

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

TAJMLAJN.CZ - MOBILNÍ/WEBOVÁ SLUŽBA  
PRO PLÁNOVÁNÍ DOSAŽENÍ ÚKOLU

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

TOMÁŠ CHLÁPEK

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

## TAJMLAJN.CZ - MOBILNÍ/WEBOVÁ SLUŽBA PRO PLÁNOVÁNÍ DOSAŽENÍ ÚKOLU

TAJMLAJN.CZ - MOBILE/WEB SERVICE FOR PLANNING OF GOAL ACHIEVEMENT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ CHLÁPEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IGOR SZÓKE, Ph.D.

BRNO 2013

## Abstrakt

Cílem této práce je tvorba mobilní a webové služby, jež umožní uživateli tvorbu časového plánu pro dosažení úkolu. Časový plán nabízí uživatelům možnost si lépe a kvalitněji rozvrhnout svůj čas určený pro práci na zadaném úkolu. Aplikace je určena pro operační systém Android a komunikuje s MySQL databází prostřednictvím webové služby umístěné na vzdáleném serveru. Je brán zřetel na stálou a pravidelnou komunikaci s uživateli, jenž aplikaci testují. Na základě jejich zpětných reakcí dochází ke změnám a určuje se tímto další vývoj.

## Abstract

The aim of this work is create a mobile service, which allows user to make timeline for achieving a goal. Timeline offers opportunity for users to organize their projects more efficiently than before. The application is developed for Android OS and communicates with MySQL database through web service running on remote server. Important aspect is permanent and regular communication with early adopters. Changes are made based on feedback from users that determine future of development.

## Klíčová slova

android, mobilní služba, webová služba, timeline, startup, java, timemanagment, MySQL, SQLite, PHP

## Keywords

android, mobile service, web service, timeline, startup, java, timemanagment, MySQL, SQLite, PHP

## Citace

Tomáš Chlápek: Tajmlajn.cz - mobilní/webová služba pro plánování dosažení úkolu, bakalářská práce, Brno, FIT VUT v Brně, 2013

# Tajmlajn.cz - mobilní/webová služba pro plánování dosažení úkolu

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Igora Szókeho, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Tomáš Chlápek

15. května 2013

## Poděkování

Děkuji mému vedoucímu Ing. Igoru Szökemu, Ph.D. za odborné vedení mé práce, konzultace a profesionální přístup.

© Tomáš Chlápek, 2013.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1 Úvod</b>	<b>3</b>
<b>2 Mobilní aplikace formou startupu</b>	<b>4</b>
2.1 Vývoj pod operačním systémem Android	4
2.1.1 Historie	4
2.1.2 Verze	4
2.1.3 Architektura	6
2.1.4 Vývoj	7
2.1.5 Rozdělení aplikace	8
2.2 Ukládání dat	10
2.2.1 SQLite	10
2.2.2 MySQL	10
2.3 Webová služba	11
2.3.1 PHP5	11
2.3.2 JSON	12
2.4 Startup techniky	13
2.4.1 Vývoj metodou lean startupu	13
2.5 Zhodnocení konkurenčních řešení	15
2.5.1 Astrid	15
2.5.2 Preceden	15
2.5.3 Moment Garden	16
2.5.4 Srovnání	16
<b>3 Návrh kalendáře a ukládání dat</b>	<b>17</b>
3.1 Zobrazení kalendáře	17
3.1.1 Vykreslování	18
3.2 Komunikace se vzdáleným serverem	19
3.2.1 Zpracování na serveru	19
3.3 Databáze uživatelů, událostí a stavů	20
3.3.1 Správa databáze	20
<b>4 Časový plán a manipulace s událostmi</b>	<b>22</b>
4.1 Registrace a přihlášení	22
4.2 Tvorba a editace událostí	24
4.2.1 Správa timeline	24
4.2.2 Správa událostí	25
4.2.3 Detail události	25
4.3 Sociální vazba	25

4.4	Vykreslení časových plánů . . . . .	27
4.5	Podpůrná webová služba . . . . .	28
4.6	Testování . . . . .	29
4.6.1	Testování v průběhu vývoje . . . . .	29
4.6.2	Závěrečné testování . . . . .	31
<b>5</b>	<b>Závěr</b>	<b>33</b>
<b>A</b>	<b>Obsah CD</b>	<b>35</b>

# Kapitola 1

## Úvod

Každý rok se potýká řada studentů vysokých škol s nelehkým úkolem. Tím je vypracování diplomové práce. Studenti si své práce volí, konzultují se svým vedoucím, v lednu část odprezentují, během letního semestru vyvíjejí a v květnu opět konzultují a odevzdávají. Toto je standardní pracovní struktura. Řada studentů má však aktivnější přístup a před začátkem vývoje si vytváří předběžný plán, kterého se chtějí držet. Pokud si takovýto student vybere aktivního vedoucího, jenž ho v tomto podporuje, tak má solidní základ k vytvoření kvalitní diplomové práce. Hlavním cílem je vytvoření a nasazení mobilní aplikace a s ní spolupracující webové služby, primárně zaměřené na tyto studenty, kteří se chtějí aktivně věnovat svým diplomovým pracím, vytyčovat si cíle, překonávat překážky, dynamicky zaznamenávat svůj progres a zdárně dokončit projekt. Druhou skupinou, na kterou se aplikace zaměřuje, jsou vedoucí prací, jež mohou takto sledovat vývoj svých studentů a definovat milníky.

Účelem tohoto díla je vytvořit nástroj, umožňující vytvářet časové plány, realizované formou timeline, pro mobilní zařízení běžící na systému Android. Tyto timeline si kladou za úkol přinést možnost zaznamenávat, editovat a sledovat vývoj při řešení dlouhodobého projektu. Nabízejí také možnost sdílet veškeré aktivity mezi spřátelenými uživateli a dodat jim tak možnost vidět kromě vlastního vývoje také splněné úkoly a úspěchy svých kolegů či přátel. Vedoucím prací je tímto nabídnuta možnost nahlížet na progres jejich studentů a reagovat na něj.

## Kapitola 2

# Mobilní aplikace formou startupu

TimeTrace je mobilní aplikace určená pro studenty, týmové projektanty a vedoucí diplomových prací ke sledování pokroku a vývoje za určitá časová období. Aplikace je vyvíjena formou startupu<sup>1</sup>.

### 2.1 Vývoj pod operačním systémem Android

Mobilní aplikační služba je vyvíjena pod operačním systémem Android. Jedná se o počítačový software založený na open source<sup>2</sup> platformě, jenž je určený primárně pro mobilní zařízení. Operační systém je vyvíjen konsorciem Open Handset Alliance. Je u něj kladen důraz na nezávislost na chipsetu<sup>3</sup>, velikosti a rozlišení obrazovky zařízení.



Obrázek 2.1: Logo Android

#### 2.1.1 Historie

Android vznikl v roce 2003 v Kalifornii založením společnosti Android, Inc., zabývající se vývojem mobilních zařízení. V polovině roku 2007 došlo k odkoupení společností Google a k oznámení prvního mobilu postaveného na platformě Android, jenž je pojmenovaný jako HTC Dream a stojí na linuxovém jádře verze 2.6.25. Jedná se o platformu zastřešující operační systém, middleware<sup>4</sup> a aplikace.

#### 2.1.2 Verze

Operační systém Android od momentu kdy byl odkoupen společností Google vydal množství aktualizací, z nichž každá přináší řadu nových inovací a podstatných zlepšení v reakci na přání a stížnosti uživatelů. Počínaje verzí 1.5, Google začal všechny nově vycházející

<sup>1</sup>životní cyklus podnikání s cílem zavést novou službu nebo produkt za nejistých podmínek

<sup>2</sup>počítačový software s otevřeným volně dostupným kódem

<sup>3</sup>sada elektronických komponent zajišťující datový tok mezi procesorem, pamětí a periferními zařízeními

<sup>4</sup>software nabízející služby aplikacím skrze operační systém



verze pojmenovávat. Tato označení jsou specifická tím, že odrážejí jména zákusků[1]. S každou novou aktualizací vychází nová verze API<sup>5</sup> rozšiřující funkcionalitu předešlé. Tento fakt přináší nutnost uvažovat při vývoji aplikací s možností zpětné podpory. Ta se definuje prostřednictvím speciálního souboru AndroidManifest.xml, ve kterém je třeba určit minimální podporovanou verzi a maximální verzi, pro kterou je aplikace optimalizována a testována.

## 1.0

První verze Androidu vydaná v září 2008. Oproti dosavadním mobilním operačním systémům přináší nové uživatelské prostředí a podporu portálu Android Market primárně sloužícího k distribuci aplikací. Tohoto zavedeného konceptu se Android drží dodnes.

## 1.5 Cupcake

Jedná se o první verzi oficiálně pojmenovanou po zákusku. Přináší podporu pro widgety<sup>6</sup>, automatickou změnu orientace displeje, možnost nahrávat a přehrávat videa a ty pak následně sdílet skrze portál Youtube.

## 1.6 Donut

Verze Androidu přinášející svižné vyhledávání, podporu WVGA rozlišení a zvýšení rychlosti. Pro vývojáře patří mezi nejzajímavější novinky možnost vkládat obsah do výsledků pro vyhledávání a zpracování gest skrze Gesture Builder.

## 2.0 Éclair

Éclair dovoluje uživatelům používat více účtů v rámci jednoho zařízení, což dosti rozšířilo možnosti využití. Má zabudovanou podporu pro vyvíjející se HTML5, Bluetooth 2.1 a novou verzi Google Maps. Nabízí možnost ukládat SMS a MMS zprávy, vylepšené funkce fotoaparátu, rychlejší a chytřejší klávesnici a spolupracuje s více typy zařízení s rozdílnými velikostmi a rozlišeními.

## 2.2 Froyo

Froyo přináší jako hlavní novinku službu Android Cloud to Device Messaging. Ta umožňuje zasílání upozornění a zpráv na mobilní telefon a jeho synchronizaci s jinými zařízeními. Dalšími novinkami je zabudovaná podpora JIT kompilátoru urychlujícího překlad, možnost vytvořit Wi-fi hotspot, nový javascript engine V8, podpora Adobe Flash a Microsoft Exchange a služba Over-the-Air přinášející možnost instalovat v reálném čase aplikace z webové verze Android Marketu přímo do telefonu. Aplikace lze nyní instalovat na paměťovou kartu. Tato vylepšení přivádí řadu nových uživatelů k systému Android. Díky těmto nově zabudovaným funkcím se tato verze stává zásadní. Aplikace TimeTrace tuto verzi již podporuje.

