# 

Spojená škola, Komárňanská 28, Nové Zámky

o.z. Stredná priemyselná škola elektrotechnická S.A.

Jedlika – Jedlik Ányos Elektrotechnikai Szakközépiskola

„Mobilná aplikácia pre Študentov SPŠE“

Vlastný projekt

Praktická časť odbornej zložky maturitnej skúšky

Nové Zámky riešiteľ:

# 2021/2022 Adam Hadar

ročník štúdia: **štvrtý**

konzultant: Ing. Mária Poláková

Spojená škola, Komárňanská 28, Nové Zámky

o.z. Stredná priemyselná škola elektrotechnická S.A.

Jedlika – Jedlik Ányos Elektrotechnikai Szakközépiskola

# PRAKTICKÁ ČASŤ ODBORNEJ ZLOŽKY MATURITNEJ SKÚŠKY

Vlastný projekt

|  |  |
| --- | --- |
|  |  |
| Meno študenta: | Adam Hadar |
| Trieda: | IV. AI |
| Školský rok: | 2021/2022 |
| Študijný odbor: | IST |
| Interný konzultant: | Ing. Mária Poláková |
| Externý konzultant: |  |

# Názov projektu: „Mobilná aplikácia pre Študentov SPŠE“

..................................

Žiak

.................................. ..................................

Externý konzultant Interný konzultant

..................................

Zástupca riaditeľky školy

V Nových Zámkoch 15.10.2021

Spojená škola, Komárňanská 28, Nové Zámky

o.z. Stredná priemyselná škola elektrotechnická S.A.

Jedlika – Jedlik Ányos Elektrotechnikai Szakközépiskola

Čiastkové úlohy:

1. Analýza problému, výber najvhodnejšej technológie pre vývoj
2. Výber databázovej technológie a návrh dátového modelu
3. Zabezpečenie a enkryptovanie prihlasovacích údajov v databáze
4. Návrh funkcionality aplikácie a jej dizajnu
5. Výber spôsobov autentifikácie do aplikácie
6. Prepojenie databázy a dát z webu s aplikáciou
7. Programovanie, testovanie a dolaďovanie

# Čestné vyhlásenie

Vyhlasujem, že som túto práce vypracoval samostatne s pomocou konzultanta Ing.

Mária Poláková a uviedol som všetku použitú literatúru.

........................................

# Poďakovanie

Chcel by som sa poďakovať môjmu konzultantovi práce, Ing. Mária Poláková, za vedenie a cenné pripomienky pri záverečnom spracovaní práce.

......................................

# OBSAH

Úvod .................................................................................................................................. x

1. Problematika a prehľad literatúry ..................................................................................... x
2. Ciele práce ........................................................................................................................ x
3. Materiál a metodika .......................................................................................................... x
4. Výsledky práce ................................................................................................................. x
5. Diskusia ............................................................................................................................ x

Zhodnotenie výsledkov práce ....................................................................................... x

Záver ............................................................................................................................. x

Resumé ......................................................................................................................... x

Zoznam použitej literatúry

Zoznam ilustrácií

Prílohy

# ÚVOD

Hlavným cieľom našej mobilnej aplikácii pre študentov SPŠE je uľahčenie, urýchlenie a spríjemnenie štúdia. Tieto ciele sme sa snažili dosiahnuť zoskupením zdrojov ktoré pre svoje informovanie študenti používajú dennodenne. Tieto zdroje pozostávajú z rozvrhu zo školského portálu EduPage. Ďalej dnešného ako aj týždenného obedového menu zo stránky našej školskej jedálne, jedalne.sk. A ako poslednou a najhlavnejšou súčasťou našej aplikácie je zobrazenie mapy hlavnej budovy školy a taktiež budovy „6ky“ a následne aj navigácia medzi jednotlivými učebňami a miestnosťami.

Dôvodov výberu našej práce je viacero. Hlavným faktorom pri výbere práce bola naša osobná skúsenosť. Ako študenti našej školy sme pociťovali, že takáto aplikácia by sa medzi nielen študentmi, a to hlavne medzi prvákmi, ale aj učiteľmi alebo zamestnancami veľmi rýchlo uchytila.

Naša skúsenosť s vývojom mobilných aplikácia pomocou frameworku Flutter začala v januári 2021 štúdiom pomocou online kurzu na stránke Udemy od tvorcu Maximiliána Schwarzmüllera. Našim prvým projektom pomocou Flutteru bola fitness aplikácia TAM fitness. Tento projekt bol vytvorený pre konferenciu IT v Praxi pod záštitou IT akadémie. Na základe našich novo nadobudnutých skúseností sa vývoj mobilnej aplikácie vo Frameworku Flutter stal aj základom vývoja tohto vlastného projektu.

