

**Obchodní akademie, Vyšší odborná škola a Jazyková škola
s právem státní jazykové zkoušky Uherské Hradiště**



MATURITNÍ ČETBA

Prohlášení:

Souhlasím s tím, že s výsledky mé práce může být naloženo podle uvážení vedoucího maturitní práce a ředitele školy. V případě publikace budu uveden jako spoluautor.

Prohlašuji, že jsem na celé maturitní práci pracoval samostatně a veškeré použité zdroje jsem citoval.

V Uherském Hradišti, dne

podpis absolventa

RESUMÉ

Tento projekt je určen pro studenty SŠ, kteří budou maturovat z ČJ. Pomůže to studentům k vybrání si maturitních knížek, které budou vypsány v knihovně.

OBSAH

ÚVOD.....	8
1 TECHNICKÁ DOKUMENTACE.....	9
2 FIREBASE	10
2.1 Authentication	10
2.2 Realtime Database	10
2.3 ER model databáze	11
3 MOBILNÍ APLIKACE	12
3.1 Spuštění	12
3.1.1 Spuštění v Android studiu (pro vývojáře)	12
3.1.2 Spuštění aplikace ve vlastním mobilním zařízení	12
3.2 Použité vývojové prostřední a SDK	13
3.2.1 Android Studio	13
3.2.2 Flutter SDK 1.22.5	13
3.2.3 Dart SDK 2.10.4	13
3.3 Ovládání.....	14
3.4 Důležité třídy	16
3.4.1 Třída AuthenticationService & model GetUser	16
3.4.2 Třída DatabaseService	16
3.4.3 Třída MyApp	17
4 WEBOVÁ STRÁNKA.....	18
4.1 Spuštění	18
4.1.1 Spuštění na localhostu (pro vývojáře)	18
4.1.2 Spuštění na webu	18
4.2 Použité knihovny a frameworky	18
4.2.1 CodeIgniter 3	18
4.2.2 Bootstrap.....	18
4.2.3 Popper.JS	18
4.2.4 Firebase Admin PHP SDK	19
4.2.5 Composer.....	19
4.3 Diagram tříd.....	19
4.4 Struktura souborů	20
4.5 Controllers	20
4.5.1 Application	20

4.5.2	Deniky_controller.....	20
4.5.3	Home_cotntroller.....	21
4.5.4	Knihy_controller.....	21
4.5.5	User_controller.....	22
5	ZÁVĚR	23
6	ZDROJE	24

SEZNAM OBRÁZKŮ

Obrázek 1 Ukázka databáze	10
Obrázek 2 ER model	11
Obrázek 3 Home page.....	14
Obrázek 4 Login.....	14
Obrázek 5 Deník	14
Obrázek 6 Diagram tříd - aplikace	15
Obrázek 7 GetUser.....	16
Obrázek 8 User auth.....	16
Obrázek 9 Diagram tříd - web.....	19

ÚVOD

Tento projekt je určen pro studenty SŠ, kteří budou maturovat z ČJ. Pomůže to studentům k vybrání si maturitních knížek, které budou vypsány v knihovně.

Uživatel se musí přihlásit na svůj účet. V knihovně budou zapsané základní informace o těchto knihách, jako je autor, žánr, počet stran, stručný obsah. Studenti budou moct procházet si různé knihy, od různých autorů. K pomoci vybrání knihy, v daném období budou moct studenti na začátku listu knih si vybrat zrovna to období, ve kterém si chtějí vybrat knihu. Dále si v aplikaci budou moct zapsat nějaké základní informace a poznámky o knize, kterou zrovna přečetli, nebo čtou. Také si zde budou moct procházet svoje už sepsané deníky a zároveň je převést do pdf a stažení tohoto pdf.

Webové stránky budou fungovat velice podobně jako aplikace, smyslem je uživatelům nabídnout používat služby aplikace i na webu, web a aplikace budou vzájemně propojeny pomocí účtů, který si uživatel vytvoří. Webové stránky nabídnou možnost procházet maturitní četbu uživateli školy, základní informace o jednotlivých knihách formou knihovny, možnost si vybrat oblíbenou četbu a případně si o ní vpsat vlastní poznámky formou deníku a web bude uživateli doporučovat knihy podle již vybraných žánrů, autorů, a další kritéria.

Klíčovou technologií tedy bude databáze schopná provázat aplikace a webové stránky, proto jsme si zvolili službu Firebase a to především její funkce Authentication, která zaručí bezpečné řízení uživatelských účtů a funkci Realtime Databse, tedy NoSql databázi, ve které budou uloženy jak knihy tak uživatelské zápisky přístupné jak z mobilní aplikace tak z webových stránek.