---

<sup>5</sup>protokol sloužící ke komunikaci se softwarovými komponentami

<sup>6</sup>malá periodicky obnovovaná aplikační okna vkládaná na pracovní plochu

## 2.3 Gingerbread

Gingerbread je v současnosti nejrozšířenější verzí systému Android. Hlavní novinkou je podpora Near Field Communications. Jedná se o bezdrátovou komunikaci pro přenášení dat na krátké vzdálenosti. Využití je převážně ve čtení elektronických značek vyskytujících se v letácích, billboardech či plakátech. Dalším zlomovým bodem je možnost uskutečnit volání skrze SIP<sup>7</sup> protokol. Rozrostla se také podpora využívání událostí senzorů určených pro vývoj nativních aplikací, nové multimediální formáty a přístup k více kamerám.

## 3.2 Honeycomb

Honeycomb rozšiřuje Android o možnost jej zobrazovat na velkých obrazovkách tabletů. Tato verze je určena výhradně pro tablety a pro mobilní telefony nikdy nevyšla. Hlavními novinkami je značně rozšířená podpora různých velikostí displejů a jejich rozlišení a nové grafické uživatelské rozhraní přinášející možnost práce s fragmenty. Tato verze je prezentována jako herní platforma. Neopomenutelnou novinkou je nově zavedený Action Bar nabízející navigaci, zobrazení statusu a možnosti nastavení v horní části obrazovky.

## 4.0 Ice Cream Sandwich

Ice Cream Sandwich spojuje ovládání a vzhled verzí Gingerbread a Honeycomb do jednoho celku. První novinkou je integrace systémových tlačítek přímo na obrazovku, čímž vzniká možnost zbavit se tlačítek doposud ležících mimo displej. Spolu s touto verzí přichází možnost pořídit screenshot. Tato novinka umožňuje lidem testujícím aplikace zaslat vývojáři zobrazení grafické stránky aplikace na jejich zařízeních a tím urychlit proces ladění, čehož bylo v rámci testování často využíváno. Pro oblast plánování je důležitým milníkem přidání podpory pro dynamické změny v kalendáři skrze rozhraní Google Calendar API. Toho bylo využito pro rozšíření aplikace o možnost přidávání událostí do kalendáře poskytovaného společností Google.

## 4.0 Jelly Bean

Jelly Bean přináší s novou verzí systému Android řadu změn. Do vyhledávače je nově zakomponována služba Google Now, jež působí jako inteligentní osobní asistent odpovídající na položené otázky. Došlo ke zlepšení podpory přehrávání videa v HTML5, rozšířili se vlastnosti fotoaparátu, hodin a klávesnice. Velkým přínosem je rozšíření možností ladění aplikací poskytované skrze nastavení pro vývojáře.

TimeTrace podporuje mobilní zařízení Android počínaje API 8 vztahující se k verzi Froyo. Podíl zařízení na trhu se staršími verzemi je dle statistik poskytovaných společností Google zanedbatelný.

### 2.1.3 Architektura

Architektura systému je rozdělena na 5 základních vrstev (viz obrázek 2.2).

- jádro postavené na Linuxu 2.6

---

<sup>7</sup>protokol umožňující volání skrze internet

- knihovny nabízející uživatelům specifické funkce (v rámci aplikace využity Surface-Manager a SQLite)
- android runtime využívající virtuální stroj Dalvik
- aplikační framework<sup>8</sup> zajišťující přístup ke službám
- aplikace



Obrázek 2.2: Systémová architektura[2]

### 2.1.4 Vývoj

Mobilní aplikace byla vyvinuta v programovacím jazyce Java využívajícím Android Software Development Kit (dále jen SDK). Tento balíček nabízí velkou škálu nástrojů nezbytných pro vyvinutí funkční plnohodnotné aplikace. Prvním nástrojem, se kterým se každý vývojář android aplikací setká, je Android SDK Manager. Rozděluje jednotlivé komponenty do balíčků a umožňuje skrze uživatelské prostředí možnost jejich stažení či aktualizací. To nabízí možnost stažení nejnovějších API ihned po vydání nových verzí Androidu na trh. Nástroj AVD Manager poskytuje možnosti vytváření a správy virtuálních zařízení. Lze s ním vytvořit nepřehledné množství virtuálních zařízení, jež odpovídají svým reálným předloham. Lze je poté spouštět skrze emulátor. Tohoto bylo hojně využíváno především v poslední části vývoje. Na základě zpětných reakcí od zákazníků byla vytvářena virtuální zařízení emulující jejich mobilní telefony a tablety a zde byla prováděna velká část testování.

<sup>8</sup>softwarová platforma určená k vývoji aplikací

Nevýhodou tohoto řešení jsou velmi značné hardwarové nároky emulátoru, které mají vliv na zpomalení práce.

Samotné testování na reálných přístrojích se liší v závislostech na výrobci, verzi a kusu zařízení. Pro zprovoznění ladění je potřeba propojení mobilního zařízení s počítačem skrze USB rozhraní, instalace potřebných ovladačů a zajištění komunikace skrze ADB. Nástroj ADB, dodávaný prostřednictvím Android SDK, slouží ke komunikaci s instancí emulátoru či připojeným přístrojem na bázi klient-server.

Za vývojovou platformu pro implementaci aplikace bylo zvoleno vývojové prostředí Eclipse s využitím Android Development Tools pluginu<sup>9</sup>. Tento plugin rozšiřuje možnosti Eclipse o vytváření a správu Android projektů, vytváření uživatelských prostředí, ladění a export výsledného projektu do APK<sup>10</sup> balíčku.

Každý projekt je po vytvoření uspořádán do specifické stromové struktury adresářů, které zpřehledňují práci a umožňují rychlý přístup k jednotlivým složkám[11].

Mezi základní prvky patří:

- AndroidManifest.xml - XML soubor poskytující operačnímu systému informace o aplikaci, jež musí být součástí každého projektu
- assets - Adresář obsahující statické soubory (například písemné fonty)
- bin - Adresář obsahující projekt po jeho kompilaci
- gen - Adresář obsahující soubory vygenerované překladačem
- libs - Adresář obsahující knihovny
- src - Adresář obsahující zdrojový kód projektu
- res - Adresář obsahující návrhy grafického rozhraní, obrázky, informace o barvách, jazycích a jejich hodnoty pro různé typy zařízení

### 2.1.5 Rozdělení aplikace

Veškeré aplikace v Androidu jsou rozděleny na komponenty. Každá z těchto komponent je vstupním bodem uživatele či systému do aplikace.

#### Aktivity

Komponenty poskytující uživateli možnost aktivně působit. Aktivity jsou svázány a vzájemně se ovlivňují. Při startu nové aktivity je předešlá pozastavena a vložena na vršek zásobníku, odkud je zavolána v případě, že se uživatel z nově zobrazené obrazovky chce vrátit zpět. Ke spouštění dochází voláním metody `startActivity()`. Před voláním mohou být cílové aktivitě zaslány specifikované hodnoty, jenž takto umožňují přímočarou komunikaci. Tohoto je v aplikaci využíváno pro předávání unikátních identifikátorů událostí a uživatelů. Každá aktivita nabývá vždy jednoho ze 4 stavů[12]. V závislosti na stavu mohou být nadefinovány a prováděny specifické akce (viz obrázek fig:imgact).

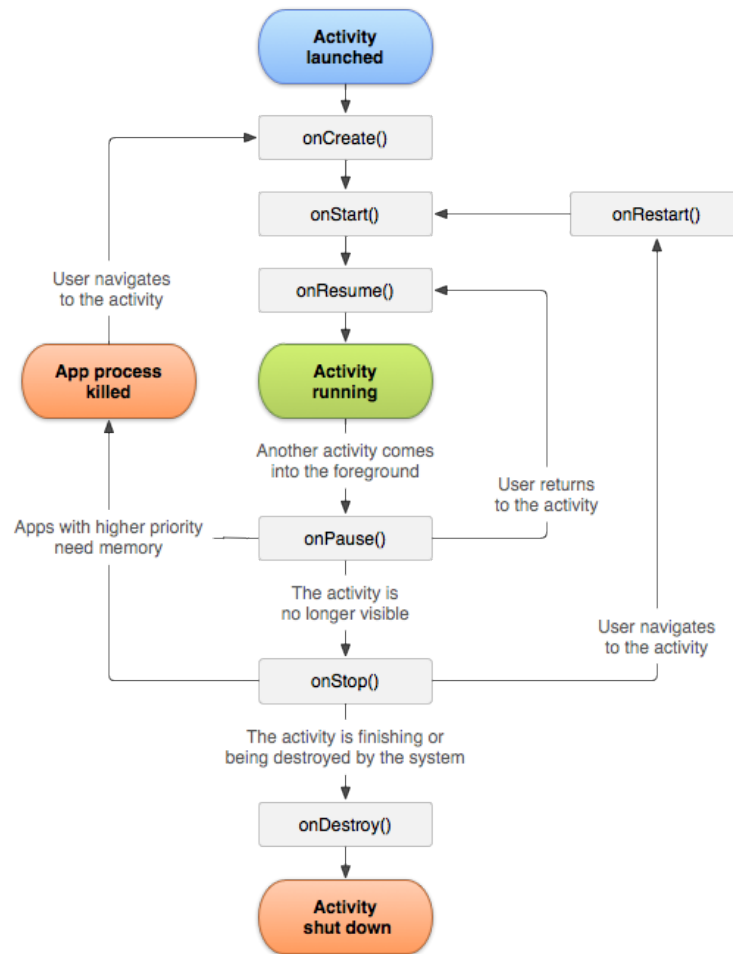
---

<sup>9</sup>zásuvný modul

<sup>10</sup>Application Package File

Stavy:

- Aktivní - Aktivita běží v popředí
- Pozastavená - Aktivita běží v pozadí, je viditelná
- Zastavená - Aktivita běží v pozadí, je neviditelná
- Mrtvá - Aktivita neběží



Obrázek 2.3: Životní cyklus aktivity[4]

## Služby

Komponenty zajišťující běh dlouhodobých procesů na pozadí, jenž pracují nezávisle na jakékoliv aktivitě.

## Dodavatelé obsahu

Dodavatelé obsahu spravují přístup k datům a jejich zabezpečení rozličným aplikacím. Jejich velké uplatnění lze nalézt v případech, kdy aplikace potřebuje získat přístup

ke kontaktům, seznamu hovorů či obrázkové galerii. Dodavatelů obsahu bylo využito pro navázání komunikace a přenosu dat mezi aplikacemi TimeTrace a Google Calendar.

## Záměry

Systémové zprávy informující o externích událostech jako je nízká hodnota baterie, přijetí SMS zprávy či vložení paměťové karty do telefonu. Lze je vytvářet a spouštět v závislostech na určitých podnětech.

