Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Trilog Universal

2017

Martin Mazáč

Vedoucí práce: Mgr. Martin Trnečka, Ph.D.

Studijní obor: Aplikovaná informatika, prezenční forma

Bibliografické údaje

Autor:	Martin Mazáč
Název práce:	Trilog Universal
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2017
Studijní obor:	Aplikovaná informatika, prezenční forma
Vedoucí práce:	Mgr. Martin Trnečka, Ph.D.
Počet stran:	42
Přílohy:	1 CD
Jazyk práce:	český

Bibliograhic info

Author:	Martin Mazáč
Title:	Trilog Universal
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2017
Study field:	Applied Computer Science, full-time form
Supervisor:	Mgr. Martin Trnečka, Ph.D.
Page count:	42
Supplements:	1 CD
Thesis language:	Czech

Anotace

Práce popisuje implementovaný tréninkový deník v podobě mobilní aplikace a webovou stránku, která slouží k prohlížení záznamů. Zabývá se technickým popisem a uživatelskou příručkou. Hlavní funkcionalitou mobilní aplikace je uchovávat a prohlížet záznamy.

Synopsis

The thesis describes implemented training diary in the form of a mobile app and a web page designed to browse the entries. It contains technical description and user manual. The main function of the app is store and browse individual entries.

Klíčová slova: Tréninkový deník; Trilog Universal; Webová aplikace

Keywords: Training diary; Trilog Universal; Web application

Děkuji Martinu Trnečkovi za obecné rady a ochotu vést práci.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce podpis autora

Obsah

Seznam obrázků

Seznam zdrojových kódů

1 Úvod

Pohybové aktivity slouží člověku k zlepšení jak fyzické, tak psychické stránky. Při různých intenzitách fyzické zátěže obecně člověku narůstá, nebo naopak klesá fyzická kondice, která se také označuje jako „fyzička“. Pro sledování této problematiky je zapotřebí pečlivě uchovávat záznamy o každém sportovním výkonu.

Nabízí se zde uchovávání informací v papírové podobě. Ovšem vypracování jednotlivých grafů, porovnávání výkonů a mnohé další zabere spoustu času. Uživatel aplikace Trilog Universal může jednoduchým způsobem naplňovat svůj tréninkový deník a tím mít přehled o jednotlivých záznamech. Dále může vytvářet libovolné sporty, u kterých chce uchovávat záznamy, porovnávat jednotlivé sportovní výkony mezi sebou, sledovat grafy za určité časové období.

Uživatel má také možnost sledovat aktivity pomocí webové aplikace.

1.1 Název Trilog Universal

Možná Vás napadne, co má zrovna společného sportovní aplikace s názvem Trilog Universal. Odůvodnění je však logické a přímočaré. Slovo *Tri* v obecném překladu znamená tři. *Log* zase znamená záznam nebo zaznamenat. Spojením těchto dvou slov dostaneme Trilog. Aplikace má v základním použití předvytvořené tři sporty a to plavání, kolo a běh. Z tohoto důvodů první část názvu Trilog.

Druhá část názvu obohacuje první. *Universal* znamená v obecném překladu universální. Tím, že jsou v základu vytvořeny tři základní sporty, aplikace má možnost libovolné sporty přidávat, mazat a upravovat. Spojením první a druhé části dostáváme již zmíněný název aplikace Trilog Universal.

1.2 Obsah projektu

Projekt Trilog Universal se skládá ze dvou základních částí. Stěžejní částí je nativní mobilní aplikace pro platformu Android. Této části se budeme věnovat podrobně v uživatelské dokumentaci, protože obsahuje mnoho funkcí, které je zapotřebí popsat.

Vedlejší části projektu je webová aplikace. Tuto část podrobně rozebereme v části technické dokumentace, protože využívá bohaté a zajímavé technologie.

1.2.1 Nativní aplikace

Aplikace slouží jako tréninkový deník, kde se odehrává hlavní funkcionalita. Běžný chod aplikace a využití hlavních funkcí je možný v režimu offline[1]. Při spárování s webovou aplikaci je zapotřebí mít zařízení online[2]. Na obrázku 1 je znázorněna úvodní obrazovka mobilní aplikace. Hlavní funkce aplikace:

- Přidávání/mazání záznamů

[1] offline režim - stav zařízení, které není aktuálně připojené k síti internet
[2] online režim - stav zařízení, které je aktuálně připojené k síti internet

- Přidávání/mazání sportů a atributů
- Zobrazování grafů v časovém období
- Porovnávání daného sportu v časovém období
- Porovnávání sportů v časovém období
- Editace hlavní obrazovky
- Shrnutí dat za časové období
- Detailní výpis záznamů

1.2.2 Webová aplikace

Responzivní webová stránka slouží k zobrazování dat, která jsou bezprostředně uložena v databázi. Uživatel mobilní aplikace v režimu online jedním tlačítkem odešle data z telefonu na webový server a tím webové aplikaci zpřístupní data. Dále také vykresluje grafy, vypisuje nejlepší výsledky a záznamy za určitě časové období. Webová aplikace také umožňuje exportovat soubor záznamů ve formátu `CSV` za dané časové období, nebo všechna data.

Obrázek 1: Úvodní obrazovka mobilní aplikace

2 Aplikace stejného typu

V následující sekci postupně představíme aplikace stejného typu. U jednotlivých aplikací budou popsány funkční záležitosti, vzhled, ovladatelnost, klady a zápory. Na konci sekce shrneme všechny aplikace a porovnáme je s aplikací Trilog Universal.

2.1 Endomondo

Sportovní aplikace, která nabízí základní funkčnost zadarmo. Rozšíření aplikace je zpoplatněno. Aplikace je podporována pro více platforem jako například iPhone, Android a Windows. Vylepšená PRO verze se od základní bezplatné liší například o vylepšení grafických prostředků a možnosti intervalových tréninků [5].

Aplikace je příjemně zpracovaná a přehledná. Nepředstavuje čistý tréninkový deník, ale sportovní aplikaci. Uživatel si před měřením sportovní aktivity zvolí jeden z již vytvořených sportů. Poté zvolí jeden z cílů aktivity. Mezi cíle patří:

- Nastavit limit vzdálenosti a času.
- Překonat kamarádův nejlepší výsledek.
- Překonat vlastní nejlepší výsledek.
- Základní měření vzdálenosti a času.

Jednotlivé záznamy uživatel může sledovat v záložce *History*. Po kliknutí na daný záznam se otevře detailní obrazovka s mapou zdolané trasy.

2.2 Sports Tracker

Aplikace je podporovaná pro platformy Android a iPhone. Aplikace má již předem vytvořené základní sporty, u kterých je možnost uchovávat záznamy. Měření aktivit probíhá v následujících krocích. Uživatel na úvodní obrazovce, která slouží jako informační, spustí nové cvičení. V následujícím kroku zvolí sport nebo danou aktivitu a spustí se měření. Během měření uživatel vidí zdolanou vzdálenost, čas, aktuální rychlost, průměrnou rychlost a mapu. Jednotlivé záznamy lze sledovat v položce *Diary*. Výpis je nepřehledně zpracován, protože jsou zde vypsány veškeré záznamy o aktivitách, které jsou seřazené podle data. Uživatel se snadno poté ztratí ve vyhledávání. Po rozkliknutí daného záznamu se otevře detail, kde jsou vypsány základní informace. Detail také umožňuje vypisovat grafy, průběh aktivity a celkové shrnutí. Každá aktivita má možnost sdílení na sociální síti Facebook. Součástí aplikace je webová stránka.

2.3 Zdraví

Zdraví je aplikace určena pro operační systém iOS. Od určité verze operačního systému je automaticky nainstalována. Aplikace poskytuje uchovávaní záznamů o aktivitách, výživě a spánku. Aktivity jsou přidávány automaticky, avšak dají se přidávat také manuálně. Mezi základní aktivity patří chůze a běh, kroky a vystoupaná patra. Grafické zpracování aplikace je velmi příjemné a přehledné. Funkcionalita aplikace uživateli umožňuje vytvořit novou aktivitu podle nabídky sportů. Škála sportů je poměrně veliká, avšak nepokrývá veškeré sporty. Jednotlivým sportům také nejde přiřadit libovolný atribut na měření. Aktivity podle data vykreslují grafy a také detailní výpis záznamů.