Výsledným produktem tedy bude mobilní aplikace a webové stránky, na kterých bude mít uživatel svoje zápisky přístupné a synchronizované.

Jako studenti střední školy, jsme při zadávání projektu objevili, že žádná podobná aplikace přizpůsobená školám není k dostání. Tohle byl taky jeden z důležitých faktorů, při konečném ustanovení formy projektu.

V následujících stranách technické dokumentace můžete prohlédnout informace o použité službě Firebase a také samotné mobilní aplikaci a webové stránce.

Tato mobilní a zároveň webová aplikace je sama o sobě unikátní, jelikož neexistuje žádná ji podobná aplikace. Jak v českém, tak anglickém jazyce. Z toho důvodu nejde aplikaci kvalitativně porovnat s ostatními.

1 TECHNICKÁ DOKUMENTACE

Technická dokumentace je rozdělena na tři části: Firebase, webovou stránku a aplikaci.

2 FIREBASE

Firebase je moderní NoSql databáze, na struktury používá třídy a ukládá je jako JSON. Jedná se o službu od společnosti Google a nabízí své služby v limitované verzi zdarma. Jedná se o klíčovou technologii celého projektu, umožňuje synchronní propojení mobilní aplikace a webové stránky a je uložště všech účtů a dat aplikace tak dat uživatelsky vytvořených. Pro náš projekt jsme využili funkcí Authentication a Realtime Database.

2.1 Authentication

Jedná se o funkci Firebase, která obstarává účty uživatelů a to jak registraci tak přihlášení uživatelů. Umožňuje vytvořit uživatele za pomoci zvolené metody v našem případě za pomoci e-mailové adresy a uživatelského hesla. Funkce Authentication se poté stará o bezpečné zahashování a uschování uživatelských hesel. Funkce Authentication také každému uživateli přiřadí uid, které poté hojně využíváme pro zobrazování a ukládání obsahu danému uživateli a to pomocí funkce Realtime Database.