## 2.2 Ukládání dat

Android nabízí pro perzistentní ukládání dat 5 možností[5]:

- SharedPreferences - Ukládání primitivních datových typů ve formátu klíč - hodnota.
- Interní úložiště - Ukládání dat do interní paměti. K datům kromě samotné aplikace nemá nikdo jiný přístup.
- Externí úložiště - Ukládání dat do paměti zařízení či na paměťovou kartu. K datům možno přistupovat.
- SQLite databáze - Ukládání dat do databáze. K datům lze přistupovat prostřednictvím definovaného jména skrze aplikaci.
- Síťové připojení - Ukládání dat do vzdáleného úložiště. Nutnost internetového připojení.

Pro účely aplikace TimeTrace je využito ukládání strukturovaných dat do MySQL databáze na vzdáleném serveru. Informace o uživateli jsou zaznamenány do SQLite databáze. Pro perzistentní uložení přihlašovacích údajů je využíváno SharedPreferences.

### 2.2.1 SQLite

SQLite je databázový stroj využívaný operačním systémem Android, jenž implementuje téměř celý standard SQL92. Nabízí důležité vlastnosti pro tvorbu a správu databáze. Je však kompaktní a odlehčený oproti jeho konkurenci. Data jsou do něj ukládána s využitím primárního klíče. Hlavními přednostmi je podpora transakcí a ukládání celé databáze do jediného souboru. Tímto odpadá potřeba serveru.

Systém Android nabízí plnou podporu SQLite databází. Ty jsou po jejich vytvoření přístupné prostřednictvím svého jména všem aplikačním třídám. Mimo aplikaci nelze k vytvořeným tabulkám přistupovat. V rámci aplikace je naimplementována SQLite databáze pro ukládání informací o aktuálně přihlášeném uživateli. To oproti práci a komunikaci se vzdálenou databází umožňuje rychlejší odezvu a možnost kdykoliv přistoupit k osobním datům.

### 2.2.2 MySQL

MySQL je relační databázový server, jenž vznikl roku 1995 ve Švédsku pod vedením firmy MySQL AB. Je u něj kladen velký důraz na výkon a především škálovatelnost softwaru. I přes absenci možností jako jsou ukládání procedur, triggerů či transakce se stal za krátkou dobu velmi populárním.

Mezi jeho hlavní výhody patří[8]:

- Zpracování více tabulek - HEAP, InnoDB, MyISAM
- Replikace
- Fulltextové vyhledávání
- Rozšíření mezi platformami
- Pokročilé možnosti zabezpečení

## 2.3 Webová služba

Mobilní aplikace TimeTrace je službou, jenž svá veškerá data zasílá vzdálenému serveru. Pro práci je potřeba registrace nového účtu a následné přihlášení. Po přihlášení se již uživatel dostává ke všem funkcím, jež aplikace nabízí. Hlavním důvodem pro toto jednání je, aby uživatel mohl ke svým datům přistupovat nezávisle na tom, ze kterého zařízení se přihlásí. Za tímto účelem byla na vzdáleném serveru vybudována webová služba, jenž přijímá požadavky aplikace a zasílá jí poté nazpět všechna potřebná data. Pro realizaci byl zvolen jazyk PHP5.

### 2.3.1 PHP5

PHP je multiplatformní skriptovací programovací jazyk určený pro vývoj webových aplikací a tvorbu dynamických internetových stránek. Na serveru dochází ke generaci HTML kódu, který je poté zasílán klientovi. Klient získává výsledky, jež mu server vygeneroval. Nevídi však již samotný program.

Jazyk PHP vzniknul v roce 1995 jako skript v jazyce Perl umožňující zaznamenávat informace o návštěvnicích. Počet návštěvníků poté prezentoval na své webové stránce. Z důvodu enormního zájmu vznikla sada nástrojů pojmenovaná jako Personal Home Page (PHP). Po jisté době bylo upuštěno od jazyka Perl a vývoj pokračoval v jazyce C. V roce 1997 došlo ke změně názvu na Hypertext Preprocessor. S překročením hranice milionu uživatelů se stal nejpobulárnějším skriptovacím jazykem na světě.[9] S příchodem verze PHP4 bylo hlavní změnou zdokonalení zacházení s prostředky z důvodu využití jazyka na úrovni korporací. Bylo zavedeno šifrování, jež je v rámci webové služby využito pro zakódování citlivých uživatelských dat. PHP začal podporovat objektově orientované programování, což přineslo řadu výhod, jež byly v aplikaci využity. Taktéž došlo ke zpřístupnění podpory Javy a vázání se na její objekty.

Webová služba podporující správu uživatelů a jejich dat byla implementována prostřednictvím jazyka PHP5. Ten přinesl mnoho novinek a zdokonalení v oblasti objektově orientovaného programování, jako jsou explicitní konstruktory a destruktory, abstraktní třídy, klonování objektů či podpora rozhraní. Nově se nabízí možnosti zpracování výjimek, zlepšení podpory XML a zavedení SQLite databáze.

Pro potřeby vývoje byl zvolen pokročilý textový editor Sublime Text 2. Tento editor nabízí možnosti dodatečné instalace velkého množství zásuvných modulů, jež urychlují práci při psaní PHP skriptů. Ty nejsou limitovány pouze na generování výstupu v podobě HTML. Jejich výstupy mohou být obrázky, text, PDF dokumenty či XML soubory. Webová služba

komunikuje s mobilní aplikací na bázi zasílání zpráv ve formátu JSON. Pro získání požadovaných dat služba spolupracuje s databázovým serverem MySQL. Skript se za tímto účelem připojuje ke vzdálené databázi.

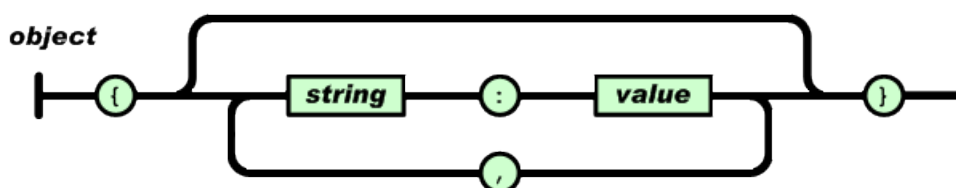
Důležitým bodem jsou uživatelská oprávnění. Skript při každém pokusu o připojení k MySQL z důvodu navázání komunikace musí vybrat požadovanou databázi. Na to potřebuje daná oprávnění, bez nichž ke komunikaci nedojde a spojení bude ukončeno. Po identifikaci a jednorázovém připojení k databázi dojde k ustanovení spojení, které je udržováno až do vykonání všech odpovídajících dotazů.

### 2.3.2 JSON

JSON<sup>11</sup> je odlehčený formát pro výměnu dat. Důraz je kladen na jednoduchou čitelnost a nezávislost na platformě. Je zavedeno 6 základních typů, jež lze do něj přímo bez jakékoli konverze vložit.

- JSONNumber
- JSONBoolean
- JSONString
- JSONNull
- JSONArray
- JSONObject

K přenosu dat se využívá dvou nabízených struktur. První z nich je JSONObject, jež se skládá z netříděných dat (viz obrázek 2.4). Pro každou datovou položku připadá klíč definovaný svým jménem ve formátu JSONString. Díky možnosti zanořovat objekty do sebe lze vytvářet složitější datové struktury. Veškeré data získaná ze vzdálené databáze prostřednictvím webového serveru jsou do aplikace zasílána ve formátu JSONObject. První položka každého takto zasláního objektu je nastavována jako indikátor úspěšně vykonaného dotazu. V případě nastavení její hodnoty na nulu dochází na straně aplikace ke zjištění, že došlo k chybě.

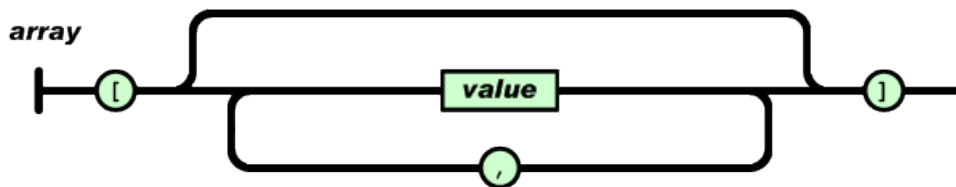


Obrázek 2.4: JSONObject[6]

<sup>11</sup>JavaScript Object Notation



JSONArray je setříděná kolekce hodnot ohraničená hranatými závorkami (viz obrázek 2.5). Hodnota může být jakéhokoliv datového typu JSON. V rámci aplikace je JSONArray využíváno pro seskupování položek stejného typu.



Obrázek 2.5: JSONArray[6]

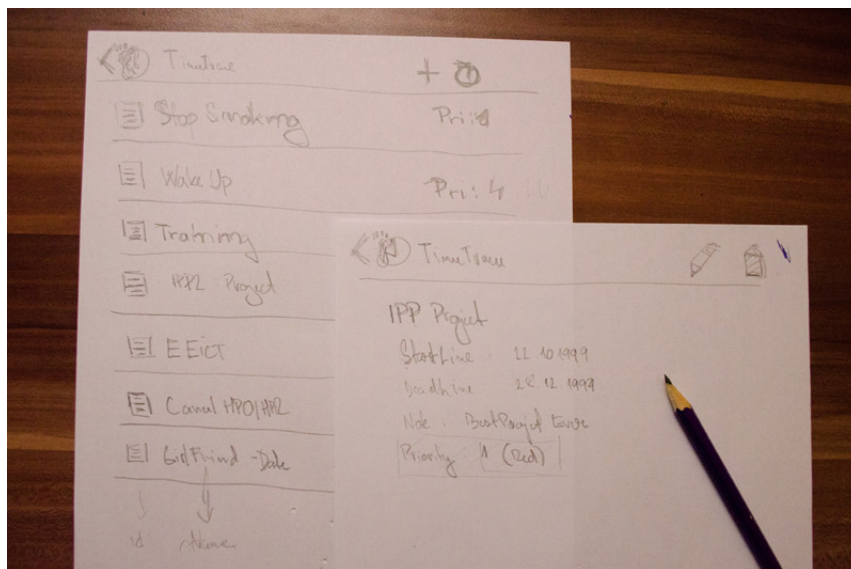
Oproti formátu XML<sup>12</sup> je JSON výrazně menší, rychlejší a jednodušeji analyzovatelný. Existují metody, jež umožňují oba formáty navzájem převádět.

## 2.4 Startup techniky

### 2.4.1 Vývoj metodou lean startupu

Aplikace TimeTrace od začátku svého vývoje čerpá principy Lean Startup[10]. Cílem je se dostat od bodu počátečního nápadu a myšlenky, až k funkční životaschopné službě.

V počátku vývoje byl vytvořen hrubý návrh aplikace, tzv. mockup (viz obrázek 2.6). Ten byl předložen a konzultován s testery. Od první fáze návrhu došlo vzhledem k požadavkům testerů k mnohým změnám, které se odrazily na dalších verzích.