Framework Flutter sa stáva viac a viac populárnym v celosvetovej sfére vývoja mobilných a multiplatformových aplikácii. V nedávnej minulosti bola novou verziou Flutteru 2.0 pridaná možnosť vývoja ako pre Web tak aj vývoj desktopových aplikácii pre všetky platformy. V súčasnosti sa Flutter stáva stále rozšírenejším a mnohé veľké firmy a ich aplikácie prechádzajú na tento Framework. A to napríklad Google Assistant, Google Ads, Stadia, Aplikácia BMW a mnoho ďalších. Tieto skutočnosti nás len utvrdili že vývoj v tomto frameworku poskytne perspektívu do budúcnosti ako nám, tak aj ostatným vývojárom v tomto odvetví.

# PROBLEMATIKA A PREHĽAD LITERATÚRY

* 1. Vývoj mobilných aplikácií

Vývoj mobilných aplikácií je proces vytvárania programových aplikácii určené pre mobilné zariadenia ktoré následne využívajú sieťové pripojenie pre prácu so vzdialenými výpočtovými zdrojmi. Výsledkom vývoja aplikácie je inštalovateľný programový balík.

Okrem samotného programovania a návrhu aplikácie, vývoj spočíva aj z vytvorenia beckendových služieb ako napríklad vlastné API a v poslednom rade testovanie aplikácie na cielených zariadeniach a platformách.

Zameranie vývoju väčšinou pozostáva z dvoch najdôležitejších platforiem moderného trh.. Na trhu mobilných operačných systémov dominujú dve, a to Android od Google a iOS od spoločnosti Apple.

Aj napriek menším podobnostiam pri vývoji medzi týmito dvoma platformami, vývoj pre iOS a Android prebieha za pomoci použitia odlišných vývojových nástrojov (SDK) a odlišných prostredí.

* 1. Alternatívy pre vývoj mobilných aplikácii

Pre vývoj aplikácií existujú štyri najrozšírenejšie prístupy:

* Natívne mobilne aplikácie
* Cross-platformové natívne mobilné aplikácie
* Hybridné web aplikácie
* Progresívne web aplikácie

Každý prístup má svoje výhody ako aj nevýhody. Pri rozhodovaní pre najvhodnejší prístup k jednotlivému projektu, ako hlavne faktory zavažujú používateľskú skúsenosť, výpočtové možnosti a natívne požiadavky od systému vyžadované aplikáciou. Následne to môže byť budget na vývoj aplikácie, časové ciele alebo zdroje potrebné pre prevádzkovanie aplikácie.

* + 1. Natívne aplikácie

Natívne mobilné aplikácie sú programované v programovacom jazyku a frameworku ktorý bol vytvorený a je poskytovaný tvorcom platformy. Takéto aplikácie bežia priamo na operačnom systéme zariadenia.

Výhodami sú : najlepší výkon, priamy prístup k prostriedkom zariadenia

Nevýhodami sú : Náročnejší a drahší vývoj a údržba aplikácie, každá platforma potrebuje vlastný zdrojový kód



O Java a Kotlin

Programovacie jazyky pre vývoj natívnych aplikácii pre platformu Android



O Swift a Objective-C

Programovacie jazyky pre vývoj natívnych aplikácii pre platformu iOS

* + 1. Cross-platformové natívne mobilné aplikácie

Cross-platformové aplikácie môžu byť programovane v rôznych programovacích jazykoch a frameworkoch. Tieto jazyky sú kompilované do natívneho kódu aplikácie ktorá beží priamo na operačnom systéme zariadenia.

Výhodami sú : Pre viacero platforiem postačuje jeden základ kódu, ľahká a nenáročná údržba chodu.

Nevýhodami sú : Závislosť na premosteniach a knižniciach natívnych funkcii, znížený výkon spôsobený pridaným proces premosťovania

* + 1. **Hybridné web aplikácie**

Tieto aplikácie sú navrhované a programované pomocou štandardných webových technológií ako JavaScript, CSS, HTML. Následne je z nich vytvorený inštalačný balíček. Tento balíček sa po nainštalovaní spúšťa pomocou „Webového kontajnera“ ktorý ponúka prostredie pre chod aplikácie rovnaký ako vo webovom prehliadači. Pre prístup k natívnym funkciám a knižniciam sa používa natívne API premostené pomocou Apache Cordova.