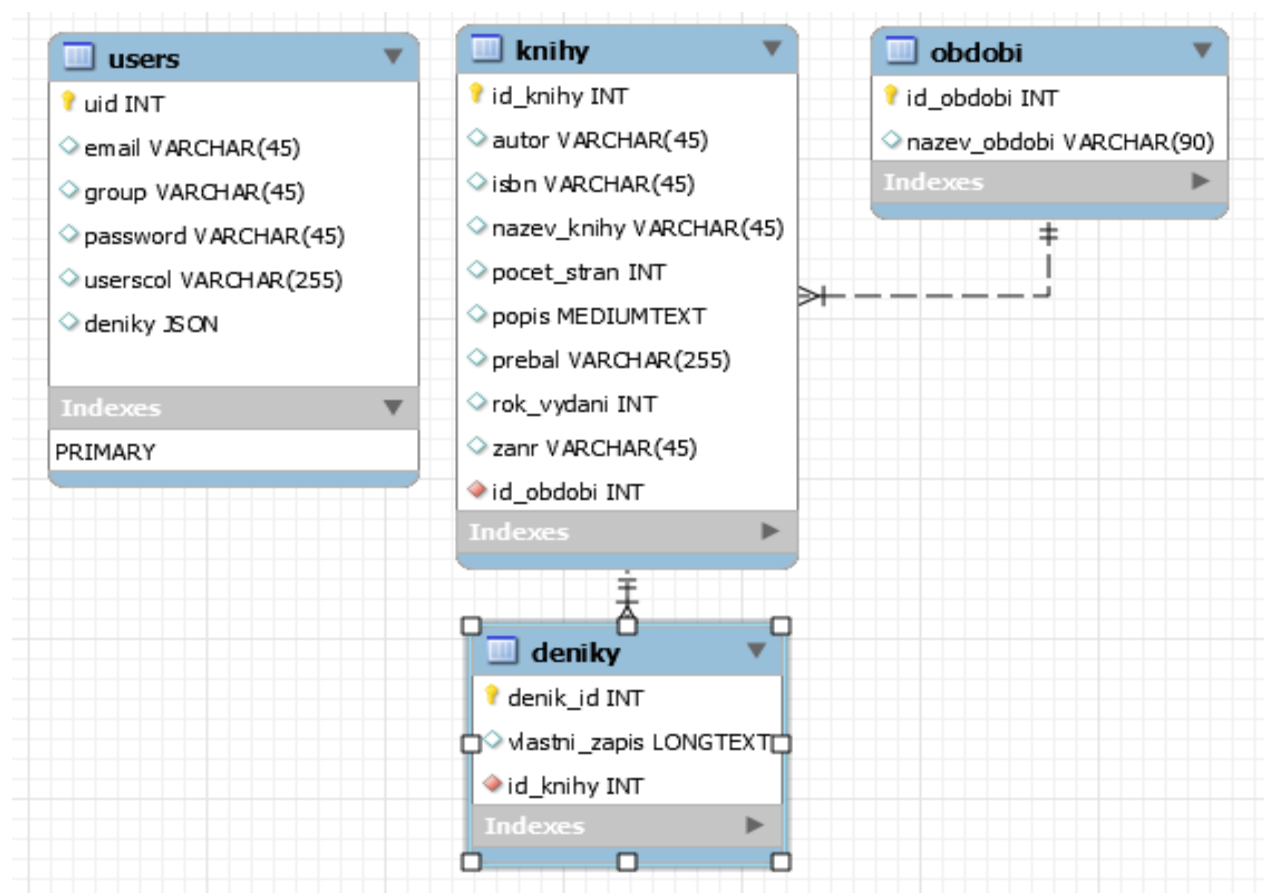
2.2 Realtime Database

Jedná se o funkci Firebase, která ukládá všechna data projektu v NoSql databázi. Zde jsou uloženy všechny knihy, některé informace o uživateli, které funkce Authentication neukládá a všechny uživatelsky vytvořená data. Uživatelské data jsou rozlišována podle uid, které je otiskem uid z funkce Authentication, tudíž umožňuje zobrazovat správný obsah správnému přihlášenému uživateli. Struktura NoSql Realtime Database pro náš projekt je:

```
{
  "skola": {
    "book_id": {
      "autor": "",
      "id_knihy": "",
      "isbn": "",
      "nazev_knihy": "",
      "pocet_stran": "",
      "popis": "",
      "prebal": "",
      "rok_vydani": "",
      "zannr": ""
    },
    "denik": {
      "uid": {
        "denik_id": {
          "author": "",
          "book": "",
          "genre": "",
          "isbn": "",
          "notes": "",
          "období": "",
          "pocet_stran": ""
        }
      }
    }
  },
  "users": {
    "uid": {
      "name": "",
      "surname": ""
    }
  }
}
```

Obrázek 1 Ukázka databáze

2.3 ER model databáze



Obrázek 2 ER model

Databáze využívá celkem ze 4 tabulek společně s funkcí Authentication, kterou přímo Firebase podporuje. V tabulce users jsou uloženy informace o uživateli. Tyto informace jsou uloženy právě v [Authentication](#). Dále je zde tabulka Knihy, ve které jsou uloženy všechny důležité informace ohledně knih, které jsou v databázi. Tyto knihy se řadí podle období, kdy v poměru n:m jsou tyto tabulky propojeny. Nakonec je zde tabulka deniky, ve které si uživatel zapisuje vlastní zápisky ke knize.

3 MOBILNÍ APLIKACE

3.1 Spuštění

3.1.1 Spuštění v Android studiu (pro vývojáře)

Projekt otevřeme pomocí vývojového prostředí Android Studio, budeme potřebovat:

- Pluginy pro Android Studio „Flutter“ a „Dart“
- SDK soubory pro Flutter a Dart
- Android emulátor API level alespoň 16
- Musíme nastavit cestu k SDK souborům v File->Setting...->Languages & Frameworks
- Zde nastavíme cestu k SDK souborům pro Flutter i Dart
- Stáhněte všechny soubory a rozbalte je do vaší požadované složky

3.1.2 Spuštění aplikace ve vlastním mobilním zařízení

Aplikaci je možné v současné verzi spustit pouze na Android zařízeních.

Pro instalaci a spuštění aplikace postupujte následovně:

1. Přejděte na odkaz: https://github.com/JakubKatrnak/Maturitni_cetba/tree/main/product
2. Stáhněte si soubor Maturitní četba.apk
3. Povolení z cizích zdrojů:
 - a. Pokud nemáte povolené ve Vašem zařízení instalaci z cizích zdrojů, tak by se Vám mělo zobrazit vyskakovací okno s příslušným tlačítkem pro povolení, které Vás přesměruje do nastavení
 - b. Pokud si nejste jisti, že máte povolenou instalaci z cizích zdrojů, ale žádné vyskakovací okno se Vám nezobrazí, tak přejděte do **Nastavení->Zabezpečení->Neznámé zdroje/Instalovat neznámé aplikace** a povolte instalaci např. z prohlížeče Google, nebo z jiného, který používáte.
4. Po povolení instalace z cizích zdrojů přejděte zpátky do složky se staženým souborem a soubor už pouze nainstalujte.
5. Aplikaci můžete spustit a bude plně funkční.