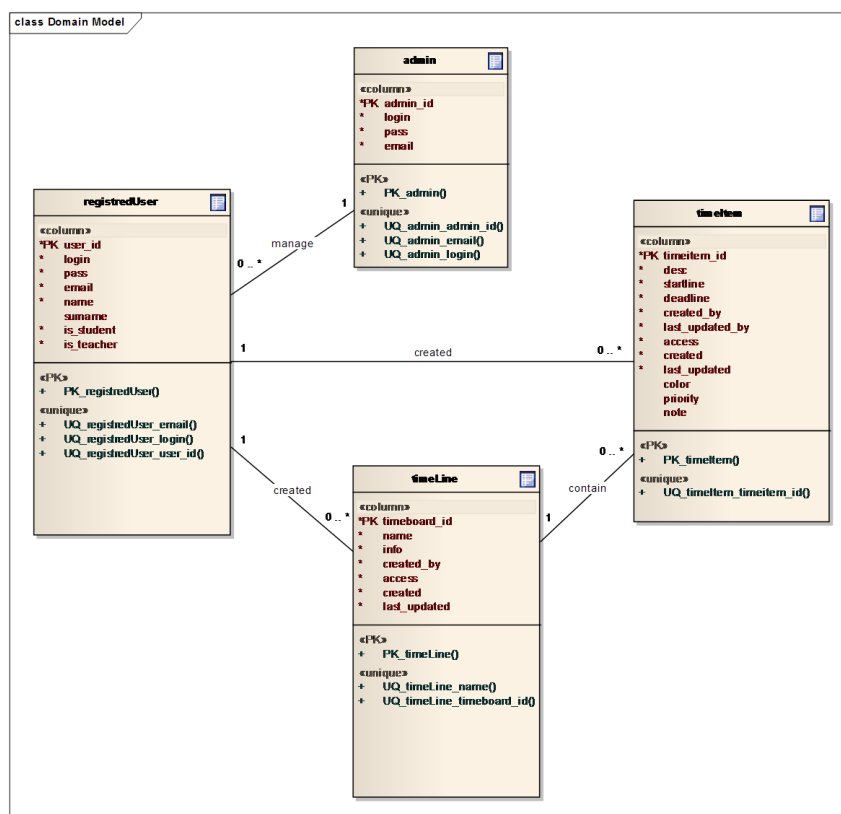


Obrázek 2.6: Mockup.

Před začátkem samotné implementace byl sestaven doménový model tříd, na jehož základě se navrhovala struktura celé databáze. Původní koncept zamýšlel vytvořit 4 základní

<sup>12</sup>eXtensible Markup Language

objekty. Mezi tyto objekty měl patřit administrátor, registrovaný uživatel, timeline<sup>13</sup> a událost. Každý z nich měl mít svůj unikátní identifikační řetězec, jméno, popis a potřebné časové údaje (viz obrázek 2.7). Model byl navrhnutý tak, aby si každý nový uživatel při registraci musel zvolit zda jde o studenta či učitele a na základě toho se měl program dále větvit.



Obrázek 2.7: Doménový model

Od tohoto nápadu bylo upuštěno, jelikož se uživatelům nelíbila představa, že by byli jako studenti omezeni jen na určité činnosti. Jelikož je vývojářem aplikace jen jedna osoba, jež aplikaci sama implementovala, bylo rozhodnuto o vynechání tabulky administrátora. Místo tohoto nápadu byl zvolen koncept nastavení veřejného sdílení timeline napříč ostatními uživateli zařazenými mezi přáteli.

<sup>13</sup>časová osa

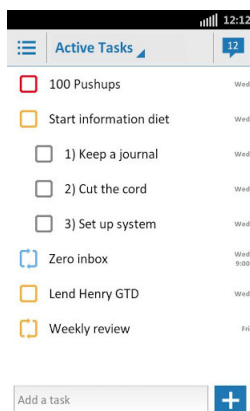
## 2.5 Zhodnocení konkurenčních řešení

V oblasti řešení dlouhodobých problémů s použitím časové osy se setkáváme s několika konkurenčními řešeními, na které se v této podkapitole zaměříme.

### 2.5.1 Astrid

Astrid je mobilní službou, sloužící uživatelům k nahrávání jednoduchých poznámek a tvorbě TODO listů. Vše je po zaregistrování ukládáno na server a nabízí se možnost jednotlivé úkoly procházet a editovat skrze webové rozhraní. V základní verzi umožňuje psaní pouze textových úkolů, nastavení připomenutí a základní malý widget<sup>14</sup> na plochu s omezeným množstvím informací.

Astrid v současné době používá dle informací z Google Play přes milion uživatelů. Důvodem je uživatelsky přívětivé prostředí, lokalizace do mnoha jazyků, jednoduchá možnost nastavení a zakomponování různých vtipných filtrů (viz obrázek 2.8). Aplikace samotná neřeší plánování dlouhodobých úkolů, jejich rozdělení do milníků ani synchronizaci s přáteli (pouze prostřednictvím sociálních sítí).



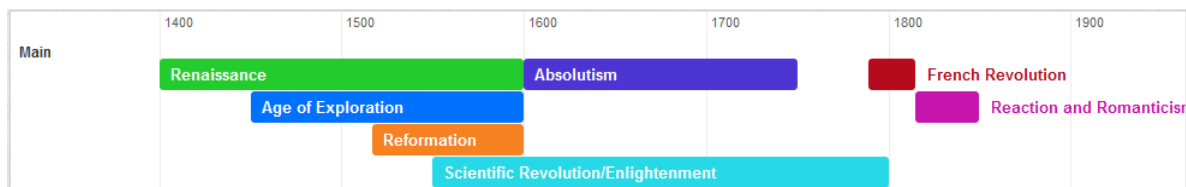
Obrázek 2.8: Snímek obrazovky mobilní aplikace Astrid.

### 2.5.2 Preceden

Preceden je webová aplikace umožňující tvorbu časových os. Pracuje na principu tvorby vrstev, do kterých se vkládají jednotlivé události pro určité období. Aplikace pracuje s konceptem vytvoření účtu v roli učitele či studenta. Učitelé mohou poskytnout specifický klíč k jejich účtu svým studentům, kteří poté mají možnost editovat časové diagramy (viz obrázek 2.9). Tato myšlenka bohužel není dotažena do konce, díky čemuž mohou po získání klíče studenti provádět na učitelově účtu kromě změny hesla prakticky cokoliv.

Oblíbenost aplikace Preceden v posledních měsících klesá. Z tohoto důvodu autoři upustili od původního omezení bezplatného použití aplikace na tvorbu pouze určitého množství vrstev a událostí. Služba nedisponuje mobilní aplikací.

<sup>14</sup> ovládací prvek grafického rozhraní sloužící pro komunikaci s uživatelem



Obrázek 2.9: Snímek časové osy webové aplikace Preceden.

### 2.5.3 Moment Garden

Služba MomentGarden je webovou a mobilní aplikací sloužící k vytvoření jedné unikátní timeline, určené pro všechny rodiče s malými ratolestmi. Základní vizí projektu MomentGarden je zachytávání důležitých životních událostí dětí a dokumentace jejich růstu a pokroků. Služba dovoluje ukládat fotografie i videa spolu se záznamy a je naprosto zdarma. Je však určena výhradně pro rodiče dětí, čemuž odpovídá i zpracování.

### 2.5.4 Srovnání

Každá z výše uvedených aplikací má své výhody a jasná specifika, díky kterým si vybudovala svou škálu uživatelů. Tyto výhody se však týkají jen pevně daných bodů, na které se zaměřují. Služba Preceden umí vytvářet působivé timeline, avšak pouze skrze své webové rozhraní. MomentGarden nabízí možnost sdílet události mezi platformami. Jedná se ale pouze o události spojené s růstem a vývojem ratolestí. Naproti tomu aplikace Astrid je zaměřená všeobecně. Neřeší však plánování dlouhodobých událostí a jejich sdílení mezi uživateli. TimeTrace se oproti tomu zaměřuje na odvětví, ve kterém se žádná z výše zmíněných aplikací nepohybuje.

## Kapitola 3

# Návrh kalendáře a ukládání dat

V rané fázi vývoje byly prostudovány možnosti tvorby aplikací, způsoby implementace a publikování. Na základě průzkumu byl za cílovou platformu pro tvorbu mobilní aplikace zvolen operační systém Android. Hlavními aspekty byla jeho aktuální rozšířenost, možnost volně publikovat skrze distribuční službu Google Play a také zájem kolegů vlastníčích mobilní zařízení s Androidem o to podílet se na vývoji. Klíčovým bodem bylo zavedení služby, jež umožňuje uživatelům vytvářet, spravovat a sledovat vývoj jejich projektu. Kromě toho byl kladen důraz na sociální stránku aplikace. Za tímto účelem byla pořízena doména [time-trace.eu](http://time-trace.eu) a zajištěn odpovídající webhosting. V rámci něj byla vybudována MySQL databáze. Ta shrnuje informace o uživateli a jejich vzájemných vztazích a plánech. Pro zajištění komunikace mezi mobilní aplikací a databází byla navržena webová služba běžící na vzdáleném serveru. Ta je implementována prostřednictvím skriptovacího programovacího jazyka PHP5. Pro zajištění podpory a poskytnutí základních informací pro uživatele byla navržena webová stránka. K tomu byly využity jazyky HTML a CSS.

### 3.1 Zobrazení kalendáře

Po nastudování možností, jež systém Android s jeho aktualizacemi nabízí, byla zvolena verze 2.2 Froyo, jako minimální podporovaná. Od toho se odvíjel veškerý další návrh. Nevýhodou tohoto rozhodnutí byla absence uživatelského panelu, jež přišel až s verzí 3.0. Řešením problému bylo rozšíření projektu o externí podpůrnou knihovnu ActionBarSherlock, jež vnáší podporu uživatelského panelu i do starších verzí systému.

Z grafické stránky došlo k rozvržení aplikace do 4 celků:

- TimeTrace - Vykreslení kalendáře a jeho prvků
- Profil - Informace o profilu
- Organizace - Správa plánů a vytyčování cílů
- Detail - Zobrazení podrobností o událostech

Každá z těchto částí má vlastní grafické rozvržení. Tato rozvržení byla předmětem konzultací s uživateli a během vývoje prošla mnoha úpravami.

### 3.1.1 Vykreslování

Nejvíce změnami prošlo zobrazení a práce s kalendářem. Záměrem bylo vytvořit intuitivní kalendář, jenž umožní uživateli zobrazit všechny plány a úkoly, které si v rámci své timeline zaznamená on sám nebo jeho přátelé. Původním návrhem bylo vytvořit zobrazení na úrovni týdnů, měsíců, čtvrtletí a roků a ty pak libovolně přepínat skrze kontextové menu. Od tohoto návrhu bylo upuštěno a místo něj byl implementován pinch-zoom<sup>1</sup>. Po jeho aplikaci nastal problém s vykreslováním pravé části obrazovky, jenž začala při oddálení obrazovky blikat. Intenzita blikání se odvíjela od množství položek v kalendáři. Tento jev byl zpozorován pouze na tabletech. Řešením situace bylo omezení rozsahu oddalování obrazovky.