Výhodami sú : Zdieľaný kód medzi mobilnou a webovou aplikáciu, využitie znalosti web vývoja pre tvorenie mobilných aplikácii

Nevýhodami sú : Nižší výkon v porovnaní s natívnymi aplikáciami, limitovaná podpora natívnych funkcionalít

* + 1. Progresívne Web Aplikácie (PWA)

PWA ponúkajú odlišný a unikátny prístup v porovnaní s tradičnými vývojovými postupmi. Tieto aplikácie neje potrebne inštalovať, čim sa odstraňuje využívania Obchodov s Aplikáciami. PWA sú webové aplikácie ktoré pre chod využívajú funkcionality webového prehliadača.

Výhodami sú : Rovnaká aplikácia ako pre web tak aj mobilné zariadenia, nie je potrebná inštalácia, sú prístupne pomocou URL

Nevýhodami sú : Limitované schopnosti využívať natívne funkcie zariadenia. Aplikácie sú závislé na prehliadací.

* 1. Framework Flutter

Flutter je vyvíjaný spoločnosťou Google. Prvá verziu Flutteru, pod názvom Sky, bola vydaná v roku 2005. Prvou stabilnou verziou bola verzia 1.0 ktorá bola predstavená 4. Decembra 2018. Nasledujúcou veľkou stabilnou verziou je Flutter 2.0, predstavenou 3. Marca 2021 v ktorej bola po prvý krát pridaná podpora pre web a desktop aplikácie.

Flutter je multiplatformové, open-source, SDK určené na vývoj aplikácií pre mobilné zariadenia(Android, IOS) ako aj pre desktopové(Windows, Linux, Web). Charakteristikou Flutteru je možnosť znovu použitia kódu na všetky platformy so zachovaním jeho funkčnosti s minimálnou nutnosťou zmien. Aplikácie majú vysoký výkon aj na priek tomu že nie sú písané v natívnom kóde(Android – Java, Kotlin; IOS – Swift, Objektíve-C), no vďaka využívaniu priamej komunikácie s danou platformou sa výkon aplikácii takmer vyrovná natívnemu kódu.

Pre vývoj sa používa jazyk Dart. Následne jadro aplikácie Flutter Engine, ktorý je písaný v jazyku C++, zabezpečí kompiláciu, vykresľovanie a prepojenie s grafickým enginom SKIA, zabezpečuje sieťovú komunikáciu a ďalšie súčasti frameworku potrebné pre beh aplikácie.



O - Flutter Architektúra

Flutter sa dá rozdeliť na rôzne vrstvy, od najnižšej to sú:

* Základné triedy, stavebné prvky služieb ako napríklad animácie, knižnica kreslenia, gestá
* **Vrstva vykresľovania** poskytuje prvky pre riešenia rozloženia. Vytvára strom vykresľovateľných objektov. S týmito objektami sa dá manipulovať, a menia sa pre odzrkadlenie zmien
* **Vrstva Widgetov** je vrstva kompozície. Každý vykresľovateľný objekt z vrstvy vykresľovania má pridelenú triedu v strome Widgetov
* **Knižnice Materiáli a Cupertino** ponúkajú sety ovládania ktoré využijú kompozíciu widget vrstvy pre aplikovanie Material(Android) alebo Cupertino(IOS) designu.

# CIELE PRÁCE

Úplne prvým krokom projektu je analýza problému. Výsledkom tejto analýzy je predstava o výsledku a spôsobu vývoju a návrhu. Nasleduje výber najvhodnejšej technológie pre vývoj spomedzi veľkej konkurencie frameworkov určených na vývoj mobilných aplikácii.

1. Výber databázovej technológie
2. Zabezpečenie a enkryptovanie prihlasovacích údajov v databáze
3. Návrh funkcionality aplikácie a jej dizajnu
4. Výber spôsobov autentifikácie do aplikácie
5. Získanie a prekreslenie plánov školy
6. Návrh navigačných bodov a ich prepojenie
7. Prepojenie databázy a dát z webu s aplikáciou
8. Programovanie, testovanie a dolaďovanie

# MATERIÁL A METODIKA

Návrh a vývoj našej aplikácie bol uskutočnený pomocou nasledujúcich programov a  zdrojov :

* Framework Flutter na vývoj aplikácii (V našom prípade mobilných)
* Emulátor zariadenia Android pre potreby testovania
* Editor VSCode pre úpravu zdrojového kódu
* Verziovanie a zálohovanie zdrojového kódu pomocou služieb Git a Git Hub
* Návrh a design aplikácie v softwari Figma
* Návrh plánov školy a navigačných bodov v softwari Figma
  1. Návrh a design aplikácie

Na účely vytvorenia funkčného modulu aplikácie a jej následné designovanie sme využili bezplatný software Figma. Figma je nami overený a v minulosti už využívaný program s ktorým sa pracuje ľahko. Ďalšia výhoda ktorá viedla k jej využívaniu je cloudové ukladanie projektov a teda jednoduché využívanie na viacerých zariadeniach.



