



Jihočeská univerzita v Českých Budějovicích

Přírodovědecká fakulta

Mobilní aplikace STAG

Bakalářská práce

Tereza Králová

Vedoucí práce: PhDr. Miloš Prokýšek, Ph.D.

České Budějovice 2019

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

ZADÁVACÍ PROTOKOL BAKALÁŘSKÉ PRÁCE

Student: Tereza Králová
(jméno, příjmení, tituly)

Obor – zaměření studia: Aplikovaná informatika

Katedra: Ústav aplikované informatiky

Školitel: PhDr. Miloš Prokýšck, Ph. D.
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Garant z PFF:
(jméno, příjmení, tituly, katedra – jen v případě externího školitele)

Školitel – specialista, konzultant:
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Téma bakalářské práce: Mobilní aplikace STAG

Cíle práce:

Cílem práce je vytvořit mobilní aplikaci, která prostřednictvím API studijní agendy STAG poskytne náhled na data pro studenty. Aplikace umožní například získat přehled o absolvování povinných / nepovinných předmětů, zobrazit rozvrh apod. Řešitel provede rešerši stávajících aplikací obdobného typu a stanoví vhodnou funkcionalitu. Zároveň bude součástí práce popis relevantních částí API nutných pro běh aplikace.

Součástí práce budou bezpodmínečně následující dokumenty:

- Zadávací dokumentace
- Analýza funkčních a nefunkčních požadavků
- Dokumentace kódu
- Testovací dokumentace
- Uživatelská dokumentace

Základní doporučená literatura:

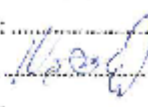
Financování práce :

Vedoucí práce : PhDr. Miloš Prokýšck, Ph. D.podpis : 

U externích vedoucích fakultní garant práce.....podpis :

Garant oboru bak. studia (nepožaduje se u zaměření „příprava na mag. studium biologie)

..... podpis :

Vedoucí laboratoře: PhDr. Milan Novák, Ph. D. podpis 

Přítlačný souhlas vedoucího ústavu AVpodpis :

V Českých Budějovicích dne

Převzal/a dne..... 23.7.2019 podpis : Miloslav T.

Králová, T., 2019: Mobilní aplikace STAG. [Mobile application STAG. Bc. Thesis, in Czech] – 42 p., Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Bakalářská práce se zabývá zkoumáním a následným vývojem mobilní aplikace nad IS/STAG pro Jihočeskou univerzitu v Českých Budějovicích. Aplikace poskytuje studentovi přehledné zobrazení informací obsažených v IS/STAG. Součástí je také možnost plánování předmětů na budoucí semestry.

V práci dochází k rešerši existujících programů na podobném principu. Na základě tohoto zkoumání je navržen dotazník, jehož vyhodnocení je následně využito jako podklad pro tvorbu programu.

Praktická část práce se stává z vytvoření programu za pomoci frameworků, které byly v práci detailně popsány. Těmito frameworky jsou: Ionic, Angular, Apache Cordova. Výsledný program funguje jako front-end zobrazení informací z API IS/STAG a vlastní API pro uchovávání dat plánovače.

Klíčová slova

IS/STAG, mobilní aplikace, Ionic framework, Angular, Apache Cordova

Anotation

This bachelor's thesis focuses on research and subsequent development of mobile application build upon IS/STAG for University of South Bohemia in České Budějovice. Application will provide students with organized view of information in IS/STAG. Included with possibilities to plan subjects for oncoming semesters.

The work contains research of existing programs on a similar principle. Based on this research survey is created. And its evaluation is used as basis for program development.

Practical part focuses on creating software using frameworks, which have been described in detail. Aforementioned frameworks are: Ionic, Angular, Apache Cordova. Resulting application works as front-end view of information from API IS/STAG and own API for planner data retention.

Keywords

IS/STAG, mobile application, Ionic framework, Angular, Apache Cordova

Prohlášení:

Prohlašuji, že svoji bakalářskou práci jsem vypracovala samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích, dne 22. května 2020

Tereza Králová

Poděkování:

Děkuji vedoucímu bakalářské práce, PhDr. Miloš Prokýšek, Ph.D. za cenná doporučení a trpělivost při konzultacích.

Obsah

1. Úvod.....	1
1.1 Cíl práce	2
2. Analýza funkčních požadavků	3
2.1 Rešerše	3
2.2 Dotazník	5
2.3 Výsledky dotazníku.....	6
3. IS/STAG.....	12
3.1 Webové služby IS/STAG.....	15
3.2 Využité webové služby nad IS/STAG	16
3.2.1 Veřejné služby (bez nutnosti přihlášení)	18
3.2.2 Služby pro uživatele (s nutností přihlášení)	19
4. Ionic Framework	21
4.1 Angular.....	23
4.2 Apache Cordova.....	24
5. Popis praktické části.....	25
5.1 Vlastní komponenty	27
5.1.1 Menu.....	28
5.1.2 Rozvrh	28
5.2 Domovská stránka	30
5.3 Moje známky.....	30
5.4 Info uživatele.....	31
5.5 Zkoušky.....	31
5.6 Vyhledávání	32
5.6.1 Detail	33

5.7	Plánovač	33
5.7.1	Databáze	35
5.7.2	Zobrazení naplánovaného rozvrhu	36
5.8	Generování aplikace	39
6.	Diskuze.....	41
7.	Závěr	42
8.	Seznam použité literatury.....	43
9.	Seznam obrázků	47
10.	Seznam grafů.....	48
11.	Seznam tabulek	49
12.	Přílohy	50

1. Úvod

Tato bakalářská práce se zabývá vytvořením mobilní aplikace pro zobrazení informací z IS/STAG¹ na mobilních zařízeních. Některé univerzity, vysoké školy a vyšší odborné školy tento informační systém studijní agendy, zkráceně IS/STAG, používají pro administraci. Mezi základní informace poskytované internetovým portálem IS/STAG patří studentův rozvrh a přehled o jeho známkách, samotný IS/STAG je pak určený pro univerzity, vysoké školy a vyšší odborné školy (1).

IS/STAG ovšem nenahrazuje funkci internetových stránek výše uvedených institucí. Většina služeb, které IS/STAG poskytuje vyžadují přihlášení, avšak některé je možné použít pro prohlížení informací bez nutnosti přihlášení. Mezi tyto služby patří vyhledávání učitelů, předmětů, zobrazení kalendáře pro akademický rok, aj. Na základě přihlášení do IS/STAG jsou pak dostupné další funkce portálu odpovídající roli přihlášeného uživatele. Například pokud přihlášený uživatel bude mít roli student, může si zobrazit, které předměty absolvoval a také získá možnost vyhledávat informace o jiných studentech.

IS/STAG má také dostupnou mobilní verzi webových stránek, avšak ta obsahuje velké množství informací. Tato webová stránka má podobný vzhled jako stránka, která je využívána pro desktopové zobrazení. Z tohoto důvodu se spousta uživatelů v mobilní verzi webových stránek velice špatně orientuje. Právě proto bylo vyvinuto několik mobilních aplikací, které se tímto způsobem pokoušejí vyhýbat zobrazení webových stránek IS/STAG na mobilních zařízeních. Tyto mobilní aplikace jsou z větší části vyvíjeny pro určitou vysokou školu, pouze několik z nich podporuje zobrazení IS/STAG pro více vysokých škol (2). Takto upravené zobrazení IS/STAG je již více vyhovující, avšak většina z nich je na úkor přehlednosti nedostatečně vybavena

¹ Blíže viz kapitola 3 této práce

menším množstvím funkcí. Na základě těchto nedostatků je v rámci této práce vyvíjena mobilní aplikace pro správu informací z IS/STAG.

1.1 Cíl práce

Cílem práce je vytvořit mobilní aplikaci, která poskytne studentům Jihočeské univerzity náhled na data z IS/STAG. Konečná funkcionalita aplikace bude zvolena na základě preferencí studentů z Jihočeské univerzity, které budou zjištěny pomocí dotazníku sestaveného v rámci této práce. Příkladem funkcionality je zobrazení rozvrhu, předmětů a známek přihlášeného studenta, či možnost zápisu na zkouškové termíny.

Výsledná aplikace bude vyvíjena za pomoci frameworku pro hybridní aplikace Ionic. Zobrazovaná data budou získávána prostřednictvím API IS/STAG. Další součástí práce bude rešerše stávajících aplikací obdobného typu, dokumentace kódu, testovací a uživatelská dokumentace. Také budou popsány relevantní části API IS/STAG, které jsou nutné pro běh aplikace.

2. Analýza funkčních požadavků

2.1 Rešerše

Zobrazení informací poskytovaných IS/STAG na mobilních zařízeních pomocí webového prohlížeče je z pohledu většiny studentů velice nepřehledné. Právě z tohoto důvodu bylo vytvořeno několik mobilních aplikací, které mají tento problém řešit. Tyto aplikace jsou většinou mířené na určitou univerzitu, avšak některé z nich lze použít na všech školách používajících IS/STAG.

Mobilní aplikace UPlikace je určena pouze pro studenty a vyučující Univerzity Palackého v Olomouci. Nabízí přehled o studiu, interaktivní mapu s vyznačenými budovami univerzity a zobrazuje také probíhající a následující předměty. Obsahuje také rozvrh, který je ale řešen jiným způsobem než na stránkách IS/STAG. Aplikace jej zobrazuje jako předměty řazené pod sebou podle času výuky, IS/STAG naopak ve výchozím režimu poskytuje klasické tabulkové zobrazení rozvrhu, které lze přepnout na jednotlivé dny v týdnu či použít výpis podle data. Aplikace umožňuje zobrazení zkouškových termínů a také má funkce pro zapsání a odepsání studenta na termíny. Poskytuje různá upozornění, mezi které patří: zadání nové známky, zapsání posudku kvalifikační práce, nový zkouškový termín, uvolnění termínu a na konec možnosti zapsání či odepsání termínu. Student si také může zobrazit kvalifikační práce a aktuality z akademických zdrojů.