Další důležitou otázkou při návrhu vykreslování byl způsob rozvržení timeline a jejich seskládání. Každý zaznamenaný časový úsek má svůj vlastní přidělený prostor v rámci kalendáře, do kterého mu žádná jiná událost nesmí zasahovat. V rámci tohoto prostoru má neomezené možnosti. Tímto je zajištěno, aby nedocházelo ke kolizím a překrývání událostí. Pro korektní zobrazení kalendáře, jež zaznamenává delší časové období, bylo nutno zvážit možnosti zobrazení. Pokud je časový úsek vykreslovaného plánu delší než rozsah zobrazitelných dnů na displeji, je potřeba rozšířit jeho možnosti. Toto řeší implementace pan/drag<sup>2</sup> v horizontální rovině. Se zvětšujícím se počtem událostí v rámci timeline dochází k pokrytí veškerého zobrazitelného prostoru. Pro rozšíření počtu zobrazitelných událostí byl implementován pan/drag taktéž ve vertikální rovině.

#### Statické versus dynamické vykreslování

Android nabízí pro vytváření svého grafického rozhraní 2 možnosti:

- Vykreslení grafiky a animací skrze systémové vykreslovací procesy
- Přímé kreslení na canvas<sup>3</sup>

První metoda má dosti omezené možnosti a pro vytvoření vlastního kalendáře je nepoužitelná. Proto byla vybrána metoda druhá, která umožňuje absolutní kontrolu nad kreslením. Ta v závislosti na požadavcích umožňuje 2 typy překreslování<sup>[3]</sup>:

- Překreslování v rámci systémového vlákna s využitím volání metody `invalidate()` opětovné vykreslení obrazovky
- Dynamické překreslování ve vlastním vlákně spravovaném třídou `SurfaceView()`

Aplikace TimeTrace je určena pro široké spektrum mobilních telefonů s různými velikostmi displejů. Uživatelé s menšími displeji potřebují s rostoucím počtem událostí s kalendářem manipulovat a aktivně ho procházet. Proto byla zvolena možnost dynamického vykreslování ve vlastním vlákně.

---

<sup>1</sup>zoom obrazovky na základě přibližování a oddalování prstů

<sup>2</sup>rolování obrazovky na základě pohybu prstů

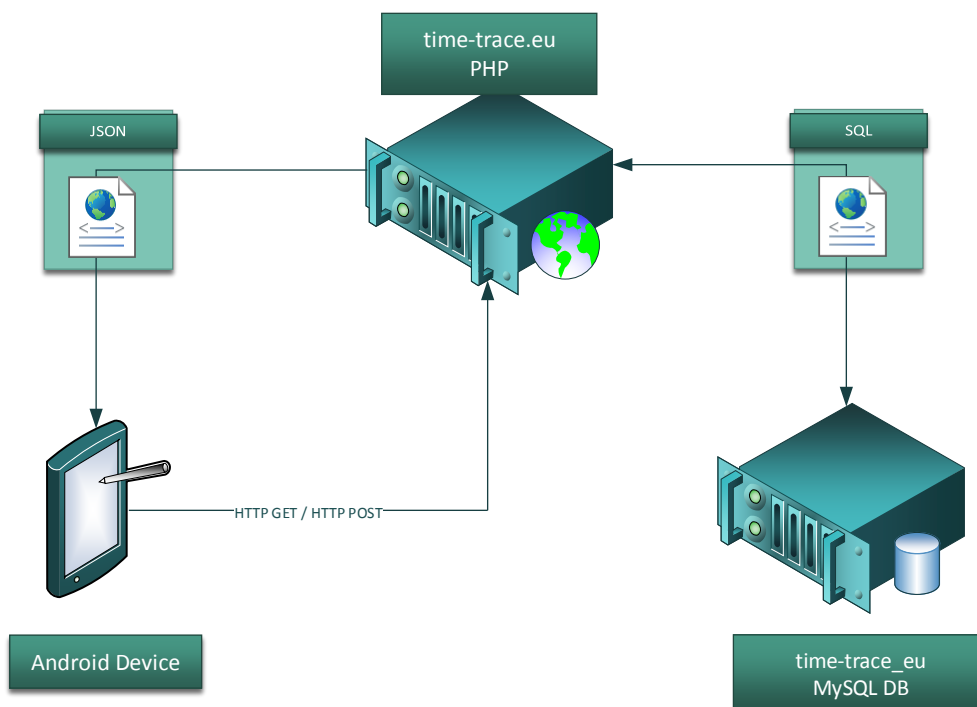
<sup>3</sup>plátno seskupující vykreslené objekty

## 3.2 Komunikace se vzdáleným serverem

Veškerá data a informace o uživatelích, timeline a událostech jsou zaznamenána a uložena ve vzdálené databázi. Proto je vzájemná spolupráce mezi mobilní aplikací a databází pro běh služby nutná. Za účelem komunikace byla navržena webová služba poskytující rozhraní mezi těmito dvěma body.

### 3.2.1 Zpracování na serveru

Pro navržení komunikačního rozhraní mezi aplikací a databází bylo navrženo schéma popisující vzájemnou výměnu dat mezi jednotlivými zdroji (viz obrázek 3.1).



Obrázek 3.1: Schéma komunikace

Veškeré požadavky na získání vzdáleně uložených dat jsou v aplikaci zabaleny a zaslány na server [time-trace.eu](http://time-trace.eu). K tomu je využito dotazovacích metod GET a POST. Metoda GET je využita pro získávání informací o uživatelských účtech, seznamech uložených timeline a detailech událostí. Metoda POST oproti tomu zasílá informaci o nově vytvořených uživatelích, timeline, událostech a veškerých změnách.

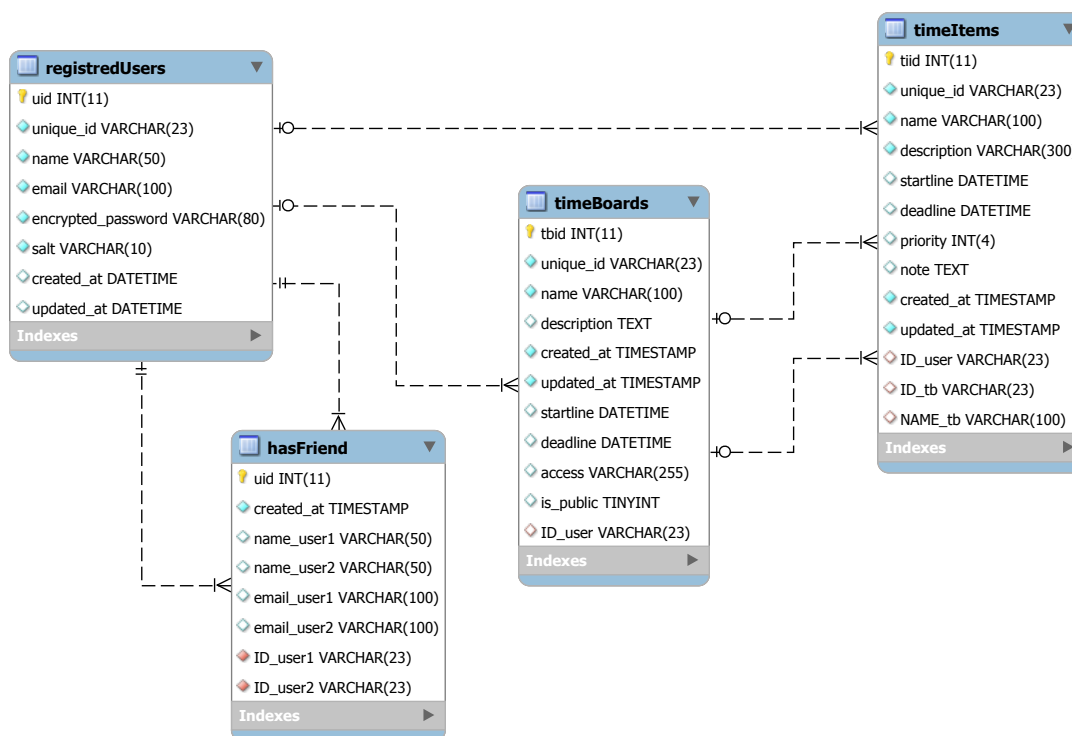
Na serveru běží sada PHP skriptů, která přijímá data z mobilního zařízení. Tato sada data analyzuje a kontroluje jejich formát. Pokud odpovídá definovaným požadavkům tak dojde k vytvoření SQL dotazu. Ten je poté zaslán do databáze, ve které dojde k vyhodnocení daného dotazu a zaslání k ní korespondující odpovědi. Ta je rozebrána na jednotlivé požadované elementy, které jsou zabaleny do objektu ve formátu JSON, jenž je zaslán zpět mobilní aplikaci. V tomto bodě již aplikace data převede do formátu se kterým bude dále manipulovat.

### 3.3 Databáze uživatelů, událostí a stavů

Databáze uživatelů, timeline a událostí byla navržena s ohledem na přímočarost řešení a možnost její okamžité duplikace či přesunu. Mobilní aplikace k databázi přistupuje při veškerých změnách. Tomuto bylo potřeba její návrh koncipovat.

#### 3.3.1 Správa databáze

Celá databáze byla rozvržena do čtyř tabulek (viz obrázek 3.2).



Obrázek 3.2: EER

#### registeredUsers

Tabulka registeredUsers slouží k uložení informací o registrovaných uživateli. Každý z uživatelů má své jedinečné číslo. Toto číslo slouží k jeho identifikaci mezi ostatními záznamy. Druhým unikátním sloupcem tabulky je email. Na jeho základě se uživatel připojuje ke svému účtu. Kromě těchto dvou informací každý nový uživatel definuje své jméno pod kterým bude vystupovat. Na základě tohoto jména ho mohou nálezt jeho přátelé. Mezi další položky patří heslo, sůl<sup>4</sup>, čas vytvoření a poslední modifikace účtu.

#### timeBoards

timeBoard je tabulka všech zaznamenaných timeline. Je definována svým identifikátorem, jménem, popisem, časem vytvoření a modifikace, počátečním a cílovým časem timeline,

<sup>4</sup>náhodně generovaný řetězec použitý při zabezpečování hesla hašovací funkcí



přístupností a přepínačem označujícím, zda se jedná o veřejnou či soukromou časovou osu. Poslední položkou je identifikátor uživatele, jenž tuto tabulku vytvořil.

### **timeItems**

Tabulka `timeItems` uchovává záznamy o všech položkách uložených v rámci timeline. Oproti tabulce časových os ukládá navíc externí poznámku, prioritu a identifikátor spolu se jménem nadřazené timeline.