O - Návrh Aplikácie

* 1. Vytvorenie repozitára v službe Git Hub a verziovanie pomocou Git

Git slúži na množstvo funkcii pri vytváraní projektu. Počas nášho procesu vyvárania aplikácie sme ho využili na:

* Zálohovanie zdrojového kódu
* Zdieľanie medzi viacerými zariadeniami
* Udržovanie záznamov zmien s prehľadnými popismi
* Napravenie chyb po nesprávnom upravení zdrojového kódu

Git Commit predstavuje súhrn zmien vykonaných od posledného nahraného Commitu. Každý commit obsahuje správu ktorá zmeny popisuje. V našom projekte sme zvolili špecifický formát commitov a to:



O Git Commit

(+) – za týmto znakom nasledované hodnoty sú pridané zmeny

(-) – za týmto znakom nasledované hodnoty sú odstránené zmeny

* 1. Požiadavky pre vývoj Flutter aplikácie

Pred vytvorením prvotného projektu bolo potrebné si stiahnuť a nainštalovať viacero programov.

* Git – Flutter ho vyžaduje k chodu ale keďže ho využívame, už je v systéme dostupný aj pre naše využitie
* Flutter SDK – tento ZIP je potrebné extrahovať do ľubovoľného priečinku, následne stačí tento priečinok pridať do PATH premennej flutter/bin
* Android Studio – môže slúžiť aj ako editor, no pre nás bolo jeho hlavnou súčasťou virtualizovanie Android zariadenia
* Visual Studio – našim preferovaným editorom je VSCode, tento editor zavážil hlavne kvôli našim predošlým skúsenostiam, integrácii množstva flutter funkcií priamo do editoru a jednoduchosť používania
  1. Prvotný projekt a prvé kroky

flutter create kop\_spse

O Príkaz vytvorenie projektu

Pre vytvorenie základného Flutter projektu sme použili príkaz

Tento príkaz nám vytvoril priečinok s názvom kop\_spse v ktorom sú všetky potrebné súbory a priečinky pre začiatok vývoja.

V ďalšom kroku vývoja sme inicializovali git pre náš projekt. Toto sme dosiahli požitím nasledujúcich príkazov v príkazovom riadku bežiacom v našom priečinku.

Pred začatím tohto procesu sme si vytvorili „Git repository“, teda projekt alebo repozitár v službe GitHub ktorý bude slúžiť ako miesto kde sa náš zdrojový kód bude ukladať.

git init

git add .

git remote add origin https://github.com/Moreplavec64/kop\_spse  
git push -u origin master

O Inicializovanie Git v projekte

Našou hlavnou vetvou je master, ktorú sme museli spacifikovať ako hlavnú v príkaze git push, týmto spôsobom sa všetky naše commity budú ukladať na vetvu master.

* 1. Štruktúra projektu



O Štruktúra nového projektu

Po vytvorení projektu je vytvorených minimálne množstvo súborov, preto sme do nášho projektu pridali naše priečinky pre organizáciu a prehľadnosť súborov.

* + 1. Priečinok assets

V tomto priečinku sú uložené všetky súbory, ktoré naša aplikácia využíva pri jej chode. Tieto súbory môžu byt obrázky, konfiguračné súbory, súbory so statickými dátami(JSON).

V našom assets priečinku sa nachádzajú tri podpriečinky:

* fonts – obsahuje údaje o nami používanom štýle písma.
* images – do tohto priečinku sme počas vývoja ukladali všetky obrázky ktoré naša aplikácia využíva
* test\_respose – obsahuje uloženú odpoveď EduPage pre testovanie a vývoj ak realtime dáta z EduPage nie sú k dispozícii



O Priečinok Assets



O Priečinok lib

* + 1. Priečinok lib

Priečinok lib obsahuje všetok zdrojový kód aplikácie. Tento priečinok sme si rozdelili na pod priečinky podľa využitia dát v nich.