IS STAG – Rozvrh tvé univerzity je aplikace dostupná pro všechny školy, které využívají služeb IS/STAG. Aplikace byla vyvinuta v rámci bakalářské práce na Univerzitě Jana Evangelisty Purkyně v Ústí nad Labem. Grafické zobrazení rozvrhu je velice podobné s aplikací UPlikace, avšak na rozdíl od ní poskytuje rozvrh i bez připojení k internetu. Mezi další funkce poskytované aplikací je také seznam zkouškových termínů, které lze také zapsat či odepsat. Aplikace také poskytuje elektronický index tedy informace včetně známky o předmětech, které student studoval

či studuje. Oproti první rozebírané aplikaci je tedy ochuzena o některé funkce, poskytuje ovšem základní funkcionalitu jako je zobrazení známek či rozvrhu.

Univerzita Hradec Králové má také svojí aplikaci na IS/STAG – Moje UHK, jenž je pouze přizpůsobenou verzí UPlikace a tak poskytuje stejné funkce, jako byly již výše uvedeny. Další variantou této aplikace je také StuduJU, která je upravena pro studenty a vyučující Jihočeské univerzity v Českých Budějovicích. Na stránkách IS/STAG je také odkaz na aplikaci Univerzity Tomáše Bati ve Zlíně – Studuj na UTB, avšak po přechodu na stránku Obchod Google Play nebyla tato aplikace nalezena.

Na Obchod Google Play lze také nalézt aplikaci IS/STAG Mobile Beta, která jak již název napovídá je teprve ve vývoji. Tato aplikace je psána samotnými vývojáři IS/STAG. Ta bude poskytovat studentům zobrazení rozvrhu a informace o studovaných předmětech. Také poskytne správu zkuškových termínů a zobrazení notifikací. Pro studenty a vyučující Západočeské Univerzity v Plzni také poskytne jídelníček. Vyučujícím pak umožní také zobrazení rozvrhu a notifikací, ale také rozesílání hromadných emailů.

Aplikace Univerzita – mobilní klient IS STAG je také v předběžném přístupu na Obchod Google Play. Poskytne přístup studentům všech škol, které používají IS/STAG. Aplikace bude obsahovat přehled studijních výsledků, rozvrh studenta a jeho přátel ve stejných formátech jako poskytuje portál IS/STAG. Student si bude moci nastavit notifikace v rámci rozvrhu a aplikace také poskytne studentovi možnost vybrat si, které rozvrhové akce se mají v rozvrhu zobrazovat. Poskytne také funkci vyhledávání v rámci univerzity a možnost přidat si přátele. Bude obsahovat také harmonogram školy s jídelníčkem a podobně jako UPlikace i mapku budov školy.

Většina aplikací se soustředí převážně na studenty, všechny aplikace obsahují přehled předmětů, správu zkuškových termínů a rozvrh, každá v jiném formátu. UPlikace, Moje UHK a StuduJU poskytují různá upozornění, avšak upozornění na rozvrhové akce poskytne pouze Univerzita – mobilní klient IS STAG. Ta pak jako jediná poskytne také vyhledávání a přidávání přátel.

2.2 Dotazník

Základem vývoje mobilní aplikace je volba jejích funkcí. Jak již bylo zmíněno v úvodu, velké množství informací poskytovaných v IS/STAG se může jevit jako nepřehledné, obzvláště v mobilní verzi. A právě protože IS/STAG poskytuje velké množství funkcí, bylo potřeba zjistit, které jsou pro studenty nejdůležitější. Z tohoto důvodu byl jako součást práce rozeslán dotazník studentům Jihočeské Univerzity.

Dotazník zjišťuje názor dotázaných studentů na požadovanou funkcionalitu mobilní aplikace pro IS/STAG. Byl sestaven za pomoci Google Forms na základě klíčových služeb poskytovaných samotným IS/STAG a služeb, které poskytují již dostupné aplikace viz. kapitola 2.1. K dotazovaným funkcím byly také přidány funkce, které žáci již nějakou dobu požadovali po samotném IS/STAG. Poté byl dotazník zaslán pomocí emailu sekretářkám jednotlivých fakult, které byly požádány o rozeslání dotazníku studentům příslušné fakulty.

Výzkumu se celkově zúčastnilo 101 studentů studujících na následujících fakultách: Přírodovědecká fakulta, Fakulta rybářství a ochrany vod, Teologická fakultě a Zdravotně sociální fakulta. Dotazovaní studenti pocházeli z bakalářských, magisterských i doktorských studií. Dotazník byl poté ručně vyhodnocen pomocí programu Microsoft Excel, jelikož data poskytnutá Google Forms byla nevhodně zpracovaná. Důvodem tohoto nevhodného zpracování byly otázky, na které studenti slovně odpovídali. Příkladem takové otázky je obor studenta kde jej lze napsat několika způsoby.

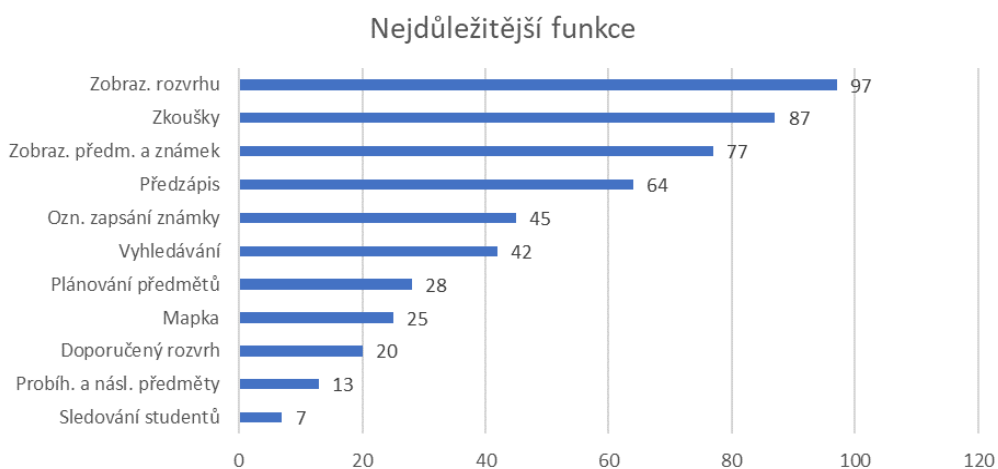
Fakulta	Bakalářské studium	Magisterské studium	Doktorské studium	Celkový počet studentů
Přírodovědecká fakulta	48	24	10	82
Fakulta rybářství a ochrany vod	11	3	0	14
Teologická fakulta	3	1	0	3
Zdravotně sociální fakulta	1	0	0	2
Celkový počet studentů	63	28	10	101

Tabulka 1 - Zúčastněné fakulty a studenti

2.3 Výsledky dotazníku

Sestavený dotazník má dvě části. V první části studenti vyplňovali základní informace o studiu a také volili důležité funkce aplikace. Mezi dotazované informace o studiu patřila fakulta, na které studují, ročník studia a typ studia, tudíž zda studují bakalářský, magisterský či doktorský obor. Výsledné počty zúčastněných studentů zobrazuje Tabulka 1. V druhé části dotazníku již studenti vybírali, jak moc jsou pro ně jednotlivé funkce důležité.

V první části dotazníku bylo již výše zmiňované zjištění důležitých funkcí mobilní aplikace. V této otázce měli studenti vybírat pět funkcí, které jsou pro ně nejvíce potřebné. Jako nejvíce požadované funkce se ukázaly: zobrazení rozvrhu, předmětů a známek přihlášeného studenta, dále možnost zápisu a zobrazení termínů zkoušek. Také se ukázalo, že mezi vysoce požadované funkce se patří možnost předzápisu. Mezi málo požadované funkce pro mobilní aplikaci se zařadilo zobrazení sledovaných studentů a zobrazení právě probíhajících předmětů. Níže zobrazený graf ukazuje konečný počet hlasů pro jednotlivé funkce. Na základě tohoto grafu bylo zvoleno pořadí vyvíjených funkcí aplikace.



Graf 1- Nejdůležitější funkce dle odpovědi studentů

Druhá část dotazníku se již věnuje důležitosti jednotlivých funkcí v mobilní aplikaci STAG. Důležitost jednotlivých funkcí je popsána stupnicí 1 (*požadované*) – 5 (*nepožadované*). Z této části vyplynulo, že zobrazení rozvrhu, předmětů a známek přihlášeného uživatele, musí být nedílnou součástí mobilní aplikace. Důležitost zobrazení rozvrhu označila většina dotázaných studentů (více než 85 %) jako 5 (*požadované*). V případě důležitosti zobrazení předmětů a známek přihlášeného uživatele označilo 65 % studentů jako 5 (*požadované*) a dalších 22 % jako 4 (*spíše požadované*).

Názor studentů na funkci vyhledávání různých informací je ve většině případů nerozhodný. Nejvíce potřebné shledali studenti vyhledávání předmětů, které označilo 88 % dotázaných studentů jako nutné. V případě vyhledávání učitelů jsou odpovědi 4 (*spíše požadované*) či 3 (*neutrální*) a vyhledávání studentů se ukázalo jako 2 (*spíše nepotřebné*). Odpovědi na vyhledávání místností a jejich rozvrhu se jeví v zásadě jako nerozhodné. Jelikož názory dotázaných studentů na tyto funkce jsou rovnoměrně rozprostřeny, bylo rozhodnuto, že funkce vyhledávání budou implementovány v mobilní aplikaci STAG.

Další z dotazovaných funkcí je sledování studentů. Tato funkce měla sloužit, jako pomocník, kde si přihlášený uživatel zadá osobní číslo požadovaného studenta. Poté by si mohl přes seznam sledovaných studentů otevřít informace o daném studentovi místo toho, aby často používal funkci vyhledávání studentů. Avšak jak je vidět na Graf 2, podle názoru dotázaných studentů se tato funkce ukázala jako málo vyžadovaná, právě z tohoto důvodu bylo rozhodnuto, že tato funkce bude v práci vyvíjena až mezi posledními.

Funkce související s termíny zkoušek jsou podle odpovědí zúčastněných studentů obzvláště důležité. Mezi tyto funkce patří zobrazení vypsanych i zapsanych zkoušek přihlášeného uživatele, zapsání se na vybraný zkouškový termín a oznámení na nastávající zapsaný zkouškový termín. Jako 5 (*požadované*) zvolilo více než 90 % studentů zobrazení zkouškových termínů a více než 85 % studentů pak požadovalo

zapisování na zkouškové termíny. Pouze o něco méně studentů by využilo oznámení pro termín zkoušky. Díky takto vysokému zájmu byly tyto funkce začleněny do základní funkcionality mobilní aplikace STAG.