### **hasFriend**

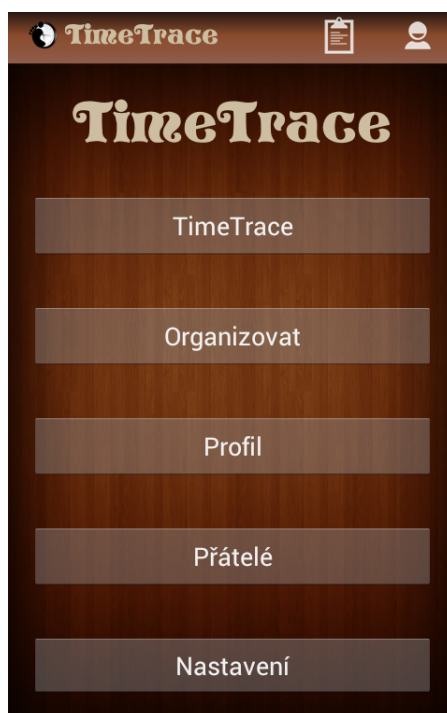
Pod tabulkou `hasFriend` jsou uchována data o všech přátelstvích vytvořených v rámci aplikace `TimeTrace`.

Databáze byla vytvořena v rámci školního serveru `eva`. Po zaregistrování domény byl pořízen hosting a databáze byla přenesena na nový server. Pro tento účel byl vytvořený skript `timetraceDB.sql` s příkazy pro vygenerování a počáteční inicializaci všech tabulek. Ke správě byl použit systém `phpMyAdmin`<sup>[7]</sup> a nástroj `MySQL Workbench`.

## Kapitola 4

# Časový plán a manipulace s událostmi

Aplikace TimeTrace je rozdělena do tří balíčků. První z nich slouží ke správě aplikace, zobrazení dat, registraci a přihlášení uživatelů. Druhý balíček se stará o vykreslování kalendáře a úkolů v něm obsažených. Třetí je sestaven z pomocných tříd pro práci s databázemi, internetovým připojením, nastavením a ukládáním dat pro trvalé přihlášení.

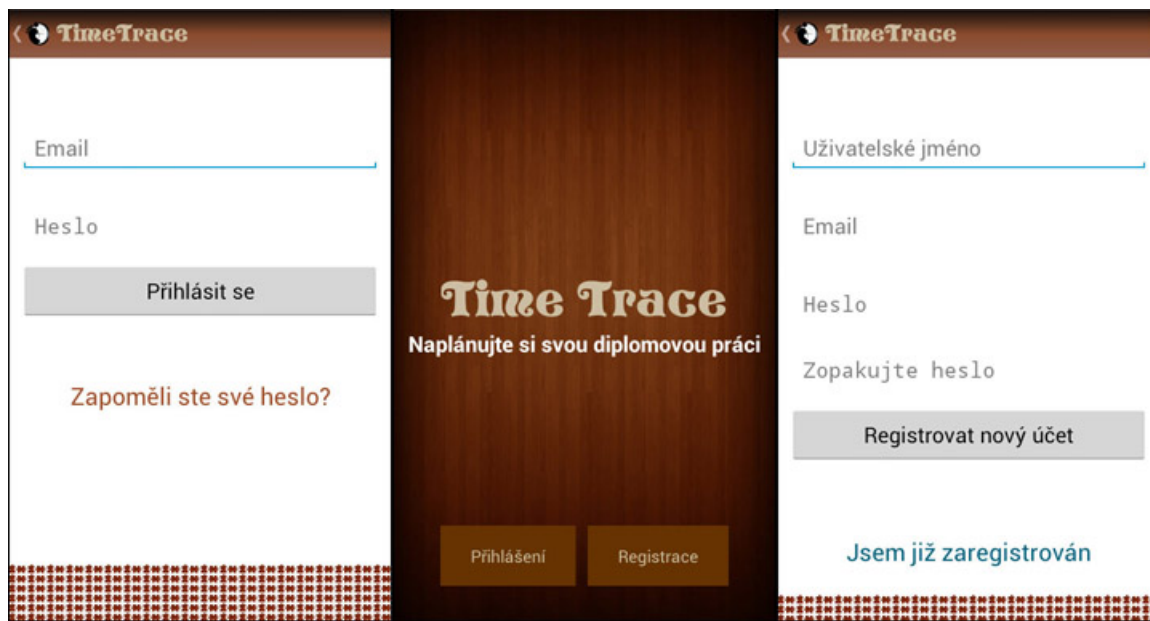


Obrázek 4.1: Menu

### 4.1 Registrace a přihlášení

Úvodní obrazovka, která se každému uživateli po instalaci aplikace zobrazí, nabízí možnost se registrovat nebo přihlásit k uživatelskému účtu. K její realizaci slouží třída `StartActivity`.

Dochází zde ke kontrole dostupnosti internetového připojení. Poté dojde k ověření zda se již uživatel v minulosti přihlásil. To je zjišťováno třídou `SessionControl` uloženou v balíčku s vlastnoručně definovanými knihovny. Ta za tímto účelem využívá `SharedPreferences` a zjištění zda existuje záznam o uživateli, který se ukládá po jeho přihlášení. Pokud tento záznam existuje tak rovnou dochází k přesměrování do zóny jen pro přihlášené a duplikaci získaných informací o uživateli do databáze SQLite. V opačném případě se uživatel musí registrovat či přihlásit prostřednictvím nabídnutých formulářů (viz obrázek 4.2).



Obrázek 4.2: Registrace a přihlášení

## Registrace

Pro registraci nového uživatele je třeba vyplnit 4 základní formuláře :

- Uživatelské jméno - Pojmenování, na základě kterého lze uživatele dohledat za účelem přidání si jej do přátel
- Emailová adresa - Unikátní email vyplňovaný při budoucím přihlašování
- Heslo - Přístupové heslo zašifrované hašovací funkcí SHA-1
- Ověření hesla

Všechny elementy této tabulky prochází ověřením korektnosti a poté jsou zaslány metodou POST na vzdálený server k dalšímu zpracování a uložení do databáze. Pokud je registrace úspěšná, tak dojde k přesměrování do uživatelské zóny.

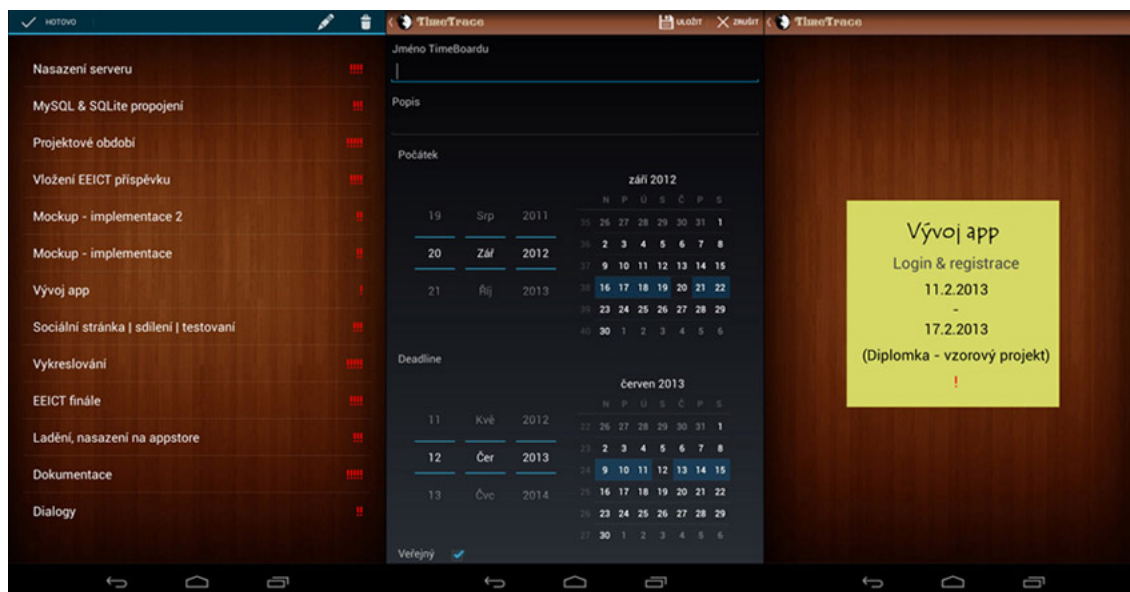
## Přihlášení

K přihlášení slouží jednoduchý formulář požadující emailovou adresu zadanou při registraci a přístupové heslo. Tyto dva údaje jsou zasílány na server, kde dochází k jejich zpracování

PHP skriptem a ověření jejich výskytu v tabulce `registredUsers`. V případě úspěchu jsou získané údaje uloženy do SQLite databáze a dochází k vytvoření záznamu o uživateli, na jehož základě dojde při příštím spuštění k automatickému přihlášení. Poté je uživatel přeměrován do zóny pro přihlášené.

## 4.2 Tvorba a editace událostí

Tlačítkem `Organizovat` se uživatel dostane do části aplikace sloužící ke správě jeho timeline a událostí v nich zahrnutých.



Obrázek 4.3: Tvorba a editace událostí

### 4.2.1 Správa timeline

Při prvním přihlášení dojde po kliknutí na tlačítko `Organizovat` k automatickému přeměrování na obrazovku s formulářem pro vytvoření první timeline (viz obrázek 4.3). Ta vyžaduje definovat jméno, popis a časový rozsah. U něj dochází ke kontrole, aby počáteční datum nebylo definováno za koncovým. Zadané datумы bylo potřeba převádět do formátu `DATETIME`, ve kterém jsou ukládány do MySQL databáze. Uživatel má rovněž možnost zvolit, zda bude jeho timeline veřejný. Ten je poté přístupný všem kdo si přidají uživatele do přátel. Po klepnutí na tlačítko `Vytvořit TimeBoard` jsou všechny údaje získané z formuláře vloženy do seznamu parametrů, který je zaslán pro další zpracování webové služby. Tento proces je vykonáván asynchronně, jelikož je časově náročný a docházelo by k nežádoucímu pozastavení hlavního vlákna. Po dobu komunikace se serverem je vykreslen dialog oznamující, že se vytváří nový timeline. Tento způsob řešení je používán při všech dalších operacích vyžadujících připojení k webové službě. K vytváření dalších timeline slouží tlačítko umístěné v rámci horní lišty.

Po vytvoření timeline dojde k jeho zařazení do seznamu. Po dlouhém kliknutí je nabídnuta možnost její editace a smazání. Rozhodne-li se uživatel smazat timeline, dojde k její odstranění z databáze a s ní i všech událostí které pod ní spadají.

