android aplikace – obsahuje zdrojový kód aplikace
4. campApp.apk – jedná se o instalační soubor pro telefony Android s vytvořenou aplikací

Zadání bakalářské práce

Autor:	David Kozel
Studium:	I2100227
Studijní program:	B1802 Aplikovaná informatika
Studijní obor:	Aplikovaná informatika
Název bakalářské práce:	Edukativní aplikace pro dětské tábory
Název bakalářské práce AJ:	Educational application for children's camps

Cíl, metody, literatura, předpoklady:

Cílem práce je vytvoření aplikace pro mobilní zařízení, ve kterých by bylo možné pracovat s QR kódy, RFID a GPS. Software bude rozdělen na více částí s přístupem pro vedoucí, děti a dospívající mládež.

Bodově, co by aplikace teda měla umět:

1. Měla by mít schopnost vytvořit, exportovat a zároveň číst QR kód pro různé typy her na dětských táborech.
2. Měla by mít možnost zapisovat na RFID čipy a zároveň je číst pro další typy her. Na příklad: Najít v oblasti 5 RFID čipů a načíst si z nich hodnoty, zde by bylo třeba 5 písmen a jejich úkolem by bylo složit z toho slovo, které by zároveň bylo heslem pro další postup v aplikaci.
3. Měla by mít schopnost rozlišovat, kdo se do aplikace přihlásil, jestli vedoucí/učitel, nebo dítě. Každý by měl jiná práva.
4. Tento bod je spíše teoreticky možný, protože by záviselo na telefonech/tabletech a jejich přesnou GPS lokací. Myšlenka je taková, že by se vytvořila mapa tábora a naimportovala by se tam zeměpisná šířka a výška. Popřípadě ohrazené google maps. Úkol by mohl vypadat následovně na mapě by se například zobrazilo červené kolečko, které by znázorňovalo poklad, oni by skrz GPS měli jít k pokladu a vykopat ho.

Edukativní aplikaci bych rád psal v Javě, ale není problém se kouknout i na jiný jazyk u Androidu jsem koukal, že se používá kotlin. Databáze by byla buďto MySQL nebo MSSQL.

Zadávací pracoviště:	Katedra informatiky a kvantitativních metod, Fakulta informatiky a managementu
Vedoucí práce:	Ing. Daniel Vondra
Datum zadání závěrečné práce:	26.1.2021