Z *Graf 2* vyplývá, že jedna z více požadovaných funkcí je oznámení přihlášenému uživateli zapsání známky předmětu. I navzdory velkému zájmu ze strany studentů o tuto funkci nebude možno jí uskutečnit jako součást funkcí mobilní aplikace STAG. Jedním z důvodů je, že samotné webové služby IS/STAG neposkytují žádnou ze služeb, která by byla použitelná pro realizaci této funkce. Tato funkce je i přesto proveditelná, ale pouze nepraktickým způsobem. Ten spočívá v neustálém dotazování se API na zapsané známky, které by se poté srovnaly se záznamem z posledního dotazu. Právě dotazování se API by bylo velice problematické, jelikož službu, která poskytuje zapsané známky je možné volat pouze přihlášeným uživatelem, tudíž by na pozadí mobilního zařízení neustále pracovala aplikace s přihlášeným uživatelem. Z těchto důvodů bylo rozhodnuto, že tato funkce nebude v mobilní aplikaci podporována.

Další z funkcí, které studenti hodnotí jako potřebné je možnost předzápisu. Je to funkce dovolující přihlášenému uživateli předzapsat si předměty, které by chtěl v následujícím semestru studovat. Tato funkce je dostupná pouze v období několika týdnů před samotným zápisem. Bohužel i přes velký zájem ze strany studentů tuto funkci nelze implementovat, jelikož webové služby nad IS/STAG neposkytují možnost předzápisu. Poskytovaná je pouze možnost zápisu a odzápisu rozvrhových akcí studenta, kterou je možné volat pouze rolemi: Studijní referentka, Administrátor a Prorektor. Z tohoto důvodu nemůže být funkcionality součástí této bakalářské práce.

Funkce zobrazení právě probíhajícího a následujícího předmětu má sloužit přihlášenému uživateli, jako rychlé ujištění, zda předmět neodpadl a zároveň je u něj poznámka, ve které budově a učebně se daný předmět vyučuje. Názor studentů na tuto funkci je spíše kladný, avšak většina dotázaných využila neutrální odpovědi. Z důvodu takto nízkého zájmu o tuto funkci bude funkce zařazena mezi méně potřebné funkce při vývoji mobilní aplikace.

Jedna ze zamýšlených funkcí byla také mapa areálu Jihočeské univerzity. Student by si mohl na mapě zobrazit veškeré budovy Jihočeské univerzity, ale také by sloužila jako pomocník pro zorientování. V případě, že by si student nebyl jistý, ve které budově se nachází daná místnost, měl by možnost právě podle zadané místnosti najít budovu, kde se daná místnost nachází. Všechny možnosti byly zastoupeny velice podobným množstvím hlasů. A právě kvůli nedostatečnému zájmu ze strany dotázaných studentů, bude tato funkce vyvíjena mezi posledními funkcemi mobilní aplikace STAG.

IS/STAG jako takový nepodporuje možnost naplánovat si seznam předmětů na budoucí semestry. Z toho důvodu je jedna z navrhovaných funkcí mobilní aplikace plánování seznamu předmětů na budoucí semestry. Jde o funkci, kde si uživatel vybere, jaký předmět by chtěl studovat v daném semestru. Takto si je uživatel schopný přehledně naplánovat všechny budoucí semestry, poté si ho uživatel bude moci uložit. Nevýhodou této funkcionality může být to, že rozvrhy předmětů se neplánují takto dopředu. Z toho důvodu může nastat situace, kdy se některé předměty budou překrývat či z důvodů nedostatečného počtu studentů nebudou otevřeny. Ohlasy studentů na tuto plánovanou funkci byly spíše kladné, a proto bude tato funkce vyvíjena hned po více chtěných funkcích.

Poslední dotaz, na který studenti odpovídali je možnost navrnutí doporučeného rozvrhu na začínající semestr. Tato funkce by sloužila především pro studenty, kteří mají problém se zapisováním předmětů. Každé specializaci a ročníku by byl vytvořen pomocný rozvrh, podle kterého by si mohli zvolit předměty. Avšak z nedostatečného zájmu dotázaných studentů vyplívá, že by tato funkce byla zbytečná, a tak byla zařazena k posledním vyvíjeným funkcím.

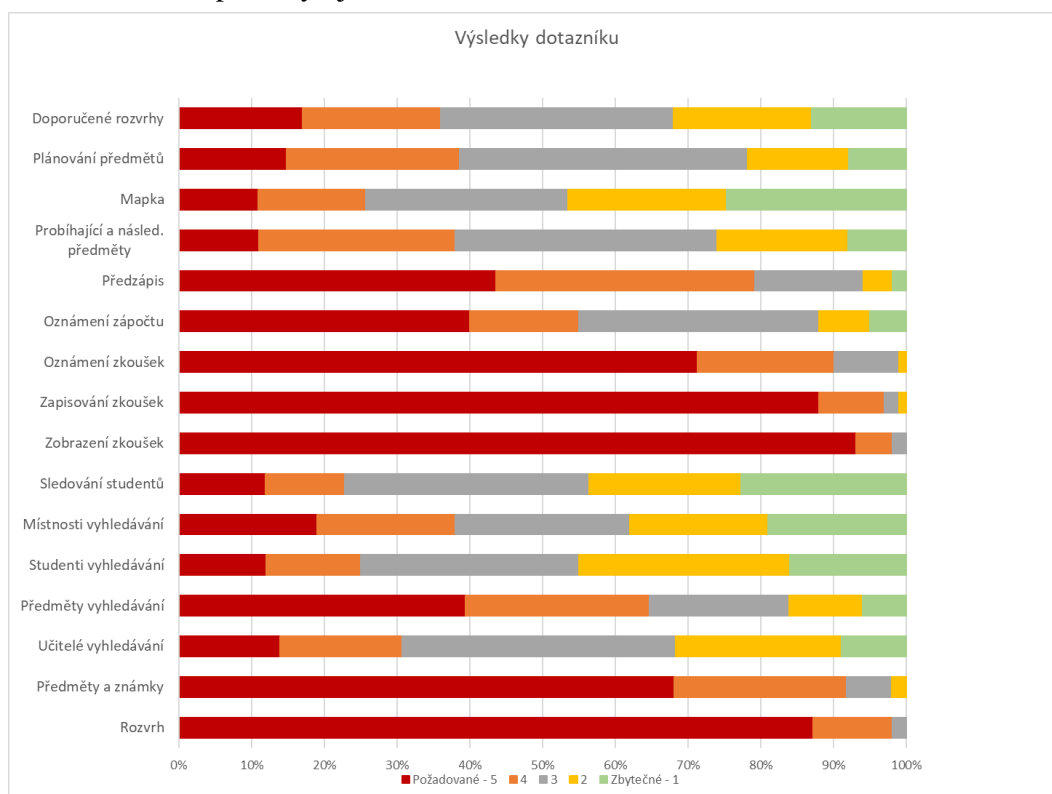
V rámci dotazníku měli zúčastnění studenti možnost vyjádřit své vlastní návrhy na funkcionalitu mobilní aplikace STAG. Jedním z návrhů je funkce pro kontrolu splněných podmínek. Tato funkce by byla určitě pro studenty přínosná, avšak jelikož byla zmíněna pouze jedním studentem tak by její praktické využití bylo velice nízké.

Mezi několikrát zmíněný návrh také patřilo oznámení při změně rozvrhu, důvodem bylo především aby studenti nebyli zmateni, pokud nastane např. přesunutí předmětu do jiné učebny či odpadnutí předmětu. Taková funkce by jistě byla využívána, avšak její realizace je z větší části komplikovaná. Její implementace by byla velice podobná již výše zmiňovanému oznámení při zapsání známky předmětu, a právě ze stejných důvodů nebude tato funkce podporována v mobilní aplikaci STAG. Dalším z návrhů byla možnost přidat externí přednášky do rozvrhu přihlášeného uživatele. Tato funkcionalita bohužel ze strany API IS/STAG není možná, jelikož IS/STAG neposkytuje možnost zapisování externích přednášek. Mezi další špatně realizovatelné funkce patří jeden z návrhů dotazovaných studentů:

„Propojení aplikace s menzou JU, a knihovním systémem lib.jcu.cz“

Takovéto funkce by byly realizovatelné pouze odkazem na dané stránky, což v rámci mobilní aplikace není problém, avšak ani jedna ze zmiňovaných webových stránek nepodporuje mobilní zobrazení. Z tohoto důvodu tyto funkce nebudou implementovány v mobilní aplikaci. Požadavek na funkci tzv. našeptávání kdy se program uživateli snaží napovědět hledaný výraz není vůbec překvapivý, jelikož skoro každý program pro vyhledávání ji má implementovanou. Bohužel toto je také jedna z funkcí, kterou API IS/STAG nepodporuje z důvodu, že je to webová služba vracející odpověď na zadaný řetězec. Tím pádem by řešení této funkce bylo neustálé dotazování se na webový server, které by v případě více vyhledávajících uživatelů mohlo způsobit zahlcení webového serveru. Funkčním řešením by bylo, pokud by daná funkce odesílala dotaz po více znacích než pouze jednom. Docílilo by se tím menší zátěže na server a funkce by se ukázala splnitelnou. Problém však nastává při dotazování se na API IS/STAG. Pokud uživatel při vyhledávání např. učitele s křestním jménem Anna zadá pouze “Ann“, tak API IS/STAG vrátí pouze prázdný řetězec. Právě z tohoto důvodu poskytují vyhledávací služby IS/STAG možnost použití zástupného znaménka %, které při vyhledávání indikuje, že za zadaným řetězcem mohou

následovat další znaky. Z důvodu podpory této možnosti bylo rozhodnuto, že tato funkce nebude v práci vyvíjena.



Graf 2- Výsledky dotazníku na funkce mobilní aplikace

3. IS/STAG

IS/STAG je zkratka pro informační systém studijní agendy zaměřený na vysoké školy a vyšší odborné školy. Tento informační systém není určený pro podporu studia, ale pouze pro jeho administraci. IS/STAG je vyvíjen Centrem informatizace a výpočetní techniky – Střediskem informačních systémů na Západočeské univerzitě v Plzni, kde byl také roku 1993 poprvé nasazen. K datu 9.4. 2020 vyživá tento systém 14 škol v České republice (1).