## 4.2.2 Správa událostí

Kliknutím na timeline dojde k jeho výběru ze seznamu a k vypsání všech úkolů, které jsou v něm zahrnuty. Každý úkol je definován svým jménem a prioritou. Tak je definována vykřičníky. Jejich počet se s rostoucí prioritou zvyšuje. Nejdůležitější úkoly jsou definovány pěti vykřičníky. Správa úkolů je obdobná jako u timeline. Uživatel má možnost si zvolit zda je vytvářený úkol jednorázový a jedná se tedy o takzvaný milník. Pokud ano, ta je datum konce nastaveno stejně jako datum počátku. V ostatních případech je mu po kliknutí na odrážku vedle položky **Deadline** aktivována možnost si jej sám určit. Aktivita pro vytváření a editaci událostí přijímá od nadřazené aktivity identifikátor, jméno a časový rozsah dané timeline pod kterou úkol spadá. Při definování události před počátkem či po konci timeline by došlo k nekorektnímu vykreslování kalendáře. Explicitní kontrolou je tomuto zabráněno. Tato nastavení jsou realizována prostřednictvím dialogů.

Novinkou, jež se byla implementována ve verzi 1.0.3 je podpora přidání nové události do aplikace Google Calendar, která je v současné době implementována jako výchozí kalendář systém Android. K tomu je využito rozhraní Google Calendar API. Údaje získané z formulářů jsou zaslány dodavateli obsahu, jež je zpracuje a vloží jako novou událost do uživatelského kalendáře. Tato možnost přidat událost do kalendáře byla implementována na základě požadavků od uživatelů. Zatím je pouze nabídnuta při vytváření nové události.

Nově vytvořená událost je ihned vložena do databáze. Díky tomu mají ostatní uživatelé možnost bez jakékoliv prodlevy vidět všechny změny v rámci veřejných timeline. Stiskne-li uživatel dlouze požadovanou událost v seznamu, nabídne mu aplikace možnost její úpravy či odstranění. To je realizováno v podobě kontextového módu, jež upravuje zobrazení horní lišty, do které jsou vloženy odkazy na aktivity pro editaci a smazání událostí.

## 4.2.3 Detail události

Všechny události nabízí možnost zobrazení jejich kompletních detailů. To je realizováno třídou `OneEventOfTimeTrace`. Ta přijímá z předcházející aktivity identifikátor požadované události, identifikátor nadřazené timeline a detaily o všech událostech uživatele uložené v asociativním poli záznamů. Po vytvoření třídy je na základě identifikátoru vybrán požadovaný záznam a pro něj odpovídající detaily. Následně je vyslán požadavek na získání seznamu registrovaných uživatelů, ze kterého je vybrán tvůrce události. Všechny údaje jsou převedeny do čitelné formy a vykresleny.

## 4.3 Sociální vazba

Všichni přihlášení uživatelé mají možnost pod položkou **Přátelé** zobrazit seznam jejich kamarádů (viz obrázek 4.4). Po kliknutí na položku **NAJÍT PŘÁTELE** je vytvořena třída `ListFriends`. V ní dojde k požadavku na získání seznamu přátel a seznamu všech uživatelů z databáze. Ze soupisu přátelství jsou vyjmuty a uloženy identifikátory kamarádů. Poté dojde k vykreslení seznamu všech uživatelů kromě těch, jež jsou definováni identifikátory a aktuálně přihlášeného. Při výběru některého z uživatelů je vykreslen dialog s možností si jej přidat do přátel. V případě potvrzení je vytvořena nová třída `CreateFriendship`. V ní jsou vloženy identifikátory, jména a emaily obou uživatelů do seznamu parametrů. Ten je poté zaslán metodou `POST` k dalšímu zpracování na server, kde je vytvořen SQL dotaz s žádostí o vytvoření nového řádku v tabulce `hasFriend`. Nazpět je zaslán `JSONObject` obsahující řetězec `success`. V případě úspěchu je na řetězec mapována hodnota "1".

Poté je na moment vykresleno okno informující uživatele o výsledku operace.

V reakci na výběr kamaráda dojde k vykreslení seznamu všech veřejných timeline dotyčného. S přáteli lze kooperovat 2 způsoby:

### Zobrazení kamarádovy timeline včetně všech událostí v něm obsažených

- K zobrazení dojde po krátkém kliknutí na požadovanou timeline
- Zobrazit lze pouze veřejnou timeline

### Přidání nové události do kamarádovy timeline

- K zobrazení dojde po dlouhém kliknutí na požadovanou timeline
- Uživatel nemá oprávnění žádným způsobem měnit timeline ani události vytvořené někým jiným. To je kontrolováno ve třídě `EditTimeTrace` porovnáním identifikátoru tvůrce dané události s identifikátorem aktuálně přihlášeného uživatele, o kterém jsou uloženy všechny detaily v SQLite databázi



Obrázek 4.4: Seznam přátel

## 4.4 Vykreslení časových plánů

První tlačítko na úvodní obrazovce (`TimeTrace`) slouží k vykreslení kalendáře úkolů (viz obrázek 4.5). Po kliknutí na něj se vytvoří nová třída `WelcomeCalendar`. Ta získává z třídy jí předcházející dva řetězce. Prvním z nich je identifikátor požadované timeline. Druhý řetězec slouží k rozpoznání, zda uživatel pracuje s vlastní nebo kamarádovou timeline. Před samotným vykreslením jsou vytvořeny třídy `LoadAllTimeboards` a `LoadTimeItems`. Zde se první z tříd dělí na dvě části. Vykresluje-li se vlastní timeline, načtou se detaily i o ostatních, které uživatel vytvořil. Jejich jména jsou poté uložena do seznamu a po kliknutí na `VYBRAT TIMEBOARD` si uživatel může jeden z nich vybrat. V případě vzdálené timeline dojde pouze k načtení požadovaných detailů. Třída `LoadTimeItems` poté na stejném principu zašle metodou `GET` identifikátor požadované timeline PHP skriptu. Ten zašle SQL dotaz k získání všech událostí pod něj spadajících. Odpověď poté pošle zpět aplikaci jako objekt složený z polí ve formátu JSON. Tato data jsou nahrána do seznamu polí. Veškeré operace navazující spojení se vzdáleným serverem probíhají asynchronně mimo hlavní vlákno.

Po načtení všech potřebných dat z databáze dochází k samotnému vykreslování, jenž je zajišťováno třídou `DrawView`. Ta implementuje třídu `SurfaceView`, která zprostředkovává dynamické vykreslování kalendáře. Jako první dojde k inicializaci třídy sledující gesta uživatele a reagující na každý pohyb jeho prstu po vykreslovací ploše displeje. Druhým bodem je vytvoření třídy spouštějící nové vlákno. Ta se taktéž stará o jeho správu a pozastavení v případě potřeby.

### Vykreslovací vlákno

Vykreslení plochy je prováděno v rámci jediného vlákna. To se spouští ve smyčce metodu `doDraw()`.

V rámci `doDraw()` jsou prováděny operace v následujícím pořadí:

1. Uložení aktuálního stavu canvasu
2. Nastavení měřítka a výchozích souřadnic
3. Zjištění aktuálního rozlišení displeje
4. Zapnutí vyhlazování a nastavení výchozího počtu dnů zobrazitelných na displeji
5. Procházení jednotlivých událostí uložených v seznamu polí.
  - (a) Výpočet počtu dnů od počátku timeline do začátku události
  - (b) Výpočet počtu dnů od počátku události do konce události
  - (c) Vykreslení barevného obdélníku reprezentujícího událost do místa určeného na základě předchozích výpočtů
  - (d) Vykreslení textového pole se jménem události do středu barevného obdélníku z předcházejícího bodu
6. Nastavení barvy pro horní lištu zobrazující kalendářní dny
7. Výpočet počtu událostí přesahujících možnosti zobrazení na displej v jednom momentě

8. Vykreslení vertikálních čar v závislosti na počtu dnů timeline
9. Nastavení barvy pro aktuální den
10. Vykreslení názvů a dat dnů kalendáře do horní lišty
11. Načtení uloženého canvasu

Zaznamenávají se souřadnice pozic, na kterých došlo k navázání a rozvázání kontaktu s vykreslovací plochou. S každým posunutím prstu po vykreslovací ploše jsou přepočítávány souřadnice pohybu. Pozice, na které jsou vykreslovány jednotlivé události jsou určovány na základě aktuálního zobrazení kalendáře a vypočtených souřadnic.



Obrázek 4.5: Kalendář událostí

## 4.5 Podpůrná webová služba

Na vzdáleném serveru běží sada 20 PHP skriptů a webová stránka, jež dohromady tvoří podpůrnou webovou službu.

Skripty lze dle jejich účelu rozdělit do 3 kategorií:

- Konfigurace (`config.php` a `DB_Connect.php`). Využívají se pro připojení se k MySQL databázi a jejím následném výběru. Jsou spouštěny před každou operací. Umožňují snadný přesun databáze zprostředkovaný pouhou změnou těchto souborů.
- Přihlášení a registrace uživatelů (`index.php` a `DB_Functions.php`). Jsou využívány pro manipulaci s uživatelskými účty. Dochází v nich ke kontrole uživatelů a šifrování a dešifrování hesel.
- CRUD<sup>1</sup> operace

<sup>1</sup>operace pro vytvoření, čtení, editaci a smazání záznamu



## CRUD

CRUD skripty plní roli zprostředkovatelů spojení mezi serverem a databází.

Skripty začínající předponou `create` přijímají pomocí metody POST sadu paramterů, jež poté vkládají příkazem INSERT do vybrané tabulky v podobě nového řádků. Aplikaci zpětně informují zasláním objektu ve forátu JSON, jež obsahuje klíč `success`. V případě úspěchu je na tento klíč mapována hodnota "1". V opačném případě hodnota "0".

Předponou `get` začíná množina skriptů přijímajících metodou GET sadu parametrů. Na základě těchto parametrů dochází k vykonání příkazu SELECT nad vybranou tabulkou. Odpověď je uložena do pole hodnot, které je spolu s položkou `success` odesláno jako JSONObject zpět mobilní aplikaci.

Na předponu `update` začínají skripty, jež využívající příkaz UPDATE k editaci parametrů definovaných metodou POST. V každém změněném řádku navíc nastavují hodnotu sloupce `updated_at` na aktuální čas. Aplikaci odpovídají obdobně jako první typ skriptů.