* models – do tohto priečinku sme ukladali súbory s triedami, ktoré slúžia iba ako modely, t.j. napríklad User model alebo model pre vyučovaciu hodinu
* providers – do priečinku providers sme ukladali súbory s triedami providerov, ktoré riešia state management v našom projekte
* screens – v priečinku screens sa nachádzajú súbory s triedami widgetov jednotlivých obrazoviek
* utils – tento priečinok obsahuje súbory so všetkými statickými metódami a konštantami pre náš projekt ktoré sa dali oddeliť od dynamickej časti kódu
* widgets – v tomto rozsiahlom priečinku sa nachádza viacero podpriečinkov, podpriečinky zlučujú widgety ktoré patria iba k daným obrazovkám. Mimo týchto podpriečinkov sú widgety ktoré patria viacerým obrazovkám naraz.
* main.dart – main.dart je jeden zo súborov vygenerovaným automaticky pri vytvorení projektu. Tento súbor obsahuje metódu main ktorá je vstupným bodom našej aplikácie a ktorej súčasťou je spúšťanie samotnej aplikácie.
  + 1. Pubspec.yaml

Súbor pubspec.yaml zohráva pri vývoji Flutter aplikácie dôležitú rolu. Do tohto súboru sme zapisovali všetky súbory a balíčky na ktorých je náš projekt závislí. Formát YAML v ktorom je tento súbor písaný je ľahko čitateľný konfiguračný súbor.

Formát tohto súboru nám prinášal často problémy, vzhľadom na to že jedna medzere na nesprávnom mieste spôsobila že projekt vôbec nefungoval. Aj napriek týmto problémom je tento súbor prehľadný a vďaka radám IDE pri chybách aj ľahko editovateľný.

Do súboru pubspec sme zapisovali všetky balíčky ktoré sme potrebovali importovať do nášho projektu. Tieto balíčky sa zapisujú každý do jedného riadku s korešpondujúcou verziou balíčku.

Súbor pubspec.yaml taktiež pre správne fungovanie aplikácie a musí obsahovať všetky obrázky a iné súbory ktoré nie sú zdrojovým kódom aplikácie. Ďalej sme do pubspecu museli zapísať náš font Sen ktorý v našej aplikácii používame.



O pubspec balíčky



O pubspec assets a fonts

* 1. Firebase autentifikácia a cloudové úložisko

Integrácia služieb Firebase je jednoducho uskutočniteľná vďaka balíčkom. V našom projekte využívame služby Firebase Authentification a Firebase Cloud Firestore.

* + 1. Firebase Cloud Firestore

Ako cloudové úložisko pre našu aplikáciu sme zvolili Firebase Cloud Firestore. Táto bezplatná služba je pre našu aplikáciu vhodnou možnosťou vďaka ľahkej implementácii, bezplatnosti a následne podpore a udržiavaniu Flutter balíčkov spoločnosťou Google.

* + 1. Firebase Authentification

Prihlasovanie bolo prvou funkcionalitou pridanou do našej aplikácie. Tento proces sme implementovali pomocou Firebase Auth. Prihlasovanie sme umožnili cez účet Google ako aj cez email a heslo. Po registrácii za vo Firebase Authentification konzole vytvorí vstup s unikátne vygenerovaným UID ktoré je pridelené k emailu používateľa.

* 1. Prihlasovanie a registrácia

Po zapnutí aplikácie je používateľ prezentovaný s prihlasovacou obrazovkou. Na tejto obrazovke sa pri prihlasovaní nachádzajú dva widgety textového vstupu TextInputWidget. Pod vstupmi sa nachádzajú tlačidlá, tlačidlo na prihlásenie emailom a heslom a tlačidlo na prihlásenie pomocou účtu Google.

Kontaktovanie Firebase Authentification služby nasleduje po úspešnom overení vstupných údajov. Autentifikačné funkcie využívajú koncept asynchrónneho programovania, vďaka ktorému na našej obrazovke zobrazujeme načítavanie ako aj chybové hlášky ktoré počas autentifikácie môžu nastať. Na samotné kontaktovanie využívame balíček firebase\_auth. Následne z tohto balíčka využijeme metódy.

final FirebaseAuth \_auth = FirebaseAuth.instance;

await \_auth.createUserWithEmailAndPassword(...);

await \_auth.signInWithEmailAndPassword(...);  
await \_auth.signInWithCredential(*OAuthCredential* credential);

O Metódy Firebase Auth

Flutter ponúka na riešenie problematiky vypĺňania vstupov widget Form. Form spája všetky vstupy do jedného celku a uľahčuje kontrolu a odovzdávanie. Form widget musí obsahovať FormKey. Cez tento kľuč sa dá formulár odovzdať. Odovzdaním sa zavolajú validačné metódy ktoré sme implementovali pre každý vstup. Tieto metódy zabezpečia správnosť údajov a zobrazia chybové hlásenie ak zadané údaje nespĺňajú podmienky metód.

final GlobalKey<FormState> \_formKey = GlobalKey<FormState>();

Form(

key: \_formKey,

child: Column(  
 children: [  
 TextInputWidget(

controller: \_emailController,

validator: (String? x) {...}),

...

LoginButton((){

if(\_formKey.currentState!.validate()){

...