Obrázek 1 - Logo IS/STAG

Pro přístup k aplikacím IS/STAG je možné využít tři různých způsobů připojení. Prvním způsobem je nativní klient pro MS Windows, pomocí kterého se tvoří formuláře a tiskové sestavy. Tento klient je především určen pro používání specializovanými pracovníky (3). Dalším způsobem je pomocí portálového rozhraní, které je především určeno pro studenty a učitele (4). Posledním podporovaným způsobem, jenž lze použít pro připojení k aplikacím IS/STAG je prostřednictvím webových služeb. Tyto služby jsou pak využívány pro připojování z jiných systémů (5). Pro vyvíjenou aplikaci bylo využito právě těchto webových služeb, které jsou dále více popsány.

Po přihlášení do portálového rozhraní získá uživatel přístup do aplikací dostupných pro jeho roli. Většina uživatelů IS/STAG jsou především studenti, kteří využívají výše zmíněného portálového klienta, právě tito studenti jsou cílovou skupinou vyvíjené aplikace.

Moje studium je zásadní záložkou pro přihlášeného studenta, ve které se nachází přehled o celém studiu studenta. Nachází se zde několik podstránek souvisejících se studiem přihlášeného studenta. První z těchto podstránek je *Průběh studia*, na této podstránce jsou informace o předmětech, které si student za dobu studia zapsal. Poskytuje informace o známkách a zápočtech zapsaných předmětů a jejich vypsané či zapsané termíny. Nachází se zde také rozvrh studenta na aktuální i minulé semestry. Tento rozvrh lze zobrazit podle data, dnů v týdnu nebo kombinovaně (6, s. 70-71).

Další podstránkou je *Vizualizace studia*, která zobrazuje podrobnosti k průběhu studentova studia. Lze zde najít počet kreditů studenta, kolik kreditů celkem potřebuje a kolik mu jich zbývá získat. Na podstránce je také graficky znázorněno kdy student absolvoval, který předmět a je zde vidět i stav samotného předmětu (6, s. 75-79).

Podstránka *Zbývající povinnosti* se také zabývá průběhem studia, avšak poskytuje i informace o dosud nesplněných povinnostech. Mezi poskytované informace patří počet kreditů, které student potřebuje ještě získat, nesplněné povinné a povinně volitelné předměty a také informace o časových prostředcích pro dokončení studia (6, s. 79-82).

Zápis na termíny je podstránka, která umožňuje studentovi zapsat se na vypsané zkouškové termíny či se z nich následně odepsat. Poskytuje také možnost stáhnout si historii registrací na termíny ve formátu CSV. Pro lepší přehled zobrazuje také náhled na zapsané předměty a jejich hodnocení (6, s. 82-85).

V podstránce *Moje údaje* jsou uvedeny veškeré osobní údaje studenta, nachází se zde i seznam a nastavení plateb, bankovní účet studenta, pokud jej používá a přehled poplatků za studium. Mezi zobrazované osobní údaje studenta patří jméno a příjmení, rodné číslo, datum narození, e-mailová adresa, místo a stát narození a také trvalé bydliště studenta (6, s. 85).

Pro předzápis jsou k dispozici tři podstránky – *Předzápis*, *Grafický předzápis* a *Kroužkový předzápis*. Podstránka *Předzápis* umožňuje klasický předzápis, kde v horní polovině je přehled o povinných a povinně volitelných přehledů, ze kterého si student

může vybírat předměty pro předzázpis. V dolní polovině se zobrazuje rozvrh studenta na plánovaný semestr (6, s. 99-101). *Grafický předzázpis* funguje na podobném principu, avšak jeho forma je velice podobná výše zmíněné Vizualizaci studia (6, s. 108-110). *Kroužkový zápis* je podstránka umožňující studentovi zapsat a odepsat se z kroužků (6, s. 123-127).

Správu témat vysokoškolských prací zajišťuje podstránka *Témata VŠKP*, zde lze vyhledat vypsaná témata či je vypisovat a následně upravovat (6, s. 273-275). Pro zadání podkladů je možné využít podstránku *Kvalifikační práce*. Na této podstránce může po schválení zadání student doplňovat během studia informace o zadané kvalifikační práci. Umožňuje také odevzdání kvalifikační práce v elektronické podobě a zde lze také nalézt posudky vedoucího práce a oponenta (6, s. 133). Na podstránce *Kontrola plagiátorství* se nachází výsledky kontrol kvalifikačních prací, kde si může student procházet výsledky pouze svých prací (6, s. 423-425).

V záložce *Moje studium* se také nacházejí podstránky pro *ECTS výjezdy* (6, s. 400) a *ECTS příjezdy* (6, s. 390) studenta, odkaz na externí stránku pro hodnocení výuky a správa studentských žádostí (6, s. 148-149). Lze zde nastavit automatické zasílání zpráv a upomínek (6, s. 157-158). V této záložce se také nachází podstránka pro nahlížení do spisu (6, s. 159-161).

Druhou podstatnou záložkou je *Prohlížení*, ta poskytuje studentovi vyhledání dodatečných informací. Mezi tyto informace patří Programy a obory, Předměty, Pracoviště apod. Pro vyhledání samotné informace musí student nejdříve zadat požadované parametry. Na jejich základě se studentovi vrátí nalezené shody. Obrázek 2 zobrazuje rozcestník pro Prohlížení, tyto služby jsou přístupné i veřejnosti až na výjimku vyhledávání studentů (6, s. 371-387).

Obrázek 2 - Záložka Prohlížení

3.1 Webové služby IS/STAG

IS/STAG má připravené vlastní rozhraní IS/STAG určené pro komunikaci s externími klienty. Právě tohoto rozhraní je využito pro zobrazení dat IS/STAGu v mobilní aplikaci STAG. Rozhraní je volně přístupné na webovém serveru vysoké školy/univerzity a pro zajištění bezpečného přenosu mezi klientem a serverem je využito protokolu HTTPS (7, s. 15). Tato webová služba poskytuje komunikaci pro své klienty dvěma způsoby. Pomocí rozhraní REST určeného především pro jednodušší klienty nebo pomocí webové služby realizované prostřednictvím protokolu SOAP, přesněji SOAP 1.1 a WSDL 1.1, které je určené pro komunikaci mezi stroji (7, s. 7-9).

Rozhraní REST je obousměrné a komunikuje pomocí HTTP protokolu. Pomocí HTTP hlavičky Content-type se určuje vstupní formát dat a pomocí další HTTP hlavičky Accept klient určuje očekávaný výstupní formát dat (7, s. 10-11). Jako vstupní i výstupní formáty dat jsou u všech webových služeb podporovány formáty: XML, JSON, YAML. U některých webových služeb je možnost výstupu v podobě formátu: CSV a XLSX. Jsou to především služby, které jako odpověď vrací data strukturovaná do “tabulek“ (7, s. 14-15).

IS/STAG obsahuje dvě verze rozhraní REST, z nichž původní REST rozhraní je od začátku roku 2020 neaktivní (7, s. 10). Proto je rámci práce využito REST2 rozhraní ke komunikaci s webovým serverem IS/STAG Jihočeské univerzity. K této volbě došlo z důvodu jednoduchosti a intuitivnosti využívání komunikace přes protokol HTTP. Specifikace pro REST rozhraní jsou k dispozici v jazyce WADL či ve formátu OpenAPI, který je momentálně podporován pouze experimentálně (7, s. 8).

3.2 Využití webové služby nad IS/STAG

Využití webové služby nad IS/STAG poskytované Jihočeskou univerzitou nacházející se na webové adrese:

<https://stag-ws.jcu.cz/ws/services/rest2/>

Informace o poskytovaných REST2 službách nejsou dostupné na stránkách Jihočeské univerzity, a proto je v práci využito dokumentace z webových stránek Univerzity Hradce Králové (8). Za tuto adresu se následně připojí název skupiny, do které náleží požadovaná služba a poté se připojí její název. Pokud požadovaná služba potřebuje zadat parametry, připojí se za otazník.

`stag-ws.jcu.cz/ws/services/rest2/predmety/getPredmetInfo?katedra=UAI`

The diagram shows the URL `stag-ws.jcu.cz/ws/services/rest2/predmety/getPredmetInfo?katedra=UAI` with four colored brackets underneath it. A red bracket underlines the first part: `stag-ws.jcu.cz/ws/services/rest2/` and is labeled "adresa serveru". A red bracket underlines the next part: `predmety` and is labeled "název skupiny". An orange bracket underlines the next part: `getPredmetInfo` and is labeled "název služby". A green bracket underlines the final part: `?katedra=UAI` and is labeled "parametry".

Obrázek 3 - Rozbor URL adresy při dotazu na webový server IS/STAG

Některé poskytované webové služby požadují přihlášení. V této práci je přihlašování do IS/STAG vyřešeno pomocí doporučeného postupu (7, s. 19-20). Ten spočívá v přesměrování uživatele na přihlašovací stránku serveru IS/STAG. Po úspěšném přihlášení server vrátí ticket pod názvem `stagUserTicket`, který se poté v programu uloží ve formě cookie. Následně se tato cookie vkládá do hlavičky volání webové služby požadující přihlášení (7, s. 17).

Odpovědi z webového serveru IS/STAG jsou v celé vyvíjené aplikaci přijímány ve formátu JSON. Na následujícím obrázku je ukázka odpovědi z webového serveru IS/STAG ve formátu JSON. Daná ukázka je dotaz `najdiPredmety` se zadanými parametry `pracoviste=UAI` a `zkratka=891`.

```
{
  "predmetKatedry": [
    {
      "katedra": "UAI",
      "zkratka": "891",
      "rok": "2019",
      "nazev": "Bakal. diplomová praxe II.",
      "semestr": null,
      "maVyuku": null,
      "vyukaZS": null,
      "vyukaLS": null,
      "jazyk1": null,
      "jazyk2": null,
      "jazyk3": null,
      "jazyk4": null,
      "nabizetPrijezdyEcts": null,
      "pocetStudentu": null,
      "aMax": null,
      "bMax": null,
      "cMax": null,
      "aSkut": null,
      "bSkut": null,
      "cSkut": null
    }
  ]
}
```

Obrázek 4 - Odpověď na dotaz ve formátu JSON

3.2.1 Veřejné služby (bez nutnosti přihlášení)

`predmety/najdiPredmety`

Vyhledá předmět podle zadaných parametrů. Mezi tyto parametry patří název, pracoviště, zkratka a rok konání předmětu. Pro vyhledávání lze použít zástupný znak %. Vrací veškeré zadané i nezadané parametry.