3.2 Použité vývojové prostřední a SDK

3.2.1 Android Studio

- Verze: 4.1.1
- Vývojář: Google
- Ke stažení: <https://developer.android.com/studio>
- Licence: Apache Licence 2.0
- Pluginy:
 - Dart 201.9245
 - Flutter 1.22.6
- Knihovny:
 - cupertino_icons 0.1.3
 - carousel_slider 2.3.1
 - loading_animations 2.1.0
 - google_fonts 1.1.2
 - flutter_spinkit 4.1.2
 - flutter_svg 0.19.1
 - firebase_database 4.4.0
 - firebase_core 0.5.3
 - firebase_auth 0.18.4
 - provider 4.3.3

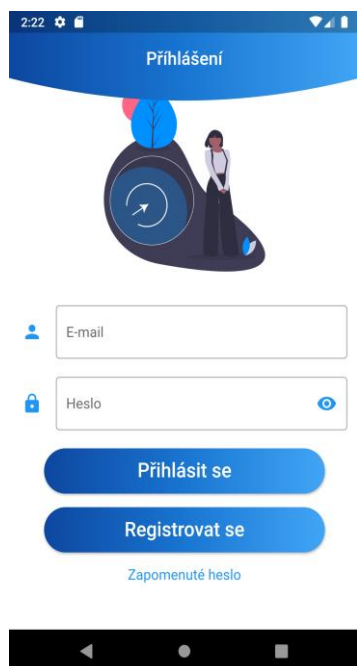
3.2.2 Flutter SDK 1.22.5

- Verze: 1.22.5
- Vývojář: Google
- Ke stažení: <https://flutter.dev/docs/get-started/install>
- Licence: BSD licence

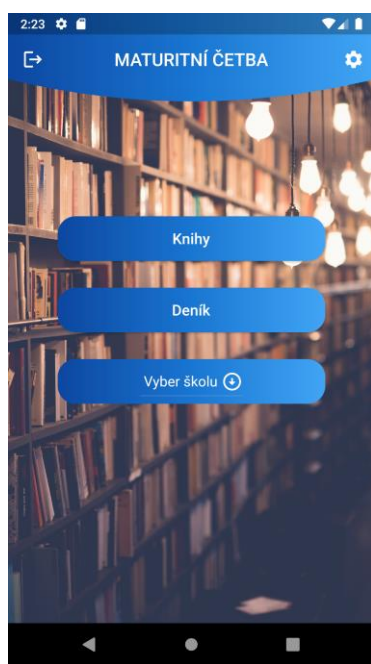
3.2.3 Dart SDK 2.10.4

- Verze: 2.10.4
- Vývojář: Google
- Ke stažení: <https://dart.dev/get-dart>
- Licence: BSD licence