2.4 Shrnutí a motivace

Sportovní aplikace, které slouží jako tréninkový deník, je na internetu nespočet. Motivaci pro vytvoření aplikace Trilog Unversal byla v hlavní řadě přehlednost a jednoduchost. Spousta aplikací tohoto typu má v sobě plno užitečných funkcí, o kterých uživatel neví, protože se v aplikaci neorientuje.

Dalším důvodem je oprostit aplikaci od reklam a sociálních sítí. Veškeré aplikace v dnešní době kladou důraz na tyto záležitosti. Snaží se sdílet výsledky a záznamy na internetu. Uživateli se navíc v aplikaci často zobrazují různé nechtěné reklamy. Aplikace Trilog Universal se zaměřuje pouze na svou problematiku.

3 Využité technologie

3.1 Java

Programovací jazyk Java, který patří mezi jazyk na vyšší programovací úrovni je silně typovaný a objektově zaměřený. Java je vyvíjena společností Oracle a jazyk se překládá se do *Bytecodu*.

Zajištění, aby byl programovací jazyk Java nezávislý na dané platformě je vyřešeno následovně. Tím, že spoustu procesorů využívá jinou instrukční sadu se vývojáři Javy rozhodli vytvořit vlastní strojový kód, do kterého se bude Java překládat. Jedná se o *Bytecode*. Tento *Bytecode* je poté interpretován virtuálním strojem, nebo je přímo překládán do ISA[3] konkretního procesoru. Tímhle způsobem vznikl přenositelný kód, který není závislý na dané platformě.

Další výhodou programovacího jazyka Java je například automatická správa paměti v podobě *Garbage Collectoru*. Objekty jsou rozděleny do tří základních kategorií - Young, Eden, Permanated a Permaned generation.

Java byla v projektu zvolena z patřičných důvodů. Hlavním důvodem je, že je aplikace vytvářena pro platformu Android, která podporuje Javu.

3.2 Android

Svou strukturou se liší od ostatních mobilních operačních systému. Obecně pracuje v několika vrstvách. Každá z vrstev má svou funkčnost. Platforma vychází z jádra existujícího systému, jako je Linux. Nad touto vrstvou jsou uloženy veškeré knihovny v jazyce C/C++ a poté podle verze Androidu je na další vrstvě běhové prostředí Dalvik.

Tento operační systém je primárně využíván v mobilních zařízeních a tabletech. Můžeme jej také najít v hodinkách, televizích a dalších zařízeních.

Android API označuje verzi Androidu. Udává přesně jaké verze knihoven a programovacích prostředků můžou být při vývoji aplikace využity. Obecně platí, že čím je API vyšší, tím je verze Androidu novější a programátor má více prostředků při implementaci [9].

Minimální verze API v projektu je využita Android API 15.

3.3 HTML 5

Jedná se o základní stavební kámen webových stránek, bez kterého by nebylo možné stránku zkonstruovat. Jazyk HTML je značkovací jazyk, který slouží pro definici webové stránky. Jednotlivé značky se dělí nejčastěji do skupin párové vs. nepárové a blokové vs. řádkové. Novější verze jazyka HTML umožňuje přiřadit dané značce také atribut přes klíčové slovo *data* a ukládat si tak aktuální hodnoty [2]. Jednotlivým značkám je povoleno nastavovat id a třídu. Jazyk HTML se

[3]ISA - instrukční sada procesoru

nejčastěji používá v kombinaci s dalšími programovacími jazyky, jako jsou PHP a JavaScript. Mezi nejčastěji využívané značky patří:

- `<p>` - označení paragrafu
- `<div>` - označení blokové značky
- `<span>` - označení řádkové značky
- `<h1>` - označení největšího nadpisu

3.4 PHP 7.1

Jazyk vykonávající se na straně serveru, nikoli na straně klienta. PHP je dynamický jazyk, tedy nemusí se uvádět datový typ, se kterým se pracuje. Zvláštnosti tohoto jazyka je, že všechny proměnné musí začínat symbolem $. Jazyk je objektový a interpretovaný. Ke spuštění skriptu je zapotřebí webový server, například Apache. Jelikož se jedná o jazyk vykonávaný na straně serveru, musí být zajištěná možnost odesílání dat na server. PHP podporuje dva možné způsoby.

- Metoda GET - za URL adresu je vložen symbol ? a následné jednotlivé parametry a k nim příslušné hodnoty.
- Metoda POST - způsob zasílání dat v těle zprávy.

Aplikace byla vyvíjená pomocí PHP verze 7.1.

3.4.1 PHP a implementace ORM systému

Jedná se o systém, který se dívá na data v databázi, jako by to byly objekty. Při vkládání dat do databáze, nebo výběru dat se vždy můžeme spolehnout na práci s objekty. V projektu každá třída ve složce `php/model` reprezentuje jednu tabulku, se kterou poté také pracuje. Práce je intuitivní, protože díky instancím třídy je evidentní, s jakou tabulku právě chceme manipulovat [7].

3.4.2 PHP a dotazovací systém do databáze

Při práci s perzistentními daty[4] je potřeba správně vytvořit dotaz, který aplikujeme na danou tabulku databáze. Složitější dotazy bývají nepřehledné a stávají se terčem chyb. Implementace třídy *QueryBuilder* programátora oprostí od těchto nechtěných a nežádoucích komplikací. Idea je taková, že si programátor vytvoří abstrakci nad dotazy do databáze. Při využívaní třídy *QueryBuilder* poté založí její instanci a volá ve správném pořadí její metody. Metody jsou navrženy tak, že pouze upravují hlavní proměnnou *dotaz*, kterou si třída udržuje. Při zavolání metody `get`, popřípadě `make` se dotaz vyhodnotí. Tím, že navíc třída *QueryBuider* využívá parametrizované dotazy, uživatel je chráněn před nejčastějšími útoky, jako je například SQL injection.

[4]perzistentní data - data přežívající výpočetní proces

3.4.3 AJAX

Anglická zkratka AJAX představuje Asynchronous JavaScript and XML [10]. Technologie AJAX se využívá v situaci, kdy potřebujeme dynamicky upravit obsah webové stránky, přičemž opakované načtení celého webu není žádoucí. Obecné využití metody AJAX pracuje v následujících krocích:

1. Internetový prohlížeč odešle požadavek na webový server.
2. Webový server požadavek vyhodnotí a zpracuje.
3. Webový server odešle výsledek požadavku zpátky prohlížeči.
4. Internetový prohlížeč obdrží data a následně je zpracuje.

3.5 SQL

Jeden z nejpoužívanějších jazyků pro práci s databázemi. Jazyk SQL byl využíván hlavně při testování a navrhování databáze, aby byly vytvořeny konkrétní vztahy mezi tabulkami v databázi. Jazyk SQL v sobě využívá třída *QueryBuilder*, který je napsán v jazyce PHP. Jazyk PHP implicitně podporuje jazyk SQL.

3.6 Databáze

Databáze slouží k uložení perzistentních dat. Databáze se skládá z jednotlivých tabulek. Tabulku poté tvoří záhlaví a tělo. Záhlaví tvoří jednotlivé atributy, které mají své typy a samotné tělo poté tvoří jednotlivé záznamy v tabulce. S tabulkou se manipuluje pomocí databázového jazyka, nejčastěji SQL ve spolupráci s jazykem PHP. Nejznámější typy databází:

- MySQL,
- PostgreSQL,
- SQLite

3.7 CSS

Kaskádové styly jsou v dnešní době nutností, bez kterých není možno se obejít, hlavně ve stylování a designu webových stránek. Kaskádové styly obecně pracují se značkovacím jazykem HTML. Každé značce lze přiřadit třídu nebo id, pomocí kterých lze poté daný element nastylovat. Obecně existují tři možné způsoby propojení jazyků HTML a CSS [4].