`predmety/getPredmetInfo`

Vyhledá předmět podle zadané katedry a zkratky. Vrací rozšířené informace o předmětu, včetně sylabu. Vrací vždy pouze jeden předmět, proto není možno tuto službu použít pro vyhledávání.

`predmety/getPredmetyByOborFullInfo`

Vyhledá předměty podle oboru. Pro vyhledávání se používá identifikační číslo oboru či číslo oboru. Vrací seznam předmětů podle zadaného oboru, s kompletními informacemi daných předmětů.

`predmety/getPredmetyByUcitel`

Vyhledá předměty podle identifikačního čísla učitele. Vrací seznam předmětů s některými jejich informacemi.

`ucitel/najdiUcitelePodleJmena`

Na základě zadaného jména a/nebo příjmení vrátí STAGovské ID učitele, jméno, příjmení, tituly a zda je zaměstnanec Jihočeské univerzity. Pro vyhledávání lze použít zástupný znak %.

`ucitel/getUcitelInfo`

Vrátí informace o učiteli podle identifikačního čísla učitele. Mezi vrácené informace patří jméno, příjmení, tituly, katedra, email a telefonní číslo.

`rozvrhy/getRozvrhByPredmet`

Vrátí rozvrh předmětu na základě kombinace katedry a zkratky předmětu. Pokud není zadán parametr rok či semestr, tak vrácený rozvrh je na aktuální rok a

semestr. Vrácený rozvrh obsahuje i popis samotných rozvrhových akcí jako např. název, budova, místnost.

`rozvrhy/getRozvrhByUcitel`

Vrátí rozvrh učitele na základě osobního identifikačního čísla. Samotný vrácený rozvrh je ve stejné formě jako v případě služby `rozvrhy/getRozvrhByPredmet`.

3.2.2 Služby pro uživatele (s nutností přihlášení)

`predmety/getPredmetyByStudent`

Vrátí předměty studenta na základě jeho osobního čísla a zadaného semestru. Pro upřesnění může být využito nepovinného parametru rok. Pokud není nastaven rok služba automaticky vrací předměty na aktuální rok. Využít lze i dalšího nepovinného parametru `nevracetUznane`. Když je tento parametr nastaven na TRUE tak tato služba vrátí pouze předměty, které studentovi nebyly uznány.

`student/najdiStudentyPodleJmena`

Vyhledá studenty na základě zadaného jména a/nebo příjmení. Vrátí seznam studentů odpovídajících zadaným parametrům spolu se stručnými informacemi o vyhledaných studentech. Pro vyhledávání lze použít zástupný znak %.

`student/getStudentInfo`

Vrátí informace o studentovi na základě jeho osobního čísla. Pokud se dotazuje přihlášený student sám na sebe či se dotazuje někdo s vyšší rolí, tak služba navrátí také položky: email, financování, číslo karty a pohlaví.

`student/getStudentPredmetyAbsolvoval`

Vrátí předměty studenta na základě jeho osobního čísla. Mezi vrácené informace patří také zda student daný předmět absolvoval, jeho známka z předmětu a počet kreditů předmětu. Tuto službu lze volat pouze pro osobní číslo přihlášeného studenta, a proto s ní nelze zobrazovat informace o předmětech jiných studentů.

rozvrhy/getRozvrhByStudent

Vrátí rozvrh studenta na základě osobního čísla. Vrácený rozvrh je ve stejné formě jako v případě služby rozvrhy/getRozvrhByPredmet.

terminy/getTerminyProStudenta

Vyhledá zkouškové termíny na základě studentova osobního čísla. Mezi vrácené termíny patří i termíny zrušené na kterých je student zapsán. U jednotlivých termínů vrací informace o zkoušce jako např. datum, budovu, místnost, obsazenost a limit obsazenosti. Mezi vrácené informace také patří, zda je tento pokus opravný a zda jej má student zapsán.

terminy/odhlasStudentaZTerminu

Odhlásí studenta ze zapsaného termínu pomocí jeho osobního čísla a identifikačního čísla termínu. Vrací textový řetězec s informací, zda bylo odepsání úspěšné či popřípadě vrátí informaci proč se odepsání nepovedlo.

terminy/zapisStudentaNaTermin

Zapiše studenta na vybraný termín pomocí jeho osobního čísla a identifikačního čísla termínu. Vrací textový řetězec s informací, zda bylo zapsání úspěšné či popřípadě vrátí informaci proč se zapsání nepovedlo.

4. Ionic Framework

Ionic Framework je framework pro vývoj hybridních aplikací², zaměřených na mobilní a desktop aplikace. Hybridní aplikace používá jeden kód pro všechny podporované platformy, tudíž nejsou psané pro jednotlivé platformy pomocí specializovaných SDK³, namísto toho využívají jazyky jako JavaScript, HTML a CSS. Tento framework využívá pro běh aplikace full-screen prohlížeč, který je podporován na všech platformách. Uživatel tento prohlížeč ani nezaregistruje, jelikož se chová jako nativní aplikace (9, s. 5).



Obrázek 5 - Logo Ionic Frameworku

Nepříjemností při vývoji hybridní aplikace se však může stát závislost na frameworku, jelikož hybridní aplikace se píše pouze pomocí frameworků a pluginů. Aplikace je pak více složitá a zároveň vzniká závislost na třetích stranách, které tyto pluginy poskytují. Aplikace psaná v Ionic frameworku využívá mix nativního a webového kódu což poskytuje možnost využít standartně hardware přístroje, ale poskytuje i možnost psát webový kód. Výhoda v podobě psaní jednoho kódu pro více platform má také jeden velký nedostatek, tím je velký overhead⁴. Proto hybridní aplikace kladou velké požadavky na systém, a tím pádem se nedoporučuje je používat na aplikace s 3D grafikou či velkou zátěží na systém. Pro takové aplikace je lepší využít nativní programování (9, s. 12).

Při programování nativních aplikací se musí vyvíjet kód pro každou platformu zvlášť a tím pádem je potřeba více času, pokud má být daná aplikace dostupná na více

² Aplikace kombinující prvky nativních a webových aplikací.

³ Software development kit – sada vývojových nástrojů pro specifickou platformu

⁴ Časové a paměťové zdroje potřebné k běhu aplikace

platformách. Proto je často potřeba pro vývoj nativní aplikace více kodérů, každého specializovaného na jinou platformu. Jak už bylo zmíněno výše, nativní programování je výhodné pro aplikace, které mají větší požadavky na systém, a to proto, že kód psaný v nativním jazyce systému je o poznání rychlejší než kód napsaný nezávisle na platformě (9, s. 14-15).

Po zvážení pro a proti na základě předchozích dvou odstavců, byl pro vyvíjenou aplikaci zvolen přístup hybridního programování. Ionic framework byl vytvořen roku 2013 společností Drifty Co. Ze začátku fungoval pouze nad Angular 1 a Apache Cordova bez použití webových komponent a bez možnosti vytvářet PWA⁵. Ionic 2 přinesl podporu Angular 2 avšak Ionic samotný se nijak zásadně pro uživatele nezměnil, to platilo i pro Ionic 3, který již podporoval Angular 4.

Od verze Ionic 4 se základní koncept zcela změnil. Nově se používají jako základ aplikace webové komponenty, které si může uživatel naprogramovat či využít již předpřipravených komponent. Pro vývoj lze zvolit dva přístupy – nativní a PWA (9). Pro programování pomocí PWA se využívá JavaScript, HTML a CSS. Byla přidána možnost využít i jiné frameworky pro front-end kromě Angular a těmi jsou React a Vue.js (11), také byla přidána možnost použít Capacitor namísto Apache Cordova (12). Pro vyvíjenou aplikaci bylo zvolena kombinace Angular 4.1 a Apache Cordova 4.2.

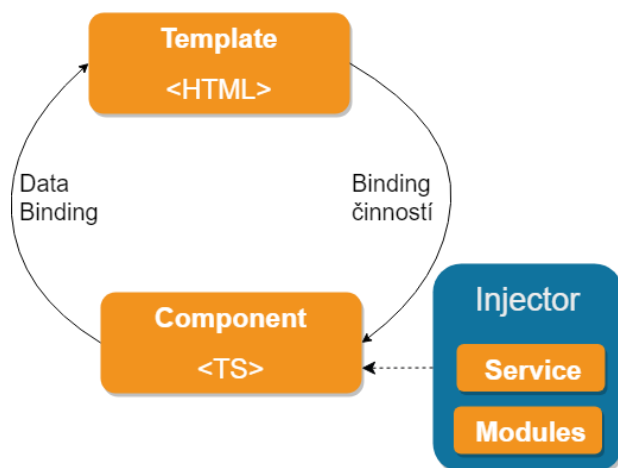
Grafická část aplikace je psaná za využití HTML a CSS, kde lze využít data-bindingu s TypeScriptem, který poskytuje Angular (13). K usnadnění tvorby grafické části aplikace se využívají komponenty zajišťované Ionic frameworkem. Ten však poskytuje i možnost vytvořit vlastní komponenty, které pak lze používat souběžně s předpřipravenými komponentami. Výhoda používání webových komponent je především v rychlejší tvorbě grafické části aplikace a to především, protože pro danou

⁵ Progresivní webové aplikace – aplikace, které vypadají jako webové stránky, ale poskytují nativní funkcionalitu

komponentu stačí napsat pouze jeden kód, který pak lze jednoduše využít v rámci celé aplikace.

Samotný Ionic framework tedy zajišťuje vizuální chování aplikace, a především se stará o to, aby aplikace byla responzivní. K tomu používá i GPU přístroje které řeší animační změny v CSS. Také poskytuje vygenerované aplikaci pocit, že aplikace byla napsaná nativním způsobem, tedy že připomíná grafické prostředí dané platformy.