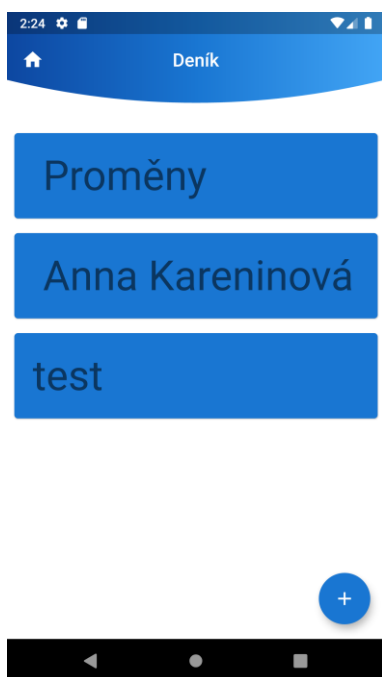
3.3 Ovládání



Obrázek 4 Login



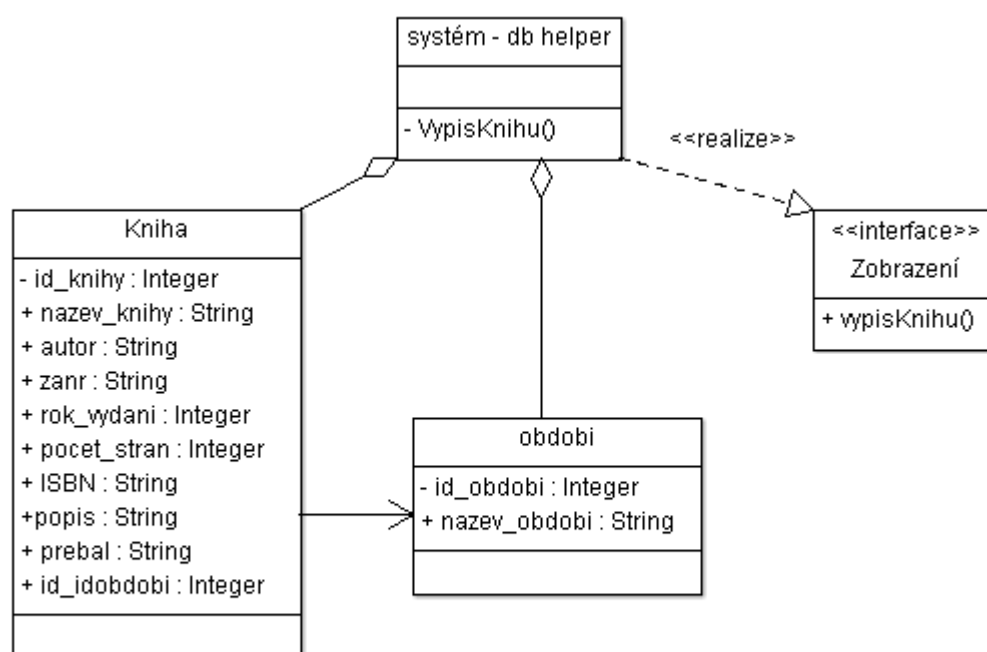
Obrázek 3 Home page



Obrázek 5 Deník

Aplikace se ovládá pomocí výrazných a přehledných tlačítek, navigace aplikací je proto jednoduchá a snadná, v horní liště můžeme najít další funkce, které značí svoji funkci ikonky, které jsou univerzálně používané, a proto by uživateli nemělo dělat problém pochopit jejich význam, například: ikony domečku vrátí na domovskou stránku a ikona ozubeného kolečka otevře nastavení. Původně bylo na navracení použita šipka zpět, ale tohle se uživatelským testováním neosvědčilo jako přehledné a proto se vybrali „ikonické“ ikony.

3.4 Diagram tříd – aplikace



Obrázek 6 Diagram tříd - aplikace

3.5 Důležité třídy

Mobilní aplikace je psaná v programovacím jazyce Dart, ve kterém je vše řízeno třídami, v téhle kapitole si představíme ty nejdůležitější třídy, které řídí celý projekt.

3.5.1 Třída AuthenticationService & model GetUser

Třída AuthenticationService komunikuje již se zmíněnou funkcí Firebase Authentication, zprostředkovává komunikaci mezi touthle funkcí a aplikací, pomocí vlastních funkcí poté registrovat uživatele do systému (funkce signUp) funkce Firebase Authentication a poté dokáže porovnávat otisky a tak umožňuje přihlašování do (funkce signIn) a odhlašování z (funkce signOut) aplikace. Klíček k úspěchu bylo ovšem propojit funkci Firebase Authentication s funkcí Firebase Realtime Database, tohle se stalo největším zádrhelem během vývoje. Bylo potřeba vytvořit model GetUser, který při komunikaci aplikace s funkcí Firebase Authentication uloží uid uživatele, tudíž s uid půjde nadále pracovat v aplikaci, jako právě ukládání dat do Realtime Database. Firebase Authentication má vlastní funkci UserCredential, která vypíše vše dostupné informace, které ukládá... Nám ovšem stačí pouze uid, takže napíšeme průlomovou funkci, která dokáže vytáhnout pouze vysněné uid:

```
//a way of getting user id
GetUser _userFromFirebaseUser(User user) {
  return user != null ? GetUser(uid: user.uid) : null;
}
```

Obrázek 7 GetUser

Poté pomocí funkce UserCredential při vytváření/přihlašování uživatele získáme a uložíme do modelu uid vypsané z Firebase Authentication na kterém stojí celý systém a nadále s ním pracujeme:

```
UserCredential result = await _auth.createUserWithEmailAndPassword(email: email, password: password);
User user = result.user;
```

Obrázek 8 User auth

3.5.2 Třída DatabaseService

Třída DatabaseService komunikuje již se zmíněnou funkcí Firebase Realtime Authentication, zprostředkovává komunikaci mezi touthle funkcí a aplikací. Stará se o zapisování dat do databáze pomocí uid, které mu je předáno z modelu GetUser. Zapisuje: dodatečné informace při registraci uživatele, které služba Firebase Authentication neukládá (funkce addUser), přidává uživatelské

knihy do deníku (funkce addDiary), umožňuje data o knihách v deníku upravit (funkce updateDiary) a také knihu z deníku smazat (funkce deleteDiary).

3.5.3 Třída MyApp

Neboli main, je třída, která se stará o to, aby pouze přihlášení uživatelé získali přístup do aplikace. Používá proto providera, které poslouchá komunikaci mezi aplikací a Firebase Authentication a tak může zjistit, zda je uživatel přihlášen, či nikoliv.

4 WEBOVÁ STRÁNKA

4.1 Spuštění

4.1.1 Spuštění na localhostu (pro vývojáře)

Stáhněte všechny soubory a rozbalte je do vaší požadované složky.

Pro spuštění na localhostu je potřeba jen změnit `base_url` v souboru `application/config/config.php` na 26. řádku. Hodnotu nastavte na svou požadovanou složku.