1. Vložení CSS souboru.
2. Definování kaskádových stylů v hlavičce.
3. Definování inline kaskádových stylů.

Kvalitní využití kaskádových stylů se odráží jak na designu webové stránce, tak na jeho přizpůsobitelnosti vůči zařízením s různými velikostmi úhlopříček. Zde byla využívá známá CSS knihovna Bootstrap.

3.7.1 Bootstrap knihovna

Knihovna Bootstrap je framework[5] založený na jazyku HTML, CSS a JavaScript. Umožňuje snadno budovat weby, které mají responsivní[6] chování. Knihovna má předvytvořených mnoho prvků, které programátorovi šetří čas.

Jedním z nejhlavnějších důvodů využití knihovny Bootstrap je responsivní chování. Knihovna využívá tzv. grid systém pro rozložení webu. Tento systém využívá rozložení šířky obrazovky na 12 sloupců podle procent. Podle potřeby je umožněno programátorovi pracovat i v jednotlivých rozlišeních.

Mimo jiné knihovna poskytuje plno dalších komponentů, jako jsou například tlačítka a jejich seskupování, navigace, thumbnaily[7], ukazatele průběhu a mnoho dalšího [1].

[5] framework - jádro softwaru, které poskytuje funkce pro práci s nim
[6] responsivní - přizpůsobitelný na úhlopříčku zařízení
[7] thumbnail - náhledový obrázek

4 Programátorská dokumentace

Projekt se skládá z nativní a webové aplikace, proto je zapotřebí strukturu projektu jednotlivě rozvést u každé části. Mobilní aplikace tvoří rozsáhlejší část.

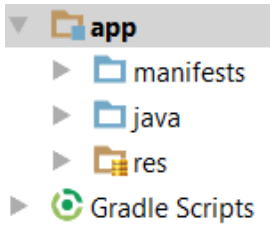
4.1 Mobilní aplikace

Na obrázku 2 je znázorněna struktura mobilní aplikace. Mobilní aplikace tvoří hlavní část projektu, proto v následující sekci budou vybrány nejdůležitější části, které budou důsledně rozebrány. Android aplikace obecně rozlišují mezi dvěma základními pojmy, co se týče vykreslování obrazovky.

- Fragment
- Activity

Activity představuje základní kámen k budování aplikace. Slouží také jako základní zprostředkovávatel komunikace při interakci s uživateli. Každá Activity má svůj životní cyklus a obslužné metody, které se v rámci životních stavech volají.

Naopak Fragment má významově jinou roli. Fragment představuje chování, nebo část chování uživatelského rozhraní v Activitě. Výhodou Fragmentů je možnost jejich kombinací v jedné Activitě a vybudovat tak několik plně funkčních částí, které mají vlastní životní cyklus. Tento cyklus se velmi podobá životnímu cyklu Activity [11].



Obrázek 2: Struktura mobilní aplikace

4.1.1 Soubor build.gradle

Představuje build systém[8]. Má jednoduchou a intuitivní syntaxi. Soubor se automaticky zakládá při vytvoření projektu, avšak lze jej jednoduše modifikovat a upravovat. Nese o projektu základní konfiguraci jako je id aplikace, minimální SDK verzi a verzi projektu. Soubor jednoduše dokáže vložit využívané externí i lokální knihovny.

[8]build systém - systém, který buduje aplikaci

4.1.2 Složka manifests

Obsahuje pouze jeden soubor se jménem `AndroidManifest.xml`. Jedná se o XML soubor, který musí obsahovat každá Android aplikace, předtím než se provede její spuštění. Soubor nese základní informace o celém projektu. Svým obsahem udává základní strukturu aplikace. Základní informace, které soubor obsahuje jsou:

- Java package, který je identifikátorem aplikace.
- Deklarace oprávnění, která aplikace vyžaduje a zasahuje tak do systému jako takového. Nejčastěji se jedná o přístupu aplikace do síti internet, WiFi, přístup do galerie nebo také komunikace mezi jinými aplikacemi.
- Definice nejnižší možné Android API, které aplikace potřebuje, aby byla spustitelná na přístroji.
- Udává celkovou strukturu aplikace.

4.1.3 Složka java

Složka java udává veškerou funkcionalitu aplikace. Jedná se o soubory s příponou `java`. Složka je dále rozčleněná dle funkcionality do dalších podsložek.

4.1.3.1 Složka java/fragments

Složka obsahující všechny základní Fragmenty. Tyto Fragmenty jsou napojeny na základní Activitu hlavní menu, která funguje jako rozcestník hlavní funkcionality. Jednotlivé Fragmenty se poté vykreslují na požadavek uživatele.

4.1.3.2 Složka java/sports

Složka, která obsahuje jednotlivé Activity. Od názvu složky zde můžeme očekávat veškeré Activity, které jsou nějakým způsobem vázány se sportem. Může se jednat o detail přidání záznamu, detail vymazání záznamu nebo také detailní výpis záznamů sportu.

4.1.3.3 Složka java/objects

Architektura aplikace je objektová, proto bylo potřeba projekt obohatit o základní objekty. Každý objekt má svou roli a podle potřeby je využíván. Mezi základní objekty patří například: *sport*, *atribut sportu*, *záznam*, *uživatel* a mnoho další.

4.1.3.4 Složka java/connection

Slouží k propojení mobilní aplikace s webovou. Soubory ve složce se také starají o synchronizaci databáze. O komunikaci s webovou aplikaci se starají tři soubory.

- `ConnectDatabaseClass.java`
- `RegisterUserClass.java`
- `SyncDatabasesClass.java`

Pro umožnění odesílání požadavků na webový server musí třídy pracovat jinak než ostatní soubory aplikace. Musí běžet v pozadí aplikace a tím neblokovat hlavní GUI vlákno. Tuhle funkčnost umožňuje Android třída *AsyncTask*, která v metodě `doInBackground` vyhodnocuje požadavky.

Soubory mají odlišnou roli. Soubor `ConnectDatabaseClass.java` obstarává základní přihlašovací údaje pro spojení s webovou službou jako jsou, typ protokolu HTTP nebo HTTPS, IP adresa serveru a přihlašovací údaje do databáze.

Soubor `RegisterUserClass.java` posílá na server požadavek o založení nového účtu. Pokud účet ještě neexistuje, je automaticky vytvořen.

Posledním souborem je `SyncDatabaseClass.java`, který slouží k synchronizaci databází. Všechny položky, které doposud nebyly synchronizovány se serverem budou odeslány a tím se zajistí synchronizace.

4.1.4 Složka res

Složka obsahující soubory s příponou `xml` a obrázky. Složka je dále členěna dle funkčnosti.

4.1.4.1 Složka res/layout

Zde se nachází pouze XML soubory, které pomocí XML syntaxe určují rozložení dané Activity nebo Fragmentu. Za zmínku by stálo podotknout, že každá Activita s sebou nese dva základní soubory. Prvním je vždy soubor začínající prefixem `activity_` a dále následuje jméno souboru. Tento typ souboru slouží jako hlavička Activity a jsou zde inicializovány věci jako je *toolbar*, *floating point button* a další. Druhým typem je soubor začínající prefixem `content_` a následuje jméno souboru. Tento soubor určuje veškeré rozložení dané obrazovky.

Při vytvoření konkretní Activity, například sport, se tedy vygenerují ve složce *layout* dva následující soubory:

- `activity_sport.xml`
- `content_sport.xml`

4.1.4.2 Složka res/drawable

Jedná se o obecnou složku objektů a položek, které je možné vykreslit. Složka obsahuje obrázky, které se používají v aplikaci, dále obsahuje soubory typu XML. Tyto soubory můžou reprezentovat typy stylů, ikony a obrázky.