}),  
 ]))

O Form widget príklad

Registrácia sa začína rovnakým formulárom, no následne sa zobrazí druhá obrazovka prihlasovanie. Na tejto obrazovke je nový formulár v ktorom sú dva vstupy pre prihlasovacie údaje do EduPage. Pod nimi je preferovaný jazyk a tlačidlo registrovať.

* 1. Ukladanie a fetchovanie dát z databázy

Začali sme navrhnutím dátového modelu pre našu databázu. Dátový model vo Firebase Firestore pozostáva z Collections. V našom prípade sme vytvorili iba jeden collection, a to USERS. Collection USERS obsahuje dokument pre každého používateľa, pomenovaný unikátnym kľúčom priradeným k používateľovi. Každý dokument obsahuje 4 vstupy. Tieto údaje vytvoríme pri registrácii a pri ich zmene sa dáta v databáze iba aktualizujú.



O Model cloudového úložiska

* 1. Získavanie dát z EduPage

Dáta z EduPage získavame sériou HTTP requestov. Tieto requesty na seba nadväzujú, preto sme museli naprogramovať do našej aplikácie funkcionalitu uchovávania cookies a spracovávania HTTP odpovedí.

Prvý request slúži na vytvorenie relácie(session) a nastavenie cookies pre ďalšie requesty. Tieto cookies získavame zo set-cookie hlavičky odpovede v ktorej sa nachádza list potrebných cookies. Z prvého requestu získame PHPSESSID cookie. Z tela odpovede je potrebné získať aj csrfauth token pre použitie v nasledujúcom requeste kde prebehne prihlásenie. Na toto získavanie sme využili regulárne výrazy.

final RegExp tokenRegex = RegExp(

r'(?<=name="csrfauth" value=")(.\*)(?=">)',

caseSensitive: true,

multiLine: false,

);

final String? csrfToken = tokenRegex.stringMatch(r.body);

O Regular Expression pre CSRF token

Do nasledujúceho requestu pridávame telo s parametrami, meno, heslo a csrfauth token. Ďalej do hlavičky requestu pridáme cookies. Tento request nám vracia stavový kód 302, čo pre nás znamená že musíme v kóde nasledovať presmerovanie. URL na ktorú nás response presmerováva hľadáme v hlavičke location.

Po nastavení ďalších cookies nám nasledujúci GET request vracia všetky potrebné dáta vo formáte JSON. Tie sú v nečitateľnom formáte, preto naša aplikácia tieto dáta parsuje a ukladá pre nás potrebne dáta do nami navrhnutých modelov. Tieto dáta pre aplikáciu budú prístupné cez EduPage provider, v ktorom ich ukladáme ako List LessonPlan objektov.



O LessonPlan model diagram

* 1. Zobrazenie a získavanie dát jedálne

Dáta jedálne získavame z webovej stránky menu a následne vybraním potrebných dát z html kódu stránky.

final document = html.parse(response.body);  
List<h.Element> menu = document.getElementsByClassName('menu-day-innertable');

O HTML parse kód

Následne spracované dáta ukladáme do Mapy v ktorej ku kľúču typu DateTime zodpovedá List String hodnôt. Tieto hodnoty sú menu na dnešný deň. Widget ktorý zobrazuje iba dnešné menu na domovskej obrazovke pristúpi k tejto mape a zobrazí menu ktorého kľuč je dnešným dátumom.

* 1. Vytvorenie plánov školy pre navigáciu

Začiatkom implementovania navigácie a zobrazovania máp bolo ich vytvorenie. Tieto mapy sme vytvorili na základe plánov naskenovaných z papiera, takže presnosť nebola vždy vhodná. Vďaka nepresnostiam bolo vytváranie plánov zdĺhavý proces. Plány sme vytvárali v softwari Figma. Ku každému poschodiu prislúchajú 3 plány:

* Mapa bez názvov učební
* Mapa s názvami učební
* Mapa s waypointami



O Mapa prízemia s waypointami

* 1. Navigácia a zobrazovanie máp

Navigácia v našej aplikácii je založená na využití Dijkstrovho algoritmu. Tento algoritmus slúži na nájdenie najkratšej trasy medzi dvoma bodmi ohodnoteného grafu prepojenia učební. Pre fungovanie takejto navigácie sme daný graf museli navrhnúť a vyhotoviť, následne aj prepísať do vhodného formátu ktorý dokážeme využiť v nasej aplikácii.

Po zmapovaní a vytvorení všetkých waypointov, ktorých počet dosiahol 208, sme tieto waypointy najprv prepojili. Takúto mapu sme aj graficky znázornili v softwari Figma. Takto vytvorené vrcholy a hrany sme prepísali do kódu vo forme Mapy, ktorej kľúčom je názov učebne a hodnotou je Mapa, v ktorej sa nachádzajú ako kľúče názvy učebni ktoré susedia a hodnotou je vzdialenosť.