Je také potřeba mít nainstalovaný Composer a v příkazovém řádku v místě vaší složky spustit příkaz *composer install*.

4.1.2 Spuštění na webu

Pro zpřístupnění projektu na webu jděte na stránku <https://maturitnicetba.herokuapp.com>

4.2 Použité knihovny a frameworky

4.2.1 CodeIgniter 3

- Autor: EllisLab
- Verze: 3.1.11
- Ke stažení: <https://codeigniter.com>
- Licence: MIT License
- Kostra projektu.

4.2.2 Bootstrap

- Autoři: Mark Otto (@mdo), Jacob Thornton (@fat)
- Verze: 4.3.1
- Ke stažení: <https://getbootstrap.com/>
- Licence: MIT License

4.2.3 Popper.JS

- Autor: Jahia Solutions Group SA
- Verze: 1.14.7
- Ke stažení: <https://popper.js.org/>
- Licence: MIT License

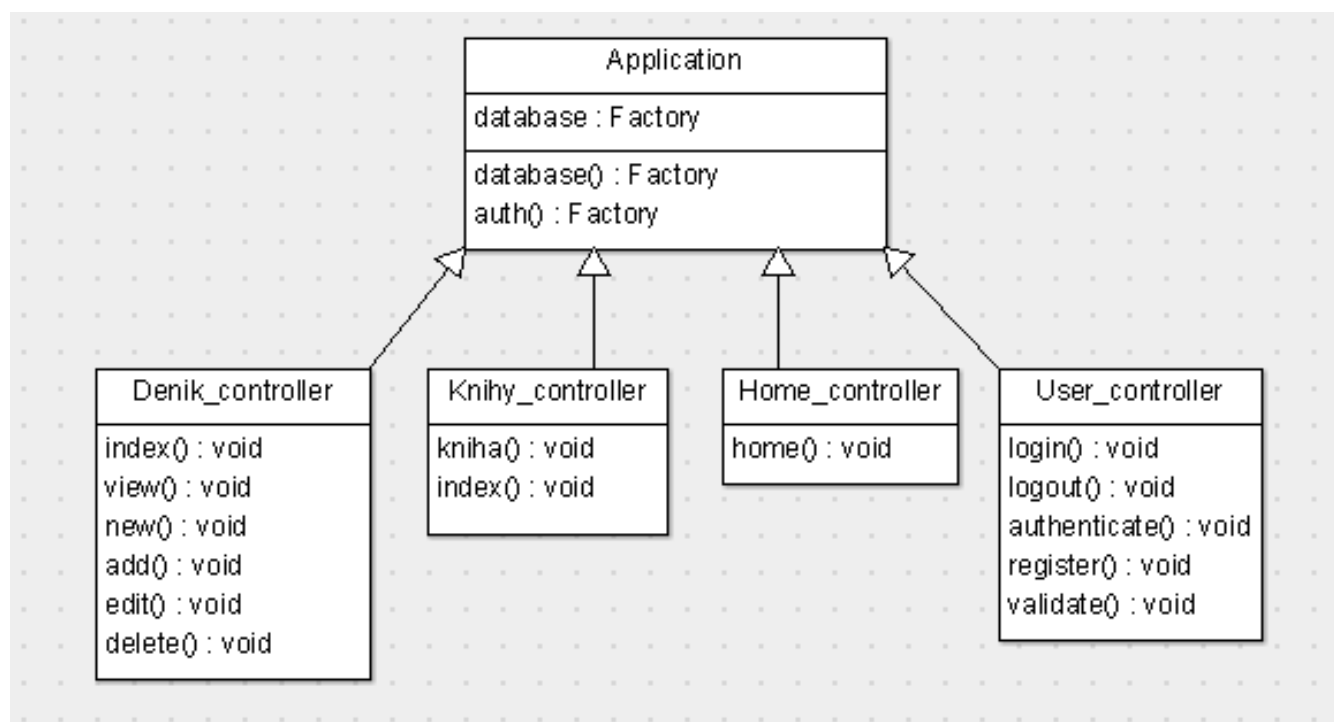
4.2.4 Firebase Admin PHP SDK

- Autor: Jérôme Gamez
- Verze: 5.20.1
- Ke stažení: <https://github.com/kreait/firebase-php>
- Licence: MIT License
- Zprostředkovává připojení k Firebase databázi

4.2.5 Composer

- Autoři: Nils Adermann, Jordi Boggiano
- Verze: 2.0
- Ke stažení: <https://getcomposer.org/>
- Licence: MIT License
- Spravuje nainstalované knihovny v projektu

4.3 Diagram tříd



Obrázek 9 Diagram tříd - web

4.4 Struktura souborů

Codeigniter 3 má pevně danou strukturu souborů. Všechny Controllery musí být ve složce `application/controllers` a views ve složce `application/views`.