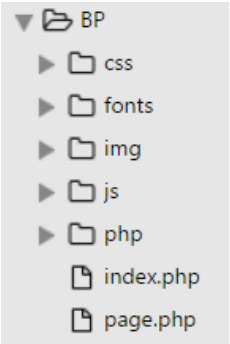
4.1.4.3 Složka res/values

Android dovoluje globálně definovat často používané hodnoty. Složka opět obsahuje soubory typu XML. Například soubor `colors.xml` umožňuje definovat nejčastěji používané barvy. Naopak soubor `strings.xml` zase uchovává nejčastější používané řetězce, jako jsou názvy jednotlivých Activit, popis tlačítek a další.

Dostupnost hodnot, které jsou pojmenovány, je pro programátora jednoduchá. Při uvedení cesty ke složce *values* a danému souboru se můžeme poté dotázat na danou hodnotu, kterou si definujeme pod námi vytvořeným jménem. Další výhodou je, že při změně hodnoty nemusíme zasahovat do více souborů, ale stačí nám hodnoty upravit pouze na jednom místě.

4.2 Webová aplikace

Obrázek 3 zobrazuje strukturu webové aplikace.



Obrázek 3: Struktura webové aplikace

4.2.1 Kořenová složka

V kořenové složce jsou uloženy všechny části projektu. Hlavní soubory jsou poté `index.php` a `page.php`. Soubor `index.php` slouží jako přihlašovací formulář do aplikace a soubor `page.php` poté zobrazuje veškerá data, která jsou uložena v databázi přihlášeného uživatele.

4.2.2 Složka php

Složka, která obsahuje nejvíce souborů. Zde je uloženo také jádro celé aplikace. Daná složka obsahuje další podsložky *core*, *ajax*, *model*, *synchronization* a *data-export*.

Soubory ve složce PHP jsou obslužné soubory, které se starají o chod, zabezpečení, synchronizaci a funkčnost celé aplikace.

- `initialize-select` - soubor, který provede inicializaci perzistentně uložených dat z databáze.
- `login-script/logout-scrip` - soubor k přihlášení/odhlášení z webové aplikace.

4.2.2.1 Složka php/core

Následující složka, která je jádrem celého projektu. Obsahuje tři základní soubory, které zprostředkovávají komunikaci s databázi. Soubor `Database.php` obsahuje pouze dvě metody. Metoda `bind`, která má jeden parametr pole s konfigurací připojení do databáze. Druhou metodou je `getInstance`, která vrátí danou instanci třídy *Database*.

Soubor `configuration.php` obsahuje základní parametry o databázi, které jsou potřeba pro připojení. Soubor je schválně oddělen zvlášť, aby uživatel měl možnost jednoduše změnit parametry podle svého nastavení.

Soubor `bootstrap.php` slouží k propojení souboru `configuration.php` s `Databse.php`. Jednoduše jsou soubory importovány do jednoho souboru a konfigurace je poté vyhodnocena pomocí metody `Databse::bind()`.

4.2.2.2 Složka php/model

Jednotlivé tabulky databáze jsou reprezentovány vlastní třídou. Všechny tyto třídy jsou obsaženy ve složce `php/model`. Mají jednoho společného předka a tím je abstraktní třída `Table.php`. Myšlenka implementace takového systému je přímočará. Jednotlivé objekty obsahují instanci třídy *QueryBuilder* a vytváření dotazu nad určitou tabulkou je poté jednoduché. Data z tabulky databáze jsou převedeny do objektů dané třídy. Programátor poté zpracovává data podle potřeby, se kterýma pracuje jako s objekty. Jedná se implementaci vlastního ORM systému.

4.2.2.3 Třída QueryBuilder

Zdrojový kód 1 znázorňuje implementaci třídy *QueryBuilder*.

```
1  class QueryBuilder
2  {
3    private $query;
4    private $database;
5
6    public function __construct() {
7      $this->database = Database::getInstance();
8      $this->query = "";
9    }
10
11   public function select( ...$attributes )
12   {
13     $this->query = "SELECT ";
14     $this->query .= implode($attributes, ", ");
15     $this->query .= " FROM ";
16     return $this;
17   }
18
19   public function where( $attribute, $relation, $value ) {
20     $this->query .= " WHERE {$attribute} {$relation} '{$value}'";
21     return $this;
22   }
23
24   public function table ($table) {
25     $this->query .= $table;
26     return $this;
27   }
28
29   public function get()
30   {
31     $statement = $this->database->prepare($this->query);
32     $statement->execute($this->values);
33     return $statement->fetchAll(PDO::FETCH_CLASS);
34   }
35 }
```

Zdrojový kód 1: Příklad využití třídy QueryBuilder

Třída *QueryBuilder* slouží k budování příslušného dotazu do databáze. Vytváří jistou abstrakci pro programátora při práci s front-endem[9] a back-endem[10]. Jednotlivé metody slouží k postupnému vytváření dotazu podle potřeby. Metody jsou navrženy tím způsobem, že pokaždé vrací svůj vlastní objekt. Tím

[9] front-end - část webové stránky, kterou vidí běžný uživatel

[10] back-end - část webové stránky, která slouží k úpravě front-endu, označuje se také administrace

pádem metody můžeme dle libosti skládat za sebou tak, aby neporušovaly syntaxi dotazovacího jazyka. Aplikace třídy *QueryBuilder* je velice obdobná jako JavaScriptová knihovna jQuery [3].

Proměnné dané třídy jsou *dotaz* a *databáze*. Třída si v sobě udržuje stav dotazu, který postupně buduje. Tím, že metody vrací svůj vlastní objekt při použití metody je budování dotazu jednoduché. Každá metoda svým způsobem modifikuje a obohacuje daný dotaz.

Metoda `select` je začínající metoda, která inicializuje proměnnou *dotaz*. Metoda má proměnlivý počet argumentů *atributy*. Jednotlivé *atributy* slouží k vyjádření, na jaké atributy se do databáze chceme dotazovat.

Metoda `where` je metoda, která která jistým způsobem ovlivňuje sémantiku dotazu. Metoda má tři parametry. Prvním je atribut, druhým argumentem je relace a třetím je požadovaná hodnota.

Metoda `table` slouží k uvedení do dotazu s jakou tabulkou dané databáze chceme pracovat. Jejím jedním argumentem je tabulka.

Poslední metoda `get`, bez které by celá třída neměla smysl. Její chování je malinko jiné. Tahle metoda je přímým zprostředkovatelem komunikace. Její náplní je aplikovat aktuálně sestavený dotaz, který je uložený v proměnné `query`. Výsledky, které jsou vráceny, vrací jako pole objektů. Zdrojový kód 2 zobrazuje aplikaci třídy *QueryBuilder*, dále zdrojový kód 3 ukazuje překlad daného dotazu.

```
1 $query = new QueryBuilder;
2 $result = $query->select("*")->table("data")->where("id", "=", 5)->
    get();
3 var_dump($result);
```

Zdrojový kód 2: Aplikace třídy QueryBuilder

```
1 SELECT * FROM data WHERE id=5;
```

Zdrojový kód 3: Vytvořený dotaz aplikací QueryBuilderu

4.2.2.4 Složka php/ajax

Z názvu složky je patrné, že se jedná o načítání dat pomocí technologie AJAX. Tato složka slouží k inicializaci grafů pomocí programovacího jazyka JavaScript. Aby JavaScript mohl inicializovat grafy, musí obdržet data, který jsou uložena v databázi. Pro propojení JavaScriptu s jazykem PHP se v tomto případě využívá metody AJAX.

4.2.3 Složka js

Zkratka *js* v tomhle případě znamená JavaScript. Ve složce jsou obsaženy pouze soubory, které mají příponu `js`. Základním souborem této složky je soubor `main.js` a `index.js`. Soubor `index.js` inicializuje a pracuje s daty v souboru `index.php` a soubor `main.js` má stejné využití, ale v souboru `page.php`. Ostatní soubory této složky jsou využity jako knihovny v projektu, které obohacují design a usnadňují práci.