4.1 Angular



Obrázek 6 - Struktura aplikace

Angular je framework, který je implementací jazykové specifikace podle ECMAScript 6 a je zpětně kompatibilní s JavaScriptem. Jeho největší předností je umožnit aplikacím napsaných v jazyce HTML využívat funkce pro binding dat. Poskytuje také rozdělení aplikační logiky od grafické části aplikace, čímž usnadňuje práci při psaní aplikace pomocí MVVM architektury.

Framework Angular zajišťuje routing⁶ aplikace a poskytuje také různé animace a přechody. Jednou z klíčových funkcionalit Angular je tzv. Two-Way Data Binding, který zajišťuje dynamické zobrazování dat na stránce. Když například uživatel přepíše

⁶ Propojení URL adresy s určitou webovou stránkou

na webové stránce nějakou hodnotu, tak se daná hodnota následně upraví i v modelu a naopak. Pro zobrazení takto bindnutých dat se používají template psané v HTML. Další podstatnou funkcí je dependency injection, jedná se o návrhový vzor, který se stará o získávání modulů, jenž potřebují komponenty ke své činnosti (14).

4.2 Apache Cordova

Pro vývoj multiplatformní aplikace je ve frameworku Ionic využito opensource frameworku Apache Cordova. Aplikace se generuje pomocí tohoto frameworku, který pro vývoj nativních aplikací běžících na různých platformách používá webových technologií HTML, CSS a JavaScript. Samotná aplikace tak může fungovat jako webová aplikace s tím rozdílem, že díky frameworku Apache Cordova může přistupovat přímo k rozhraním mobilního zařízení.

Poskytované API funkcionality jsou: akcelerometr, foťák, kompas, geolokace, kontakty, soubory, úložiště, média, síť a notifikace (15). Z důvodu nekonzistence podpory HTML5 v mobilních prohlížečích využívá Apache Cordova HTML kód vkládaný do komponenty webView. Tato nativní komponenta se chová jako klasický prohlížeč s tím rozdílem, že uživatel nevidí bar pro zadávání URL ani žádné možnosti nastavení prohlížeče (16).

5. Popis praktické části

Aplikace byla vyvíjena ve vývojovém prostředí Visual Studio Community 2019, které poskytuje firma Microsoft a s využitím Ionic CLI. Framework Ionic umožňuje dva přístupy při debugování vyvíjené aplikace. Prvním způsobem je spuštění aplikace ve webovém prohlížeči počítače a druhým je spuštěním pomocí emulátorů či nahráním aplikace na samotné mobilní zařízení za pomoci tzv. live debugging (17). Z důvodu věrnějšího uživatelského zážitku při vývoji a testování bylo v této práci využito především debugování pomocí mobilního zařízení.

Prvním krokem při vývoji aplikace bylo zajistit komunikaci mezi serverem IS/STAG a mobilní aplikací. Z tohoto důvodu byla za pomoci Ionic CLI vygenerována klientská servisní komponenta pro komunikaci se serverem. Samotná komunikace je realizována pomocí asynchronních metod, které pomocí HTTPS odesílají dotazy metodou GET na server. Ten dotaz zpracuje a na základě dotazované služby a jejích parametrů vrátí odpověď.

Ačkoliv se může zdát, že se jedná o přímočarou a bezproblémovou cestu k propojení mobilní aplikace s poskytovatelem služeb, je potřeba ošetřit několik zásadních omezení.

Jedním z komunikačních problémů bývají pravidla CORS⁷. Například server IS/STAG vrátil tuto chybovou hlášku:

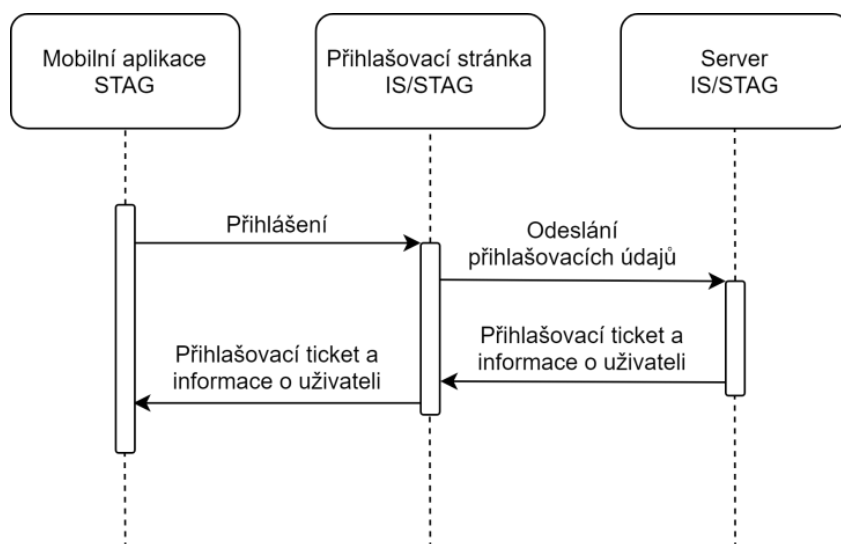
```
No 'Access-Control-Allow-Origin' header is present on
the requested resource.
Origin 'http://localhost:8100' is therefore not allowed
access.
```

To znamená že server má zakázáno přijímat dotazy, které nemají v hlavičce nastaveno `Access-Control-Allow-Origin`.

⁷ Cross-origin resource sharing – princip sdílení zdrojů webového serveru pro jiné domény

Zdrojem tohoto problému je Ionic framework, který nenastavuje správně hlavičku HTTP. Jako řešení byl využit plugin poskytovaný frameworkem Cordova – HTTP. Tento plugin využívá Apache Cordova pro komunikaci s HTTP servery a je schopný nastavit komplexní HTTP hlavičku (18). Následně bylo do servisy pro komunikaci se serverem IS/STAG napsáno několik základních metod pro aplikaci např. získání informací o studentovi. Další asynchronní metody byly přidány podle toho, pro které funkce aplikace byly vyžadovány.

Druhým řešeným problémem v servise `stag-api.service.ts` bylo přihlašování uživatele na server IS/STAG. Jak již bylo zmíněno v kapitole 3.2 bylo pro přihlášení studenta využito doporučeného postupu pro klienta webových služeb uvedeného v Referenční příručce pro webové služby nad IS/STAG (7, s. 19-20). Ten spočívá v tom, že klient otevře ve své aplikaci webovou stránku pro přihlášení do portálu IS/STAG, kde jako parametr `originalURL` zadá webovou stránku, ze které se klient odkazuje. Poté co se uživatel úspěšně přihlásí tak server vrátí vygenerovaný ticket uživatele a jeho informace zakódované pomocí Base64.



Obrázek 7 - Diagram přihlašování

Pro zobrazení webové stránky v aplikaci byl použit plugin InAppBrowser, poskytovaný frameworkem Apache Cordova. Ten umožňuje zobrazení webových stránek v samotné aplikaci bez nutnosti jí opouštět. Samotný prohlížeč poskytuje vlastní navigaci pro uživatele a zároveň nemá přístup k rozhraním mobilního zařízení poskytovaných frameworkem Apache Cordova (19). Ticket obdrženy po úspěšném přihlášení se následně v aplikaci uloží ve formě cookie s názvem WSCOOKIE, ta se pak používá při volání služeb vyžadujících přihlášení. Vrácené kódované informace studenta se použijí pro nastavení provideru `uzivatel.ts`, který udržuje informace o přihlášeném uživateli pro celou aplikaci, toho je docíleno pomocí návrhového vzoru singleton. Mezi udržované informace patří osobní číslo studenta, jeho role, informace vrácené službou `getStudentInfo`.

Po vyřešení problémů s dotazy na server IS/STAG se přistoupilo k samotné tvorbě aplikace. Při vytváření jednotlivých stránek aplikace se nejdříve vytvořila grafická část a poté se implementovala její funkcionality. Jednotlivé funkce byly vyvíjeny podle výsledků dotazníku ze kapitoly 2.3.

5.1 Vlastní komponenty

V práci bylo využito funkcionality frameworku Ionic poskytující možnost vytvořit vlastní komponenty, které byly poté využity na většině stránek aplikace. Komponenty byly vygenerovány pomocí Ionic CLI, přesněji `ionic generate component`. Pomocí vlastního kódu umístěného v souboru `components.module.ts` jsou pak obě komponenty distribuované ostatním stránkám. Tyto komponenty se stávají ze čtyř různých souborů: `component.html`, `component.scss`, `component.spec.ts` a `component.ts`.

HTML soubor se stará o zobrazení komponenty na stránce kam byla importována a soubor SCSS obstarává jakým způsobem se komponenta zobrazí. `Component.spec.ts` je soubor, který obsahuje specifikace dané komponenty. Obsahuje hlavní importy a

definici obsahující také podmínky vytvoření samotné komponenty. Poslední soubor obsahuje samotnou implementaci logiky komponenty.

5.1.1 Menu

Pro navigaci v aplikaci je využito komponenty s názvem menu, která se nachází na spodní straně obrazovky mobilního zařízení, poté co server IS/STAG úspěšně přihlásil uživatele. Komponenta obsahuje tři tlačítka pro pohodlnou navigaci v aplikaci. Levé tlačítko Info odkazuje na stránku obsahující základní informace o přihlášeném uživateli, prostřední tlačítko odkazuje na domovskou stránku a tlačítko napravo otevře menu aplikace.

Menu je realizováno pomocí předpřipravené komponenty ion-action-sheet poskytované frameworkem Ionic. Tato komponenta překryje svým obsahem stránku, na které byla vyvolána a zůstane aktivní, dokud ji uživatel sám nezavře. To je možné klepnutím na pozadí či volbou Zavřít ve vyvolaném dialogu. Toto překrytí obsahuje tlačítka s ikonkami a názvem, samotný přechod na danou stránku je pak realizován handlerem na tlačítku za pomoci routeru aplikace. Tlačítka, která odkazují na příslušné stránky jsou Vyhledávání, Moje známky, Zkoušky, Plánovač a jeho podmenu – Zobrazit a Naplánovat / Upravit, Odhlásit a Zavřít.