'D106':{

'D1051': 13,

'D107/D108': 16

},

'D107/D108': {'D106': 16},

O Zápis ohodnoteného grafu

Po vyhľadaní trasy sme ju potrebovali takisto aj zobraziť v aplikácii. Pre tieto účely sme museli napísať súradnice každého z waypointov do kódu, aby bolo vykresľovanie trasy možné. Takisto sa pri vykreslení trasy vyznačí počiatočná a cieľová učebná. Pre túto funkcionalitu sme do kódu zapísali aj súradnice všetkých učební.

Map<String, Map<String, Rect>> suradniceUcebni = {  
 'HBP1': {  
 'A109': Rect.fromLTWH(142, 240, 36, 16),  
 ...,  
 }  
};

O Súradnice Učební

 Map<String, Map<String, Offset>> suradniceWaypointov = {

'HBP0': {  
 'B0VYCHOD': Offset(177, 98.5),  
 ...,  
 },  
};

O Súradnice waypointov

* 1. Nastavenia a lokálne ukladanie dát

if (loginResponse.headers['location']!.contains('bad')) {

setLoginStatus = LoginStatus.LoginFailed;

throw Exception('Nespravne udaje!');

}  
final loggedInResponse = await http.post(

Uri.parse('https://$\_school.edupage.org/user/'),

headers: {'Cookie': \_cookieList},

);

\_parseEduJsonData(data: loggedInResponse.body);

updateAktualne(forceCheckHodiny: true);

setLoginStatus = LoginStatus.LoggedIn;

# VÝSLEDKY PRÁCE

Kapitola Výsledky práce spolu s kapitolou Diskusia sú najvýznamnejšou časťou a ťažiskom celej práce. V tejto kapitole sa nachádzajú len vlastné výsledky, zistenia a pozorovania. Výsledky majú byť logicky, prehľadne a zrozumiteľne usporiadané a pri popisovaní dostatočne zhodnotené. Zároveň autor komentuje všetky zistenia, skutočnosti a poznatky, ktoré autor získal a konfrontuje ich s výsledkami iných autorov. Táto kapitola sa tiež začína na novej strane, je možné ju spojiť s kapitolou Diskusia do jednej kapitoly

Výsledky a diskusia. Výsledky meraní, dotazníkov, testov a pokusov je vhodné spracovať aj do tabuliek a grafov (kvôli prehľadnosti). Pozorovanie je vhodné doplniť najdôležitejšími a najvýznamnejšími nákresmi, mapami, fotografiami.

Rozsiahlejšie tabuľky a grafy sa obyčajne umiestňujú do príloh, pričom v texte sa musia nachádzať odkazy na ne. Na tie najdôležitejšie výsledky musí byť čitateľ v texte upozornený.

# DISKUSIA

V tejto časti sa nachádzajú úvahy a porovnania vlastných výsledkov s výsledkami, ktoré dosiahli v danej oblasti iní autori. V tejto časti sa interpretujú najdôležitejšie a najvýznamnejšie zistenia a výsledky, hlavne tie, ktoré majú veľký význam vo vzťahu k riešenému problému. Diskusia musí dávať odpovede na otázky a ciele vytýčené v úvode práce. V tejto časti autor vyjadruje svoje názory a postrehy ku skúmanej problematike. Výsledky porovnáva s literatúrou a vyvodzuje z nich vlastné závery – dedukcie. Medzi ne patrí aj konkrétne vlastné riešenie, alebo vlastný návrh na vyriešenie problému, ktorý práca sleduje. Tieto časti treba osobitne vyzdvihnúť, napísať, ako by sa dali vlastné výsledky, zistenia, návrhy či poznatky autora uplatniť v praxi.

# ZHODNOTENIE VÝSLEDKOV PRÁCE

V tejto časti stručne ale jasne a presne autor popíše cieľ práce, metodiku a urobí súhrn najdôležitejších zistení, výsledkov svojej práce. Odporúčaný rozsah je 10 –15 riadkov. Je to vlastne komentovaný obsah práce. Zhrnutie je veľmi dôležitou časťou práce , pretože čitateľ po prečítaní bude vedieť, o čom práca je a čo autor zistil.