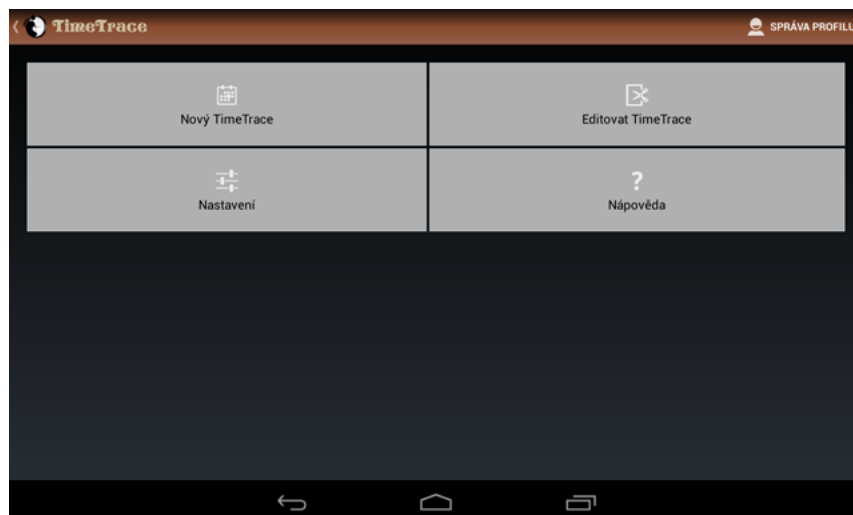
`delete` je předpona pojmenovávající skripty vytvořené za účelem mazání záznamů. Využívají pro tento účel příkazu DELETE. V ostatních bodech se podobají skriptům předešlým.

## 4.6 Testování

Po celou dobu tvorby aplikace pro plánování úkolů byl kladen důraz na hledání uživatelů pro vzájemnou spolupráci. Jedná se o aktivní uživatele mobilních aplikací využívající zařízení opatřená operačním systémem Android. Tito uživatelé, tzv. Early Adopters, mají od začátku zájem o službu poskytovanou aplikací. Z důvodu zaměření samotné aplikace se jedná z velké části o studenty navštěvující vysoké školy (především technického typu), kteří se chtějí podílet na vývoji a testování aplikace, jež jim pomůže sledovat vývoj týmových projektů na nichž pracují se svými kolegy. Early Adopters byli vybráni z řad kolegů, příbuzných a potvrzených zájemců ze studentských diskusních fór. Tito lidé se podílejí na vývoji celého projektu a je jim umožněn přístup k oficiálně nevydaným vývojářským verzím projektu.

### 4.6.1 Testování v průběhu vývoje

Během dvou semestrů vývoje prošla aplikace a s ní svázaná databáze řadou změn. Většina těchto úprav byla vykonána na základě požadavků na vyvíjenou službu. Po vytvoření první provozuschopné verze došlo k jejímu rozšíření mezi testery. První modifikací byla změna grafického návrhu, jež byl oproti oficiálně publikované verzi značně odlišný (viz obrázek 4.6). K obměnám došlo také v rámci databáze. Ta v původní verzi neposkytovala možnost vytváření přátelství ani veřejnou publikaci timeline.



Obrázek 4.6: Původní grafický návrh

Aplikace byla primárně vyvíjena pro mobilní telefony a nevyužívá fragmentů, které se používají při vývoji pro tablety. Po implementaci kalendáře bylo na základě testů zjištěno, že na žádném z tabletů nedochází ke korektnímu grafickému vykreslování. Zajímavé bylo, že na žádném z testovaných mobilních telefonů se tento problém nevyskytl (viz tabulky testovaných zařízení 4.1 a 4.2). Na základě tohoto zjištění byla modifikována část čínicí největší potíže. Tablety s nejmarkantnějšími zpozorovanými chybami byly zařazeny do seznamu nepodporovaných zařízení. Toto řešení je dočasné a hledá se příčina způsobující tuto chybu.

Zařízení	Verze	Rozlišení
HTC Desire S	4.0.4	800 x 480
Samsung Galaxy Nexus	4.2.2	1280 x 720
LG Optimus 2X P990	4.0.4	800 x 480
Samsung Galaxy I	4.2.2	1280 x 720
Samsung i9000 Galaxy S	CyanogenMod 10.0	800 x 480

Tabulka 4.1: Otestované mobilní telefony

Zařízení	Verze	Rozlišení
Google Nexus 7	4.2.2	1280 x 800
GoClever T76GPS	4.0.3	800 x 480
Fuzhou Rockchip	4.0.3	1024 x 768
GoClever Tab A104.2	4.0.4	1024 x 600

Tabulka 4.2: Otestované tablety

## 4.6.2 Závěrečné testování

Před nasazením mobilní aplikace a webové služby do ostrého provozu byl proveden přesun databáze a veškerých skriptů ze školního serveru **eva** na server veřejný. Do aplikace bylo implementováno Google Analytics SDK for Android pro detailní sledování počtu přístupů a chování uživatelů.

Mezi hlavní monitorované body patří :

- počet kliknutí na tlačítko **Registrovat**
- počet vykreslení úvodní obrazovky
- počet vykreslení kalendáře
- počet nově vytvořených timeline
- počet uživatelů přidanych do přátel

V posledním měsíci vývoje byla aplikace publikována prostřednictvím portálu Google Play. Pro zviditelnění byla vytvořena podpůrná webová stránka [time-trace.eu](http://time-trace.eu), odkazující na stažení a nabízející základní informace o aplikaci. Stránka spolu s odkazem byla zveřejněna na sociálních sítích a veřejných fórech zabývajících se danou tematikou.

Za účelem testování byla vytvořena timeline určená pro studenty pracující na diplomové práci. Timeline byla pojmenována jako "Diplomka - Vzorový projekt" (viz obrázek 4.7). Její viditelnost byla nastavena jako veřejná. Tím získal každý nově registrovaný uživatel možnost do ní nahlédnout a vložit do ní informace o svém pracovním postupu. Množství stažení a reakcí bylo negativně ovlivněno faktem, že v době veřejné publikace velká část studentů vyvíjejících své diplomové práce byla již v poslední fázi vývoje a intenzivně se věnovala psaní textových zpráv. Ze zjištěných výsledků a konzultací vyplynulo, že se podstatné části uživatelů nelíbí nutnost se pro používání aplikace registrovat. Taktéž ne všichni mají stabilní dostupné připojení k internetu, jež by jim dovolovalo aplikaci kdykoliv využít.



Obrázek 4.7: Vzorový projekt

Důležitým pomocníkem při testování je možnost odesílání zprávy o chybách vzniklých při pádu aplikace. Došlé zprávy je možné zobrazit prostřednictvím Google Play Developer Console, která slouží vývojářům ke správě aplikací nahraných na portál Google Play. V reakci na tyto aspekty byly v posledních týdnech vývoje vydávány nové verze programu odstraňující zjištěné chyby.

Přehled verzí:

- 1.0.1 - opravena chyba ve špatném zobrazení barvy panelu kalendáře způsobující pád anglicky lokalizované aplikace
- 1.0.2 - opravena chyba způsobující pád aplikace při výpadku internetového připojení během tvoření nové timeline
- 1.0.3 - dokončena kompletní lokalizace do angličtiny - přidána podpora synchronizace nové události s aplikací Google Calendar
- 1.0.4 - opravena chyba ve vykreslování při vykreslení méně než 10 dnů v kalendáři

## Kapitola 5

# Závěr

Práce se zaměřuje na pomoc studentům a vedoucím diplomových prací sledovat pokrok ve vývoji za určitá časová období, jež je zobrazovaný formou timeline. Vedoucí diplomových prací tímto mohou sledovat vývoj svůj a svých studentů, kteří disponují zařízením s operačním systémem Android. Hlavním úspěchem je realizace kalendáře a propojení všech částí aplikace se vzdálenou databází, což umožňuje okamžitě reflektovat veškeré změny. Podařilo se opravit počáteční nedostatky a mobilní aplikaci spolu s webovou službou uvést do funkčního stavu, jež je prezentován uživatelům. Kdybych na začátku práce věděl, jaké problémy mohou nastat s potřebou neustálého připojení k internetu a s nelibostí uživatelů k registraci nových účtů, tak bych při návrhu zavedl taktéž offline mód s využitím lokální databáze, jež bych do aplikace implementoval.

Práci jsem se naučil vyvíjet mobilní aplikace se všemi aspekty s tím spjatými. Pochopil jsem principy dynamického vykreslování využívaného v mobilních zařízeních a využil jsem znalosti o návrhu a práci s databázemi v praxi. Taktéž jsem si osvojil některé základní metody startupu při samotném vývoji.

Vzhledem k zájmu uživatelů a taktéž mém osobním zájmu v tomto směru chci ve správě aplikace a jejím dalším vývoji pokračovat i nadále. Dalším cílem je rozšířit aplikaci mezi širší spektrum lidí, zajistit podporu pro tablety a na základě požadavků implementovat možnost plné synchronizace s aplikací Google Calendar.

# Literatura

- [1] Android Version History [online]. 2013 [cit. 2013-05-5].  
URL <http://galaxyscientific.com/android-version-history>
- [2] App Framework [online]. [cit. 2013-05-10].  
URL <http://developer.android.com/about/versions/index.html>
- [3] Canvas and Drawables [online]. [cit. 2013-05-7].  
URL  
<http://developer.android.com/guide/topics/graphics/2d-graphics.html>
- [4] Managing the Activity Lifecycle [online]. [cit. 2013-05-8].  
URL <http://developer.android.com/guide/components/activities.html>
- [5] Storage Options [online]. [cit. 2013-05-8].  
URL <http://developer.android.com/guide/topics/data/data-storage.html>
- [6] JSON.ORG Introducing JSON [online]. [cit. 2013-5-12].  
URL <http://json.org/>
- [7] Delisle, M.: *phpMyAdmin: Efektivní správa MySQL*. Brno: Zoner Press, 2004, 264 s., 80-868-1509-9.
- [8] Gilmore, W.: *Velká kniha PHP a MySQL 5: Kompendium znalostí pro začátečníky i profesionály. Vyd. 1. [i.e. 2. vyd.]*. Brno: Zoner Press, 2007, 504-505 s., iISBN 80-86815-53-6.
- [9] Gilmore, W.: *Velká kniha PHP a MySQL 5: kompendium znalostí pro začátečníky i profesionály. Vyd. 1. [i.e. 2. vyd.]*. Brno: Zoner Press, 2007, 31-32 s., iISBN 80-86815-53-6.
- [10] Maurya, A.: *Running lean: Iterate from Plan A to a Plan That Works*. Sebastopol: O'Reilly, 2012, 207 s., 978-1-449-30517-8.
- [11] Murphy, M. L.: *Android 2: Průvodce programováním mobilních aplikací*. Brno: Computer Press, 2011, 27-28 s., 978-80-251-3194-7.
- [12] Murphy, M. L.: *Android 2: Průvodce programováním mobilních aplikací*. Brno: Computer Press, 2011, 174-175 s., 978-80-251-3194-7.

# Příloha A

## Obsah CD

- Zdrojové kódy implementované mobilní služby - /src-mobile/
- Zdrojové kódy implementované webové služby - /src-web/
- Skript na vytvoření a inicializaci databáze - /src-sql/
- Technická zpráva ve formátu pdf - /thesis-pdf/
- Zdrojové kódy technické zprávy v L<sup>A</sup>T<sub>E</sub>X - /thesis-tex/
- Manuál - /manual/