4.2.4 Zbylé složky

V projektu se ještě nachází složky jako font-awesome, css a img. Složka img obsahuje obrázky, které jsou obsaženy v aplikaci. Složka css obsahuje kaskádové stylování a známé font-awesome ikony.

4.2.5 Jednoduchá funkční rozšířitelnost

Před samotným zobrazením webové stránky jsou všechna perzistentní data inicializována v jazyce PHP v samostatném souboru. Soubor `initialize-select.php` je importovaný v hlavičce hlavního souboru. Rozšířit projekt o další funkčnost a vytvářet dotazy pomocí třídy *QueryBuilder* je zamýšlen právě v tomto souboru, který je oprostěn od hlavní funkcionality. Hlavní soubor, který s daty poté pracuje, pouze využívá naplněné proměnné daty, které se vypisují.

Uvedeme jednoduchý příklad rozšíření o novou funkcionalitu. Předpokládejme, že chceme přidat box s nejkratšími vzdálenostmi. Prvním zásahem do aplikace, který je zapotřebí, je v souboru `initialize-select.php`, kde je potřeba vytvořit pomocí třídy *QueryBuilder* správný dotaz do databáze a data následně uložit do proměnné s odpovídajícím názvem, například `minDistances`. Dalším krokem by bylo v souboru `page.php` na správném místě vytvořit daný box pomocí jazyka HTML. Daným tagům je zapotřebí přiřadit správné třídy, aby byl vzhled boxů stejný a neporušily se zásady uživatelského rozhraní. Poté pomocí PHP scriptů v podobě funkce `echo` stačí vypsat data, která jsme uložili do proměnné `minDistances`. Znázornění rozšíření funkcionality je zobrazeno ve zdrojovém kódu 4.

4.3 Navržení databáze

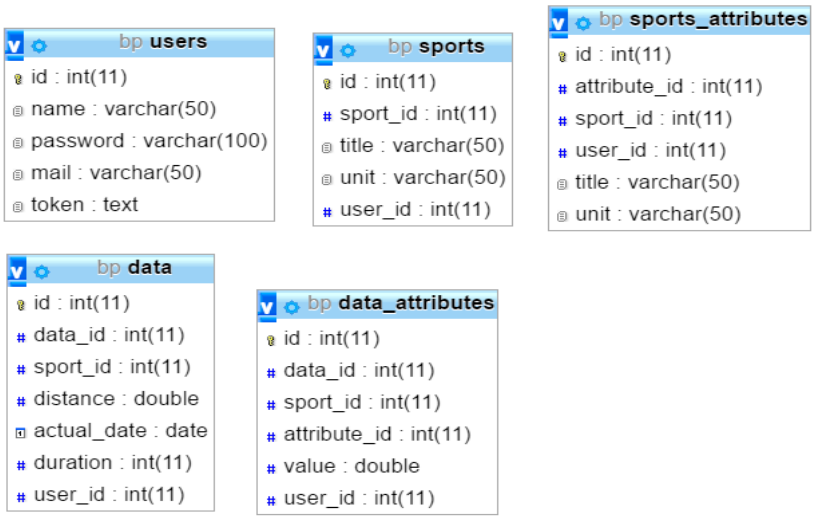
K perzistentnímu uložení jednotlivých záznamů, sportů a atributů je využita databáze. V případě mobilní aplikace je využita databáze SQLite, která umožňuje práci v režimu offline. Webová aplikace využívá typ databáze MySQL. Mobilní i webová aplikace má složení tabulek i celé databáze odlišené.

Pro tento rozdíl je celá řada důvodů. Mobilní aplikace musí uchovávat u každého záznamu, zda byl synchronizován s webovou aplikaci, či nikoli. Dalším důvodem je, že mobilní aplikace musí uchovávat konfiguraci připojení a přihlašovací údaje na webový účet. Naopak webová aplikace umožňuje víceuživatelský

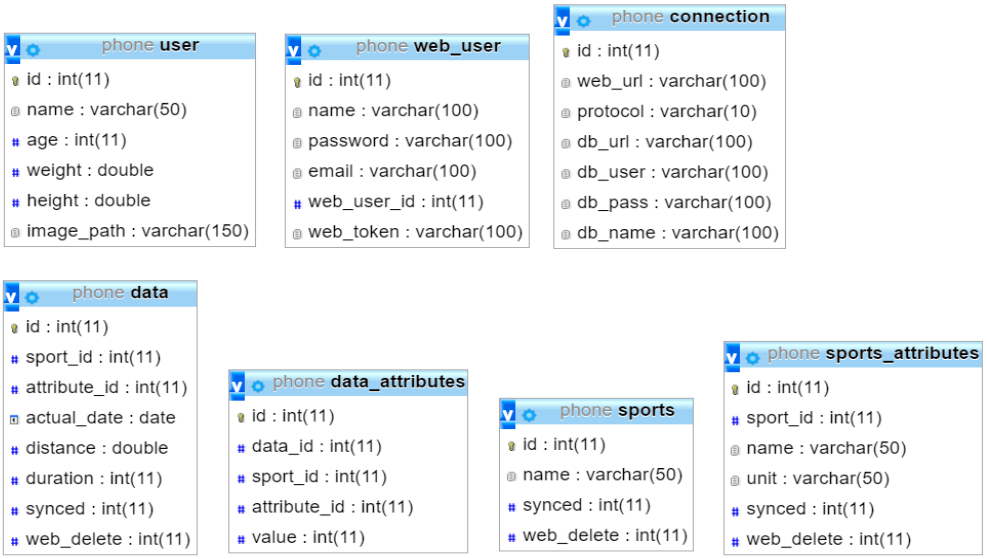
```
$minDistance = array();
$data = new Data();
$records = $data->select("sport_id", "min(distance) as min_dst")
            ->where("user_id", "=", $_SESSION['user_id'])
            ->groupBy("sport_id")->get();

foreach ($data as $record) {
  $minDistance[$record->sport_id] = $record->min_dst;
}
```

Zdrojový kód 4: Rozšíření funkcionality

přístup k aplikaci. Struktura webové databáze je vyobrazena na obrázku 4 a mobilní struktura databáze na obrázku 5.



Obrázek 4: Struktura databáze pro mobilní aplikaci



Obrázek 5: Struktura databáze pro webovou aplikaci

4.4 Zabezpečení

4.4.1 SHA-256

Ukládání hesel v plain textu[11] je velmi nebezpečné a útočník při zjištění hesla a přihlašovacího jména do aplikace se poté může vydávat za daného uživatele. Ukládání hesla proto musí byt zabezpečeno. Heslo je ukládáno pomocí hashovací funkce SHA-256, která z klasického hesla vytvoří 64-bitový řetězec. Kdyby se útočník dostal do databáze k heslu, dostane se k danému řetězci, který je nemožné dekódovat do původního řetězce [8].

4.4.2 Kontrola tokenu

Token představuje náhodně vygenerovanou sekvenci znaků, které dohromady tvoří řetězec. Při propojování aplikace s webovou službou se vygeneruje již zmíněný náhodný token, který si obě strany pamatují. Poté při každé synchronizaci databází se kontroluje hodnota tokenu. Pokud je správná, synchronizace se provede.

Metoda kontroly tokenu slouží v případě, že by útočník chtěl zaslat na server požadavek, ve kterém by nějakým nežádoucím způsobem modifikoval data nebo celou databázi obecně.

4.4.3 SQL injection

Jedná se o typ útoku, kdy útočník využije chyby programátora. Programátor při práci s jazykem SQL a obecně s jazykem, který pracuje s citlivými daty, musí dbát na bezpečnost. Nejčastější chybou jsou neošetřené vstupy, kde útočník může zadat část kódu v jazyce a ten se vyhodnotí [6].