# ZÁVERY PRÁCE

V závere autor stručne zhodnocuje dosiahnuté výsledky a splnenie vytýčených cieľov. Zdôrazňuje odlišné fakty, ich objektivitu, význam a možnosti využitia v praxi. Nemá obsahovať rozbory a štúdie, ktoré patria do diskusie. V závere prezentuje autor svoj názor na daný problém a jeho riešenie. Musí vyzdvihovať prínos návrhov autora práce na daný problém a poukázať na spôsob ich realizácie. Záver by mal načrtnúť ďalšiu perspektívu práce v danej problematike so získanými poznatkami. Odporúčaný rozsah je jeden až jeden a pol strany**.**

# RESUME

Je vlastne stručné zhrnutie obsahu práce a jej hlavných myšlienok. Píše sa v anglickom jazyku. Autor v ňom popíše cieľ práce, metodiku a urobí súhrn zistení a výsledkov vlastnej práce. Odporúčaný rozsah je 10 – 15 riadkov.

# ZOZNAM POUŽITEJ LITERATÚRY

Zoznam použitej literatúry obsahuje úplný zoznam bibliografických odkazov. Rozsah tejto časti je daný počtom použitých literárnych zdrojov, ktoré musia korešpondovať s citáciami v texte. Pomocou Zoznamu použitej literatúry sa má čitateľ práce dostať k pôvodným prameňom, ktoré boli citované v práci (a nie sa dozvedieť o autorovom teoretickom rozhľade). V Zozname použitej literatúry sa teda uvádza iba literatúra citovaná v texte. Zoznam musí byť v abecednom poradí. Obsahuje bibliografické odkazy, t. j. informácie o dokumentoch, ktoré sa skutočne použili pri písaní práce. Musia byť v ňom uvedené odkazy na pramene, uvedené v texte práce (aj pramene pod obrázkami a tabuľkami).

Techniku citovania a uvádzania bibliografických odkazov predpisujú rôzne národné i medzinárodné normy. Pre citovanie literárnych prameňov ako aj tvorbu bibliografických odkazov sa na Slovensku využíva norma STN ISO 690 (1998) a STN ISO 690-2 (2001).

„VZOR“

1. Dzianová, O. 2009. Odporúčania pre publikovanie štatistických informácií v rámci

*Národného štatistického systému*. Štatistický úrad Slovenskej republiky. Bratislava: č. 410-0001/2010

1. Kuláková, M. 2010. *Úprava tabuliek*. Slovenský stenograf. Pezinok: vydavateľstvo PARENTES, 2010
2. Kuláková, M. 2011. Didaktika administratívy a korešpondencie. Bratislava:

Vydavateľstvo EKONÓM, 2011

1. Pecinovský, J. – Pecinovksý, R. *Microsoft Word 97 snadno a rychle*. Praha: Grada Publishing, 1997
2. Sehnalová, J. – Ďurišová, H. 2008. Administratíva a korešpondencia pre 2. ročník OA.

Bratislava: EXPOL PEDAGOGIKA, 2008

1. Sehnalová, J. 2005. *Administratíva a korešpondencia*. Bratislava: EXPOL PEDAGOGIKA, 2005
2. Skalka, J. – Jakab, I. 2005. Základy PC, Windows XP, Office 2003 podrobný sprievodca pre začiatočníkov a stredne pokročilých. Nitra: Vydavateľstvo AM – SKALKA, 2005
3. Skalka, J. – Klimeš, C. – Lovászová, G. – Švec, P. *Informatika na maturity a prijímacie skúšky.* Nitra: Vydavateľstvo ENIGMA, 2007

13/28

# PRÍLOHY

Prílohy práce sú nepovinnou časťou práce. Ak sa autor pri písaní práce rozhodne, môže všetky netextové časti (tabuľky, grafy, mapy, fotografie, CD, ...) umiestniť do Prílohy. Ak je príloh viac, kvôli prehľadnosti sa odporúča vypracovať Zoznam príloh. Ten je podľa potreby rozčlenený na jednotlivé časti. Poradie nie je fixné, väčšinou sa však tabuľky a grafy umiestňujú do prednej časti a fotodokumentácia na koniec príloh. V písomnej forme práce podľa jej charakteru môžu byť Prílohy zviazané s textovou časťou, alebo dôsledne uložené vo zvláštnom obale tak, aby pri manipulácii s nimi nedošlo k ich poškodeniu alebo k strate.

Pri elektronickom spracovaní práce je potrebné Prílohy uložiť do formátu pdf. alebo ZIP (max.

32 MB) a potom vložiť do on – line systému do časti nahrávanie súborov samostatne.

Prílohy sú uvedené na Zozname príloh a očíslované. Prílohami môžu byť:

nákresy (ilustrácie), tabuľky, grafy a diagramy, mapy, fotodokumentácia a iný dokumentačný materiál, funkčné modely, technické zariadenia a iné súčasti práce