Ve složce `application/config` jsou uložena nastavení projektu (např.: `base_url`, `routing`, `session`, `firebase admin key` atd.).

4.5 Controllers

Všechny controllery v aplikaci (`application/controllers`) jsou potomky třídy **Application**, která je uložena v `application/core`. Třída `Application` je potomkem základního controlleru `CI_Controller`, který je v základu knihovny Codeigniter 3.

4.5.1 Application

Jak již bylo zmíněno, třída `Application` je rodičem všech controllerů ve složce `application/controllers`. Slouží především k práci s Firebase.

4.5.1.1 Funkce database

Funkce `database` slouží k připojení k real-time databázi a vrátí instanci třídy `Factory`, která v rámci knihovny `Firebase Admin PHP SDK` slouží k připojení k Firebase. Třída `Factory` poté vykonává další funkce, jako zapisování do databáze a vracení výsledků z databáze.

4.5.1.2 Funkce auth

Funkce `auth` slouží k připojení k službě `Authentication` od Firebase. Pomocí ní je řešeno přihlášení a registrace v controlleru `User_controller`.

4.5.2 Deniky_controller

Třída `Deniky_controller` se stará o funkčnost zapisování, zobrazování a odstraňování zápisků. Nejprve zkontroluje v konstruktoru, zda je uživatel přihlášen a až poté vykonává jednotlivé funkce. Pokud se uživatel nepřihlásil, odešle ho na stránku s přihlášením.

4.5.2.1 Funkce index

Funkce `index` vypíše uživateli seznam jeho zápisků. Na této stránce je může odstranit pomocí tlačítka *Odstranit* v pravém rohu každého zápisku.

4.5.2.2 *Funkce view*

Funkce view slouží k zobrazení jednoho zápisu. Jako parametr dostane klíč zápisu. Podle něj odešle dotaz do databáze, která vrátí příslušný záznam v deníku uživatele.

4.5.2.3 *Funkce new*

Funkce new slouží k přidání nového deníku. Má volitelný parametr `kniha_id`. Pokud jej má, před vyplněním se informace o knize v políčkách view. Po vyplnění formuláře se spustí funkce add.

4.5.2.4 *Funkce add*

Funkce add zapiše záznam do databáze. Po úspěšném zapsání je uživatel odkázán na stránku editu nově vytvořeného deníku. Klíč pro nový záznam se vygeneruje automaticky pomocí knihovny Firebase Admin PHP SDK.

4.5.2.5 *Funkce edit*

Po stisknutí tlačítka *Upravit* deník se spustí tato funkce. Jako parametr se jí předá klíč zápisu. Data v políčkách aktualizuje v databázi a odkáže uživatele na stránku editu.

4.5.2.6 *Funkce delete*

Funkce delete odstraní záznam z databáze. Jako parametr se jí předá klíč zápisu, který použije pro dotaz do databáze. Uživatel je poté odkázán na stránku index deníku a je zobrazen nápis „Záznam byl odstraněn“.

4.5.3 **Home_controller**

Třída `Home_controller` slouží pouze k zobrazení hlavní stránky webu.

4.5.3.1 *Funkce home*

Zkontroluje, zda je uživatel přihlášený a pokud ano zobrazí hlavní stránku webu.

4.5.4 **Knihy_controller**

Třída `Knihy_controller` je zodpovědná za zobrazení stránek s knihami.

4.5.4.1 *Funkce index*

Funkce index slouží k zobrazení všech knih. Seznam je zobrazen jako carousel ale lze jej zobrazit také jako tabulku.

4.5.4.2 *Funkce kniha*

Funkce kniha dostane jako parametr id knihy, podle kterého poté zobrazí podrobné informace o knize.

4.5.5 **User_controller**

Třída User_controller se stará o funkčnost loginu a registrace.

4.5.5.1 *Funkce login*

Funkce login je hlavní stránka, kterou uživatel uvidí hned po spuštění a je vyzván, aby se přihlásil. Po vyplnění a odeslání formuláře je odkázán na funkci *authenticate*.

4.5.5.2 *Funkce authenticate*

Funkce authenticate ověří údaje, které odeslal uživatel s údaji v databázi. Přihlášení je vyřešeno pomocí třídy Factory z knihovny Firebase admin PHP SDK. Pokud je přihlášení úspěšné, uloží se data o uživateli v proměnné \$_SESSION[] a uživatel je odkázán na domovskou stránku. Session je řešena pomocí knihovny Codeigniter 3.

4.5.5.3 *Funkce register*

Funkce register slouží k registraci uživatele. Uživatel nejprve vyplní požadovaná pole a odeslání formuláře je odkázán na funkci *validate*.