Uvedení modelové situace na příkladu. Jsme na webové stránce, kde je přihlašovací formulář se zadáním hesla a jména. V případě, že by útočník do jména zadal hodnotu `admin - -`; by najednou SQL příkaz měl úplně odlišný význam. V jazyce SQL se píši komentáře pomocí znaků - - a heslo by již nebylo bráno v potaz. Obranou proti útoku SQL injection jsou parametrizované dotazy, které poskytují programovací jazyky. Možný typ útoku je znázorněn ve zdrojovém kódu 5.

```
1 SELECT * FROM users WHERE name= "$loginName" AND password="
    $loginPassword";
2 SELECT * FROM users WHERE name=admin; -- AND password="";
```

Zdrojový kód 5: SQL injection

[11] plain text - uložení hesla v dokumentové podobě, heslo není nijak zašifrováno

5 Uživatelská dokumentace

Následující sekce slouží jako manuál pro uživatele, aby byl obeznámen s použitím veškeré funkcionality jak aplikace, tak webové služby. Nejprve rozebereme hlavní část tedy mobilní aplikaci, poté se zaměříme na část webovou.

5.1 Nativní aplikace

5.1.1 Hlavní obrazovka

Po spuštění aplikace se objeví modrá obrazovka s názvem aplikace Trilog Universal s krátkou animací, ve které se zobrazí úvodní rozcestník představující navigaci. Navigace je velmi jednoduchá a obsahuje pouze dvě základní položky.

- Home
- Add data

Položka *Home* odkazuje na hlavní obrazovku aplikace a položka *Add data* zase na přidání nových záznamů do aplikace.

5.1.2 Hlavní menu

Menu celé aplikace je zde využito typické pro platformy Android. Slouží zde jako základní pevný bod aplikace, ze kterého lze ovládat veškerou funkcionalitu celé aplikace. Položky menu:

- Home
- Add data
- Delete data
- Activities summary
- Edit activities
- User profile
- Setting home
- Synchronization

5.1.2.1 Home

Představuje domovskou obrazovku. Slouží k obeznámení uživatele o jeho pohybových aktivitách ve třech různých způsobech v posledním týdnu a měsíci.

1. Přehled sportů - výpis jednotlivých sportů a k ním zdolané vzdálenosti za poslední týden a daný měsíc. Procentuální porovnání s časovým rozmezím aktuálním a minulým.

2. Graf sportů - sloupcový graf vzdálenosti sportů pro vizuální porovnání záznamů.

3. Výpis záznamů - přehled posledních 10 záznamů, které byly uloženy. Záznam obsahuje datum, sport, vzdálenost, čas a atributy daného sportu.

Tato obrazovka je editovatelná v záložce *Setting home*.

5.1.2.2 Add data

Tato obrazovka slouží k přidání nových dat k již vytvořeným sportům. Uživatel uvidí seznam sportů, které má k dipozici a po kliknutí na daný sport bude přesměrován na detailní přidání záznamu.

5.1.2.3 Add data detail

Položka se v menu nenachází. Ovšem po kliknutí na daný sport v sekci add data bude uživatel na tuto obrazovku přesměrován. Uživateli je zde umožněno přidávat data dle potřeby.

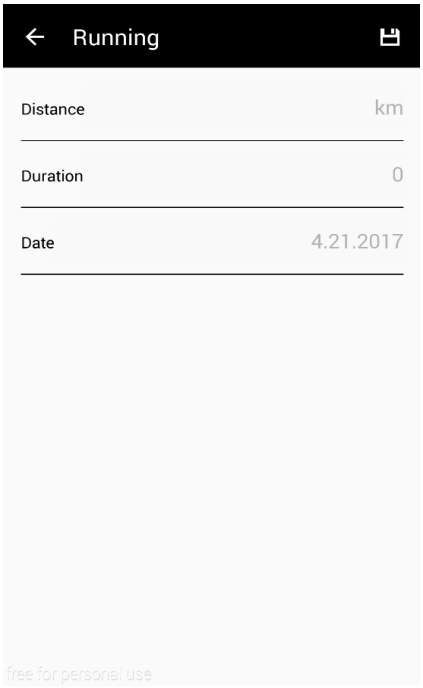
Přidání dat je intuitivní a jednoduché. Jednotlivé položky, které je možné vybrat a vyplnit, jsou zde uvedeny pro jednoduchost opět jako seznam. V levé části se nachází o jaký typ přidání se jedná a po kliknutí na daný prvek seznamu se automaticky uživateli ukáže, jak má dále postupovat.

V případě přidání vzdálenosti uživatel zadává přirozené, nebo reálné číslo pomocí interní klávesnice telefonu.

Pro zaznamenání doby trvání sportovního výkonu je zde využito modální okno. Modální okno obsahuje tři stejné vedle sebe umístěny panely s výběrem času. Zleva se vybírají postupně hodiny, poté minuty a nakonec sekundy. Výběr data záznamu si uživatel navolí také pomocí modálního okna.

Posledním typem přidávaných dat jsou atributy daného sportu, které jsou vyřešeny stejným způsobem jako zaznamenání vzdálenosti. Tedy přes interní klávesnici.

Uložení celého záznamu se provádí pomocí kliknutí na tlačítka v podobě diskety v horním panelu na pravé straně. Po úspěšném vyplnění a uložení záznamu je uživateli vykreslena v dolní části zařízení notifikace „Data saved“ a je přesměrován zpět na obrazovku Add data, kde může přidávat další záznamy. Detail obrazovky je znázorněn na obrázku 6.



Obrázek 6: Detailní obrazovka přidání záznamu

5.1.2.4 Delete data

Obrazovka slouží k mazání záznamů. V horní části je umístěn panel, který slouží na výběr časového rozsahu, ve kterém se mají nacházet položky, které chce uživatel smazat. Výběr data je řešen pomocí dialogového okna, které se zobrazí po kliknutí na ikonu kalendáře.

Pod výběrem panelu data se nachází seznam všech sportů. Po kliknutí na daný sport je uživatel přesměrován na detailní obrazovku mazání sportu.

Poslední sekci této obrazovky tvoří naposledy přidaná data s možnosti okamžitého smazání. Záznam je tvořen jako box s názvem sportu, datem, vzdálenosti, časem a atributy. V pravém horním rohu každý box obsahuje ikonu křížku. Po kliknutí bude uživatel dotázán pomocí dialogového okna, zda chce opravdu daný záznam na trvalo odstranit.

5.1.2.5 Delete data detail

Na tuto obrazovku je uživatel odkázán po kliknutí na daný sport na obrazovce Delete data. Zobrazují se zde záznamy opět v podobě jednotlivých boxů. Záznamy jsou zobrazeny v rozmezí časového intervalu, které si uživatel navolil. V horní části každého boxu je obsažena ikona křížku, která slouží ke smazání konkrétního záznamu s potvrzením dialogového okna. Po odsouhlasení bude záznam natrvalo smazán ze zařízení.

5.1.2.6 User profile

Uživatelský profil je možno vyplnit i editovat v průběhu využívání aplikace.

Fotografie se přidává jednoduchým způsobem. Kliknutím na obrázek v horní části obrazovky je uživatel přesměrován do galerie, kde si může vybrat libovolnou fotografii. Pokud obrázek není nastavený, bude automaticky zvolený anonymní obrázek uživatele.

Zbylé položky jako *Name*, *Weight* a *Height* jsou opět zobrazeny ve formě seznamu pod sebou. *Name* představuje jméno uživatele, *weight* jeho váhu a *height* výšku. Po kliknutí na libovolnou část seznamu uživatel vyplní dané položky pomocí interní klávesnice.

Uložení uživatelských dat se provede po kliknutí na ikonu v horním panelu v pravém rohu. Ikona má podobu diskety. Pokud uživatel vyplnil správně všechny údaje, profil se uloží.

5.1.2.7 Setting home

Zde je umožněna editace položek a sekcí, které se budou zobrazovat na domovské obrazovce. Upravovaná položka je vždy obsažena v levé části a na pravé části se vyskytuje aktuální stav. Stav znázorňuje viditelnost, která se nastavuje pomocí přepínače. Hodnoty přepínače se dají nastavit na hodnoty aktivní, nebo neaktivní. Položky jsou:

Show percentage - V případě, že stav bude aktivní, bude v sekci shrnutí vypočítáván procentuální nárůst, nebo pokles zdolané vzdálenosti.