5.1.2 Rozvrh

Jelikož rozvrh se zobrazuje na hlavní stránce přihlášeného studenta i při využití funkce vyhledávání, bylo rozhodnuto že bude využito vlastní komponenty pro výraznou redukci kódu. Tato možnost byla podpořena i skutečností, že server IS/STAG vrací všechny rozvrhy ve stejném formátu. Při načítání stránky obsahující rozvrh je vidět nejdříve tlačítko pro zobrazení samotného rozvrhu které po zmáčknutí zobrazí rozvrh. Tato funkcionalita byla přidána z důvodu velikosti rozvrhu, jenž zabírá zhruba polovinu obrazovky, a proto když studenta rozvrh nezajímá a potřebuje si pouze zjistit informace nemusí zbytečně scrollovat přes zobrazený rozvrh.

Grafický vzhled byl zvolen na základě vzhledu rozvrhu na webových stránkách wstag.jcu.cz jelikož jsou na něj studenti již zvyklí. Rozvrhem lze posouvat do stran, a proto bylo možno zvolit větší rozměr právě kvůli pohodlnějšímu zobrazení na mobilním zařízení.

Při zavolání komponenty na stránce se musí nejdříve zjistit pro jakou funkci se má rozvrh zobrazit. Z tohoto důvodu má komponenta specifikované dva inputy: id a druh. Způsob vložení těchto inputů je vidět na Obrázek 8. Id určuje identifikátor studenta, předmětu, učitele či místnosti, pro který chceme rozvrh zobrazit. Input druh pak určuje, o který druh identifikátoru se jedná. Právě na základě tohoto inputu se rozhoduje, jaká služba bude volána ze serveru IS/STAG.

```
<app-rozvrh [id]="uzivatel.osCislo" druh="1"></app-rozvrh>
```

Obrázek 8 - Volání komponenty s inputy

Vrácený rozvrh se následně uloží do proměnné typu Map, kde jako klíč slouží název dne v týdnu a jako hodnota se ukládají jednotlivé rozvrhové akce, pro které je vytvořena třída popisující její data. Poté se nastaví jednotlivé dny a jejich výška, tedy maximální počet předmětů překrývajících se v daný den. Nastavení dnů spočívá v jejich převedení na vkládaný HTML kód, jenž se následně v aplikaci zobrazí. Jednotlivé rozvrhové akce mají svoji pozici nastavenou na základě následujícího vzorečku:

$$(hodina * hodinaPixely) + (minuta * minutaPixely) - výchozíPosun$$

Vzoreček se použije pro vypočítání začátku i konce rozvrhové akce. Nejdříve se vynásobí hodina a minuty kdy akce začíná či končí hodnotou definující počet pixelů dané časové jednotky. Poté se odečte výchozí posun, který udává kde začíná první hodina dne (čas - 8:00).

Nastaví se jim také barva pozadí na základě typu rozvrhové akce. Následně se nastaví výška gridu v HTML souboru podle výšky jednotlivých dní. Jelikož Ionic framework zakazuje externí vkládání HTML a CSS kódů muselo být využito API frameworku Angular DomSanitizer pro funkční zobrazení vkládaného kódu (20).

5.2 Domovská stránka

První vyvíjenou stránkou byla domovská stránka, jelikož bez ní by uživatel měl velice špatný přehled. Při zapnutí aplikace se uživateli ukáže domovská stránka pro uživatele, který není přihlášen na server IS/STAG. Tato stránka obsahuje jednoduché menu, kde se uživatel může přihlásit či přejít do funkce vyhledávání. Po klepnutí na tlačítko Přihlásit jehož funkcionalita realizuje výše popsané přihlášení a následném úspěšném přihlášení aplikace přejde na domovskou stránku pro přihlášeného studenta.

V záhlaví této stránky je pro přehlednost vypsáno osobní číslo a jméno přihlášeného studenta, v zápatí se pak zobrazuje menu pro navigaci v aplikaci. Samotné tělo stránky se pak sestává z již zmíněné komponenty rozvrh a výpisu předmětů, které má student zapsán na aktuální semestr.

Zobrazované předměty jsou na základě osobního čísla uživatele získávány ze serveru IS/STAG pomocí servisy `stag-api.service` přesněji metody `getPredmetyByStudent`. Poté jsou za pomoci třídy `Predmet` uloženy do listu `predmety`. Tento list je poté zobrazen pomocí kombinace two-way bindingu a funkce pro vypisování listů v grafické části aplikace `*ngFor` (21). Mezi zobrazované informace o předmětech patří jeho katedra a zkratka, název, statut vzhledem ke specializaci a množství kreditů, které lze za daný předmět získat.

5.3 Moje známky

Jelikož vypisování veškerých předmětů, které měl student v daném studiu zapsané pokaždé, když se načítá domovská stránka, by bylo zbytečným zatížením mobilního zařízení, byla vygenerována stránka jmenující se moje známky. Tato stránka poskytuje studentovi přehled o studovaných předmětech.

Pomocí metody `getUzivatelpredmetyAbsolvoval` ze servisy `stag-api.service` se získají požadovaná data ze serveru IS/STAG. Následně se podobně jako v předchozím případě uloží do listu, který je následně vypisován. Zobrazované

informace o předmětech jsou: katedra a zkratka, název, počet kreditů, známka, rok a semestr absolvování předmětu.

5.4 Info uživatele

Po přidání možnosti zobrazit si studované předměty byla vyvíjena funkce poskytující studentovi náhled na své informace. Stránka pomocí bindingu zobrazuje informace uložené v již zmíněném provideru uživatel, které se naplňují při přihlášení studenta. V první části stránky se nachází osobní informace o studentovi jeho role, osobní číslo, e-mail a v druhé části jsou informace o studiu: název oboru, fakulta, kód oboru a ročník, ve kterém se student nachází.

5.5 Zkoušky

Další vyvíjenou funkcionalitou mobilní aplikace bylo zobrazování zkouškových termínů a jejich zapisování případně odepsání. Jelikož při vývoji této zásadní funkce neprobíhalo zkouškové období tak bohužel nebylo možno využít služeb serveru IS/STAG pro načítání dat. Proto bylo využito již dříve uloženého souboru s odpovědí serveru na dotaz `getTerminyProStudenta`.

Předměty jsou odděleny pomocí záhlaví nadepsaným katedrou a jejich zkratkou, poté následují vypsané termíny předmětu oddělené tenkou čarou. Jednotlivé termíny mají uvedený den, čas a místnost konání, obsazenost, deadline zapsání a odepsání a také případnou poznámku od vyučujícího. U termínu se také případně nachází tlačítko pro zapsání či odepsání termínu, pokud je však student již zapsán na jiném termínu stejného předmětu, žádné tlačítko se nezobrazí.

Termíny pro jednotlivé předměty získané za pomoci servisy `stag-api.service` se roztřídí do mapy na základě kombinace katedry a zkratky předmětu. Následně jsou termíny seřizeny na základě data a času konání a pomocí bindingu a funkce `*ngFor` jsou poté zobrazeny v grafické části aplikace. Při zmáčknutí tlačítka Zapsat nebo tlačítka Odepsat je zavolána příslušná metoda v servise `stag-api.service` a po provedení

příkazu se pomocí komponenty toast, kterou poskytuje framework Ionic, zobrazí hláška zda byla operace úspěšně dokončena.

5.6 Vyhledávání

Při vývoji funkce pro vyhledávání musela být zohledněna i možnost jejího využití nepřihlášeným studentem. Pokud student není přihlášen nemůže pomocí vyhledávání zjistit informace o studentech Jihočeské univerzity, a to z důvodu ochrany jejich citlivých údajů. Proto musí být kontrolováno, zda je student přihlášen. Funkce se skládá ze dvou stránek, první se stará o zjištění, co chce student hledat a druhá provádí samotné hledání.

Stránka vyhledat obsahuje tlačítkové menu pro výběr druhu vyhledávání. Na této stránce probíhá kontrola, zda je student přihlášen, a to podle nastavení singletonu uživatel. Jednotlivá tlačítka pak navigují na stránku hledat a přidávají také informaci o vybraném druhu hledání. Číselná informace o druhu hledání se předává pomocí routeru a interface NavigationExtras, který přidává možnost předání informací při přechodu na jinou stránku.

Na základě této číselné informace se pak na stránce hledat zobrazí správná hlavička a příslušná vyhledávací pole pro daný druh vyhledávání. Pro vyhledávání učitelů a studentů jsou to pole pro jméno a příjmení. Předměty lze hledat podle jejich pracoviště a zkratky či názvu, pro vyhledávání místností se využívá číslo místnosti anebo zkratka budovy. Číslo z předchozí stránky je také použito při výběru dotazu na server IS/STAG. Odpověď ze serveru je pak uložena do příslušného pole ve formě dané třídy: Ucitel, Predmet, Student či Mistnost. Následně je pole zobrazeno pomocí bindingu a `*ngFor`, které poskytuje framework Angular. Na jednotlivé nálezy lze poté kliknout pro bližší informace. Při kliknutí se pomocí servisy `stag-api.service` vyhledají informace o hledaném výrazu a poté se předají stejným způsobem, který byl popsán výše, stránce pro zobrazení příslušných informací. Jediný rozdíl je v předávání místností, kde se místo čísla předává celá třída. Důvodem je že server IS/STAG vrací

podrobné informace při vyhledávání místností, a právě proto by bylo nepotřebné znovu server zatěžovat zbytečným dotazem.

5.6.1 Detail

Pro zobrazení informací o vyhledaném výrazu byly vygenerovány stránky s příslušným jménem: ucitel, student, předmět, místnost. Stránky obsahují komponentu pro zobrazení rozvrhu a informace či seznam předmětů, mezi kterými lze přepínat. Stránka podle předaného JSONu nastaví třídu udržující získaná data a pomocí servisy `stag-api.servise` nastaví list s předměty, které jsou uloženy ve vlastní třídě. Zobrazení je znovu realizováno pomocí bindingu a vypisání předmětů je řešeno pomocí `*ngFor`.

Vypisované informace pro učitele jsou jméno, příjmení, tituly, pracoviště, telefon, e-mail a webová adresa. Pro studenta to jsou také jméno, příjmení a e-mail, na rozdíl od učitele jsou ale přidány informace o osobním čísle, studijním programu, oboru a ročníku. Při zobrazení předmětu zmizí tlačítko pro zobrazení předmětů a zobrazované informace o předmětu jsou stejné jako na webových stránkách `wstag.jcu.cz`, zobrazuje se tedy sylabus předmětu. Místnost zobrazuje svůj rozvrh a bližší informace jako je např. zkratka budovy či patro ve kterém se místnost nachází.