4.5.5.4 *Funkce validate*

Funkce validate ověří:

1. Zda je email již zaregistrován
2. Zda uživatelské heslo obsahuje minimálně 6 znaků

Pokud jsou splněny všechny tyto podmínky, uloží se data o uživateli v databázi pomocí knihovny Firebase Admin PHP SDK. Uživatel je poté odkázán na stránku loginu, kde se již může přihlásit do aplikace.

4.5.5.5 *Funkce logout*

Pokud je uživatel přihlášen, může se odhlásit pomocí kliknutí na tlačítko odhlásit se v pravém horním rohu. Po odhlášení se odstraní všechny data o uživateli z proměnné \$_SESSION[] a je odkázán na stránku loginu.

5 ZÁVĚR

Pro shrnutí náš projekt obsahuje webovou a mobilní aplikaci, které vzájemně spolu spolupracují. Všechny funkce, které jsou v mobilní aplikaci najdete také na webu. Po registraci a přihlášení se uživateli zobrazí veškeré funkce aplikace. Naším cílem projektu bylo zpřehlednit knihy, které jsou povinné k maturitní četbě a také deník, do kterého si bude moct uživatel zapisovat různé poznámky, poznatky a jiné věci, které si chce uložit. Každopádně aplikace obsahuje jak knihovnu, ve které si může uživatel zobrazit knihy, které se mu zalíbí a přečíst si o nich nějaké základní informace, tak také právě deník, do kterého si to všechno může poznačovat.

Během programování této aplikace, bylo ze začátku docela obtížné pochopit samotný Flutter, tím že to bylo pro nás všechny něco nového, tak nám dalo dlouhou dobu zabrat, než jsme pochopili jeho logiku. Jde to názorně vidět v porovnání kódu knihovny a deníku. Kdy v knihovně bylo všechno psané nepřehledně a natlačeno na sobě. Postupně, během vytváření Authentication jsme již uměli psát kód hezky a taky přehledně a jde to poznat právě na Authentication, tak na deníku.

Vyskytovalo se zde několik problémů, již od začátku, a to jak v mobilní aplikaci, tak také na webu. Pro začátek bylo těžké pracovat s Firebase databází, v té době nám ještě neznámé, jelikož jsme všichni byli zvyklí na databáze typu SQL. Ovšem Firebase má mnoho zajímavých funkcí, které nám zjednodušili práci například přímo s Authentication. Ale v tu chvíli se objevil další problém, kdy Firebase nemá žádnou oficiální funkční knihovnu, která je kompatibilní s php. Přesněji s CodeIgniterem. Z toho důvodu bylo potřeba vyzkoušet několik různých knihoven přímo od uživatelů, dokud jsme neobjevili plně funkční knihovnu, která umí pracovat právě s Firebase databází.

V celkovém projektu se nám podařilo splnit cíle projektu, a to je dokonce i překonat. Původně nebylo vůbec v plánu, aby se k projektu vytvořil samotný web, ovšem, pak jsme se rozhodli tento web vytvořit z důvodu přístupnosti uživatelům. Podle naší představy finální verze projektu vypadá přesně tak, jak jsme chtěli.

Časem jsme chtěli ještě doplnit například doporučenou četbu, oblíbené knihy, nebo například se zkontaktovat s některou firmou, která prodává e-knihy a nabízet je přímo v naší aplikaci. Také jsme chtěli rozšířit databázi o několik dalších škol. Ovšem z časových důvodů z těchto plánů sešlo, je ale možné se k projektu vrátit a pokračovat v něm v budoucnu, jelikož si myslíme, že to bude užitečné pro hodně studentů, jelikož podobná aplikace zatím není.

6 ZDROJE

1. *Flutter: framework* [online]. [cit. 2021-5-16]. Dostupné z: <https://flutter.dev/>
2. *Android studio: text editor* [online]. [cit. 2021-5-16]. Dostupné z: <https://developer.android.com/studio>
3. *CodeIgniter: framework* [online]. [cit. 2021-5-16]. Dostupné z: <https://codeigniter.com/>
4. *Bootstrap: css* [online]. [cit. 2021-5-16]. Dostupné z: <https://getbootstrap.com/>
5. *PopperJS: JavaScript* [online]. [cit. 2021-5-16]. Dostupné z: <https://popper.js.org/>
6. *Firebase PHP: Firebase Admin SDK for PHP* [online]. [cit. 2021-5-16]. Dostupné z: https://firebase-php.readthedocs.io/en/5.x/?fbclid=IwAR0RFi6uQEnr_YEOSfXdQ-C5phQwFdsawG9-yLbBQaq8B8lDnopXjzmGsT8#
7. *Composer: správa knihoven* [online]. [cit. 2021-5-16]. Dostupné z: <https://getcomposer.org/>