Show records - Zobrazí záznamy v poslední sekci. Záznam je znázorněn v podobě boxu, který obsahuje základní informace o daném sportovním výkonu. Záznam není možné nijak editovat ani upravovat. Při aktivování této sekce se navíc zobrazí na úvodní obrazovce rolovací tlačítko v pravém dolním rohu, které po kliknutí posune obrazovku na samotný vrchol.

Show chart - Zobrazí graf pod sekci shrnutí na domovské obrazovce. Jedná se o sloupcový graf, kde každý sport má svou barvu pro lepší orientaci v grafu. Jednotky jsou převedeny automaticky na kilometry.

Nastavování jednotlivých sekcí není nijak provázáno mezi sebou a je možnost v libovolné kombinaci přepínat a nastavovat viditelnost i za chodu aplikace. Základním nastavením jsou všechny položky vypnuty.

5.1.2.8 Activities summary

Jedná se o shrnutí dat v podobě spojnicového grafů. Obrazovka je rozdělena na 4 základní části podle časové rozmezí.

- 7 days - zobrazení všech záznamů z posledních sedm dní.
- Month - zobrazení všech záznamů za poslední měsíc.
- Year - zobrazení všech záznamu za poslední rok.

- Custom - libovolné nastavení.

Grafy jsou jednotlivě rozděleny podle typů sportů. Zobrazeny jsou opět jako boxy, které obsahují název sportu, samotný graf a celkové shrnutí času a vzdálenosti. Po kliknutí na libovolnou část boxu je uživatel přesměrován na detailní obrazovku, kde jsou vyobrazeny jednotlivé záznamy.

Přepínaní mezi jednotlivými částmi podle časového rozmezí lze buď na kliknutí časového rozmezí v horním panelu, nebo přejetím prstu po obrazovce podle směru, jakou sekci chce uživatel zobrazit.

V další sekci podrobně rozebereme část *custom*, jelikož má rozsáhlou funkcionalitu .

5.1.2.9 Activities summary - custom

Funguje ve dvou hlavních režimech.

- View
- Compare

5.1.2.10 View

Jedná se o nastavení vlastních parametrů, podle kterých se data mají zobrazovat. Uživatel v sekci *Period 1* vyplní časové rozmezí pomocí modálních oken po kliknutí na kalendář. Další možnosti, jak vybrat časové rozmezí, je zakliknout v boxu v horním panelu časové rozmezí a datum se automaticky nastaví. Horní panel obsahuje nejčastěji využívané časové rozsahy.

Výběr jednotlivých sportů se provádí pomocí zaškrtávacího políčka v sekci pod výběrem data.

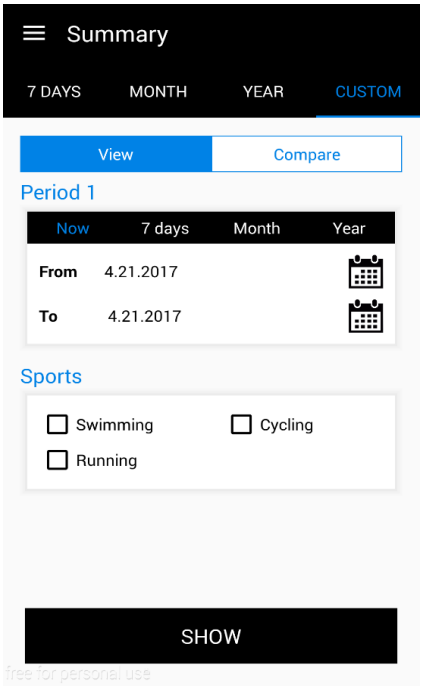
Potvrzení tlačítkem *Show* je uživatel přesměrován na novou obrazovku, kde jsou vyobrazeny spojnicové grafy. Jeden graf odpovídá jednomu sportu. Tlačítkem na levou šipku v horním panelu je uživatel přesměrován zpět. Obrazovka View je znázorněna na obrázku 7.

5.1.2.11 Compare

Obrazovka funguje stejně jako obrazovka View. Navíc je zde přidán jeden box v sekci *Period 2*, kde se opět nastavuje časové rozmezí. Tlačítko *show* zobrazí porovnání sportů v podobě sloupcových grafů a detailní výpis jednotlivých záznamu, které jsou barevně odlišeny.

5.1.2.12 Synchronization

Synchronizace slouží ke spárování aplikace s webovou službou. Při první synchronizaci je uživatel proveden základním přihlašovacím formulářem, při kterém si vyplní přihlašovací údaje ohledně databáze a připojení se ověří. Při úspěšném



Obrázek 7: Obrazovka View

připojení k serveru si uživatel v dalším kroku vytvoří webový účet. Tento účet nemá nic společného s profilovým účtem aplikace.

Následně je uživatel přesměrován zpět a nyní může libovolně odesílat data směrem na server. V případě, že někdo využívá lokálního serveru a nemá veřejnou IP adresu serveru, může ji měnit kliknutím na tlačítko ikony nastavení v horním panelu v pravém rohu pomocí modálního okna.

5.2 Webová aplikace

5.2.1 Příhlášení

Přihlášení probíhá pomocí formuláře. Uživatel zadá své přihlašovací jméno a heslo, které si vytvořil v mobilní aplikaci při první synchronizaci. Pří správném ověření hesla a jména je přesměrován na hlavní část. Přihlašovací formulář je znázorněn na obrázku 8.

5.2.2 Propojení aplikací

V této sekci detailně probereme, jak propojit mobilní aplikaci s webovou aplikací. Jedná se o nejsložitější bod celého projektu. Proto pečlivě krok po kroku rozebereme všechny možnosti. Uživatel má možnost mezi dvěma způsoby, jak provozovat webovou aplikaci. Jedním ze způsobů je provozovat webovou stránku na osobním počítači. V tomto případě je nutností doinstalování webového serveru. Druhým způsobem je provozovat webovou aplikaci přímo na serveru.

5.2.3 Localhost uživatelé

Nedílnou součástí této metody je zjistit lokální IP adresu serveru. Na platformě Windows se lokální IP adresa dá získat několika způsoby. Uvedeme zde pro uživatele nejpřívětivější cestu.

V nabídce start vyberte položku *Ovládací panely*. Poté je zapotřebí kliknout na položku *Síť a internet*. Dále *Centrum sítových připojení a sdílení*. V levém menu se nachází jedna z položek *Změnit* nastavení adaptérů. Poté je zapotřebí kliknout pravým tlačítkem na aktuální připojení a vybrat položku *stav*. Objeví se nové okno, které popisuje aktuální stav připojení. Poslední krokem je vybrat položku *Podrobnosti...* a zobrazí se další okno. V první polovině se nachází položka *IPv4 adresa*. Právě tato adresa určuje lokální IP adresu v síti.

Dalším krokem je ověření, zda router neblokuje propojení zařízení. Zde je ovšem taky několik metod, jak to zjistit. Jednou z možností je v mobilním zařízení kliknout na libovolný webový prohlížeč a do URL adresy zadat právě zjištěnou lokální adresu. Pokud vyhledávání neskončí chybou, úspěšně se mobilní zařízení může propojit s webovou aplikaci. V opačném případě je špatně nakonfigurován router.

V případě neúspěšného připojení je zde několik možnosti, jak odstranit problém. Uvedeme zde několik možných řešení:

1. Vypnout antivirový program.
2. Vypnout bránu firewall.
3. Změnit nastavení routeru.
4. Zkontrolovat správnost IP adres.

5.2.4 Serveroví uživatelé

Druhou možností je mít webovou aplikací dostupnou pomocí veřejné IP adresy. V tomto případě je nutností si zařídit hosting s databází. Následně se pomocí serverového klienta, například FileZilla, pomocí přihlašovacích údajů připojíme k serveru a nahrajeme obsah složky, kde se nachází webová aplikace. Poté se přihlásíme do databáze a pomocí tlačítka *Import* v horním menu nahrajeme databázi. Následně stačí v projektu ve složce `php/core/configuration.php` pouze nastavit správné přihlašovací údaje do databáze.

5.2.5 Obsah

Obsah je rozdělen do čtyř základních logických sekcí.

První sekce s názvem *General* slouží k vypsání základních údajů všech sportů. Jedná se o informace průměrné vzdáleností, maximální vzdálenosti, nejlepšího výsledku v závislosti vzdálenosti na čase a záznamy aktuálního dne.

Druhá sekce *Summary*, neboli shrnutí, všem jednotlivým sportům zobrazuje data vzdálenosti za posledních sedm dní a aktuální měsíc. Tyto údaje jsou potom porovnány ve sloupcovém grafu pod informacemi.

Další sekce zobrazuje vzdálenosti ve spojnicovém grafu a celkové shrnutí za časové rozmezí, jako je posledních sedm dní, měsíc a rok. Mezi celkové shrnutí patří vzdálenost, čas a sportovní atributy.

Poslední sekce je uživatelsky libovolně volitelná. Uživatel má možnost vybrat sporty a časové rozmezí. Tlačítko *show* následně zobrazí sporty ve stejné podobě jako v předchozí sekci.

Obrázek 8: Přihlášení do webové aplikace

Závěr

Náplní bakalářské práce bylo implementovat nativní mobilní aplikaci pro operační systém Android, která slouží jako tréninkový deník a webovou službu, která je přístupná z libovolného zařízení. Obě části práce byly implementovány, řádně otestovány a tedy splňují zadání.

V průběhu implementace byl kladen důraz na strukturování zdrojového kódu do logických celků a programátorskou hygienu.

Text bakalářské práce slouží jako uživatelská příručka pro uživatele a programátorská dokumentace pro programátory. Text práce mimo jiné popisuje aplikace podobného typu, využité technologie a motivaci pro zpracování.

Conclusions

The goal of the thesis was to implement a native mobile app for Android operating system. The app functions as a training diary and a web service accessible from any device. Both parts of the thesis were implemented, properly tested and therefore are in accordance with the assignment.

During the implementation there was emphasis on structuring of the source code into logical units as well as on programming hygiene.

This bachelor thesis serves as user manual and programming documentation for programmers. It also contains description of apps of a similar kind as well as description of applied technologies and motivation for the thesis.

A Zprovoznění mobilní aplikace

A.1 Potřebné požadavky

Požadavky na zprovoznění mobilní aplikace jsou:

- Fyzické zařízení s operačním systémem Android.
- Minimální API verzi, kterou aplikace podporuje, je API 15.

A.2 Postup zprovoznění

- Připojíme zařízení k počítači.
- Vložíme soubor `triloguniversal.apk` ze složky `application/phone` z přiloženého CD do mobilního zařízení.
- V zařízení nainstalujeme daný soubor.
- Po úspěšném nainstalování je aplikace připravena k použití.

B Zprovoznění webové aplikace

B.1 Potřebné požadavky

Požadavky na zprovoznění webové aplikace jsou:

- Apache server a MySQL databáze.
- PHP 7.1.

B.2 Zprovoznění lokálního serveru

- Stáhneme balíček Laragon, který nám zajistí potřebné technologie. https://laragon.org/download.html
- Nainstalujeme a následně spustíme stáhnutý balíček.

B.3 Konfigurace

Popíšeme si zde konfiguraci připojení. Již nebudeme rozlišovat rozdíly mezi veřejnými nebo lokálními IP adresami. Obecný princip je již naprosto shodný.

Databáze obsahuje jednoho již vytvořeného testovacího uživatele s příslušnými daty.

- Z přiloženého CD zkopírujeme obsah složky `application/web/data` do kořenové složky našeho serveru.
- Z přiloženého CD vložíme ze složky `application/web/database` do námi vytvořené databáze soubor `triloguniversal-db.sql`.
- V kořenové složce serveru otevřeme soubor `configuration.php`, který se nachází ve složce `php/core/`.
- V daném souboru přepíšeme připojovací konfiguraci do databáze.
- Ve webovém prohlížeči otevřeme URL adresu k našemu serveru a otestujeme připojení do webové aplikace.
- Do položky pod názvem *Nickname* napíšeme text `user`.
- Do položky pod názvem *Password* napíšeme text `password`.
- Kliknutím na tlačítko *Login* se přihlásíme do webové aplikace.

B.4 Propojení s mobilní aplikací

Údaje vyplněné v mobilní aplikaci slouží pro přihlášení k webovému rozhraní.

C Obsah přiloženého CD

Struktura přiloženého CD je složena ze čtyř základních složek, ve kterých je uložen celý projekt a dokumentace. Struktura CD je následující:

`application/`
: Složka je rozdělena na podsložky `application/phone` a `application/web`. Složka `web` obsahuje databázi a zdrojový kód potřebný na spuštění webové aplikace. Složka `phone` obsahuje instalační soubor mobilní aplikace.

`documents/`
: Složka obsahuje uloženou práci ve formátu PDF. Složka dále obsahuje potřebné soubory k vytvoření daného souboru.

`data/`
: Obsahuje zdrojové kódy mobilní aplikace a webové aplikace. Složka je dále rozčleněna na podsložky `data/phone` a `data/web`. Kde jsou v příslušných složkách vložené zdrojové soubory jednotlivých částí projektu.

`readme.txt`
: Obsahuje návod na spuštění a zprovoznění mobilní a webové aplikace. První jsou v souboru popsány instrukce na spuštění mobilní aplikace a následně webové aplikace.

Literatura

[1] BOOTSTRAP, The world's most popular mobile-first and responsive front-end framework. Bootstrap · The world's most popular mobile-first and responsive front-end framework. [online][cit. 08.04.2017]. Dostupné z: http://getbootstrap.com/

[2] HTML TUTORIAL W3Schools Online Web Tutorials [online][cit. 08.04.2017]. Dostupné z: https://www.w3schools.com/css/

[3] JQUERY jQuery [online][cit. 08.04.2017]. Dostupné z: https://jquery.com/

[4] CSS TUTORIAL W3Schools Online Web Tutorials [online][cit. 08.04.2017]. Dostupné z: https://www.w3schools.com/css/

[5] ENDOMONDO Endomondo [online][cit. 22.04.2017]. Dostupné z: https://www.endomondo.com/about

[6] SQL INJECTION W3Schools Online Web Tutorials [online][cit. 22.04.2017]. Dostupné z: https://www.w3schools.com/sql/sql_injection.asp

[7] What are ORM Frameworks? – KillerPHP.com. KillerPHP.com [online]. Copyright ©1996 [cit. 23.04.2017]. Dostupné z: http://www.killerphp.com/articles/what-are-orm-frameworks/

[8] PHP: hash - Manual . PHP: Hypertext Preprocessor [online]. Copyright © 2001 [cit. 23.04.2017]. Dostupné z: http://php.net/manual/en/function.hash.php

[9] Understanding Android API Levels - Xamarin. Developer Center - Xamarin [online]. Copyright ©[cit. 24.04.2017]. Dostupné z: https://developer.xamarin.com/guides/android/application_fundamentals/understanding_android_api_levels/

[10] AJAX Introduction. W3Schools Online Web Tutorials [online][cit. 24.04.2017]. Dostupné z: https://www.w3schools.com/xml/ajax_intro.asp

[11] Android - What are the differences between activity and fragment - Stack Overflow. Stack Overflow [online][cit. 24.04.2017]. Dostupné z: http://stackoverflow.com/questions/25822656/what-are-the-differences-between-activity-and-fragment

[12] Laragon - Turns your computer into a powerful server. Laragon - Turns your computer into a powerful server [online]. Copyright © Laragon [cit. 24.04.2017].

Dostupné z:
https://laragon.org/