5.7 Plánovač

Další vyvíjenou funkcí aplikace byl plánovač předmětů na budoucí semestry. Tuto možnost server IS/STAG neposkytuje, avšak pro implementaci plánovače bylo využito služeb IS/STAG. Veškerá funkcionalita plánovače je obsažena ve dvou nemodálních stránkách (vytvoření či upravení naplánovaných předmětů a náhled na uložené naplánované předměty) a jednoho modálního překryvného dialogu.

V záhlaví stránky pro plánování předmětů se zobrazují informace o právě plánovaném semestru a v zápatí je umístěno menu pro nástroje plánování. Nacházejí se zde tlačítka pro navigaci mezi plánovanými semestry, uložení vybraných předmětů, tlačítko pro zobrazení již přidávaných předmětů a možnost přidání předmětu, který není

v zobrazovaném seznamu. V těle stránky se pak nachází seznam předmětů, které jsou povinné pro obor přihlášeného studenta. Mezi informace zobrazované o předmětu patří katedra a zkratka předmětu, název a statut předmětu vůči oboru studenta. U některých předmětů se také zobrazí doporučený ročník, podmiňující předměty, vylučující předměty, nebo předměty, které předmět podmiňuje. Student může předmět zapsat kliknutím na checkbox, jenž se nachází vedle zkratky předmětu.

K získávání zobrazovaných předmětů ze serveru IS/STAG lze využít službu `getPredmetyByObor`, avšak bylo zjištěno, že tato služba nevrací informace o podmiňujících, vylučujících a podmiňovaných předmětech. Z tohoto důvodu musela být využita služba `getPredmetyByOborFullInfo`, která poskytuje detailnější informace. Pokud student otevře stránku a má již uložené předměty tak se v seznamu automaticky zaškrtnou. Seznam předmětů je filtrovaný podle letního či zimního semestru a podle seznamu již studentem vybraných předmětů, ty se zobrazí pouze v semestru a ročníku, ve kterém byly vybrány.

Předměty vybrané studentem se po kontrole uloží do mapy, kde je jako klíč použito číslo semestru. Zmíněná kontrola zajišťuje, aby vybíraný předmět splňoval kritéria, pokud nějaká má. Pokud má předmět nějaký podmiňující předmět zjistí se, zda jej obsahuje seznam vybraných předmětů. Pokud předmět není zapsán, tak je to pomocí komponenty frameworku Ionic `ion-alert` studentovi oznámeno. Obdobným způsobem je řešena kontrola vylučujících předmětů a také se kontroluje, zda již nebyl předmět vybrán. Pokud se předmět nachází v seznamu vybraných předmětů, tak je ze seznamu odeberán.

Jak již bylo několikrát zmíněno student může přidat předmět, který se nenachází mezi povinnými předměty pro jeho obor. Po kliknutí na tlačítko Přidat jiný předmět se vyvolá překrytí stránky a zobrazí se dialog pro vyhledávání předmětů. Student může kliknutím předmět vybrat a tím se překrytí zruší nebo klikne na tlačítko Zavřít.

Dialog je realizován pomocí komponenty `ion-modal`, kterou poskytuje Ionic framework. Modal je realizován pomocí stránky `pridat-predmet`, jež obsahuje

vyhledávací pole pro vyhledávání předmětů. Vyhledané předměty jsou opět uloženy v listu, který je za pomoci bindingu a `*ngFor` prezentován studentovi. Při vybrání předmětu se pomocí servisy `stag-api.service` načtou informace o daném předmětu a při zavření se předají stránce, která modal vyvolala, tedy stránce `planovac`. Zde projdou již popsanou kontrolou a pokud jí předmět splňuje, tak se přidá do seznamu vybraných předmětů. Ve spodním menu se také nachází tlačítko pro zjištění vybraných předmětů. Tlačítko zobrazí komponentu `ion-alert` pro jednoduché zobrazení již vybraných předmětů a semestrů, na které jsou předměty naplánovány.

5.7.1 Databáze

Pro možnost zobrazení plánovače na jiných mobilních zařízeních, než kde byly předměty vybrány, bylo využito provázání s dokumentovou databází `restDB`⁸, která je založena na principu dotazovacího jazyka z `MongoDB`. Jelikož aplikace psaná za pomoci frameworku `Ionic` nemá vlastní back-end bylo potřeba, aby vybraná databáze poskytovala vlastní API, proto byla vybrána databáze `restdb.io`. Seznam předmětů se do databáze ukládá jako soubor ve formě `JSON` pro lepší optimalizaci v aplikaci. Pro komunikaci s databází byla vytvořena servisa `db-api.service`, pomocí které se ukládají a načítají data. K ukládání dat musela být také vytvořena nová třída ve stránce `planovac`.

Při ukládání se nejdříve musí použít metoda pro načtení dat z databáze, a to ze dvou důvodů. Prvním je zjištění, zda má student již nějaké předměty uložené. Pokud nejsou uloženy, vykoná se přidání záznamu do databáze pomocí metody `POST`. Pokud má student již uloženy nějaké předměty, tak je potřeba daný záznam přepsat. To je realizováno dotazem metodou `PUT` s blíže specifikovanou `URL` adresou. Za adresu serveru se přidá zpětné lomítko s `id` dříve získaného záznamu. Při načítání dat se pro jejich vyhledání použije parametru, do kterého se zadá osobní číslo studenta. Vrácená data se poté převedou do formátu `JSON` a předají se jako návratová hodnota metody.

⁸ <https://restdb.io/>

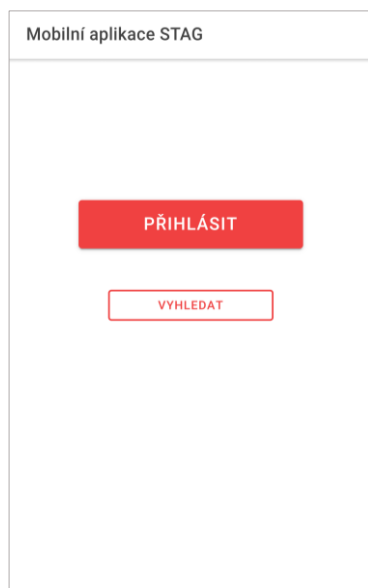
5.7.2 Zobrazení naplánovaného rozvrhu

Student si může uložené předměty zobrazit pomocí stránky planovac-zapsane, na kterou se dostane pomocí podpoložky menu plánování → zobrazit. Na stránce může student zjistit, které předměty si na dané semestry naplánoval. U předmětů je uvedena katedra, zkratka předmětu, název a statut předmětu. Případně je zde zobrazen i doporučený ročník, podmiňující, vylučující a podmiňované předměty.

```
[
  [
    1,
    [
      {
        "doporuceni": false,
        "jePridan": true,
        "podminujiciPredmetyZobraz": true,
        "vylucujiciPredmetyZobraz": true,
        "podminujePredmetyZobraz": true,
        "katedra": "FBI",
        "zkratka": "007",
        "nazev": "Práce s informačními zdroji",
        "kreditu": 0,
        "statut": "A",
        "doporucenyRocnik": "1 ZS",
        "vyukaLS": "N",
        "vyukaZS": "A"
      }
    ],
    2,
    [
      {
        "doporuceni": false,
        "jePridan": true,
        "podminujiciPredmetyZobraz": true,
        "vylucujiciPredmetyZobraz": true,
        "podminujePredmetyZobraz": true,
        "katedra": "UAI",
        "zkratka": "648",
        "nazev": "Právní minimum pro informatiky",
        "kreditu": 1,
        "statut": "A",
        "doporucenyRocnik": "1 ZS",
        "vyukaLS": "N",
        "vyukaZS": "A"
      }
    ]
  ]
]
```

Obrázek 9 - Příklad ukládaného JSONu

Z databáze restdb.io je pomocí studentova osobního čísla načten soubor ve formátu JSON s informacemi o uložených předmětech. Ten je uložen do mapy na základě své struktury, která je vidět na Obrázek 9. Mapa predmety se poté přeuloží do mapy predmetyZobraz, jež je obohacena o číslo ročníku a zkratku semestru. Poté je pomocí bindingu a *ngFor prezentována studentovi.



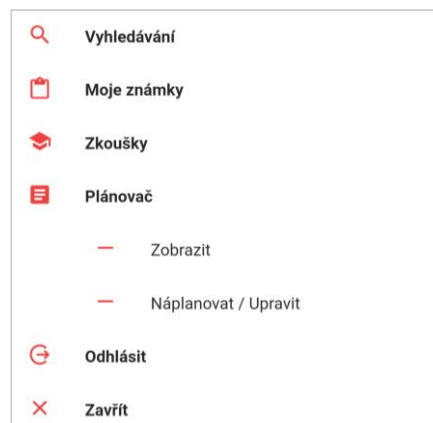
Obrázek 11 - Domovská stránka nepřihlášeného uživatele



Obrázek 13 - Domovská stránka přihlášeného uživatele



Obrázek 12 - Komponenta rozvrh



Obrázek 10 - Menu aplikace

Vyhledávání učitelů

Jméno	Příjmení
	Prokysek

HLEDAT

PhDr. Miloš Prokýšek

INFO **HOME** **MENU**

Obrázek 14 - Vyhledávání vyučujících

Moje zkoušky

KPE/SPSSN

21.1.2020 09:00:00 - 10:00:00

Obsazenost: 2/18 **Odepsat**

Místnost: PD - D224

Deadline zapsání na termín: 20.1.2020 9:00
 Deadline odepsání termínu: 20.1.2020 9:00
 Opravný termín: Ne
 Poznámka:

OJZ/310

23.1.2020 13:00:00 - 15:00:00

Obsazenost: 1/8 **Zapsat**

Místnost: BA - OJ

Deadline zapsání na termín: 22.1.2020 13:00

INFO **HOME** **MENU**

Obrázek 15 - Stránka obsahující správu zkouškových termínů

Předměty podle oboru na 1. ročník ZS

<input type="checkbox"/>	FBI/003	Statut předmětu	A
Úvod do studia a života na VŠ			
Počet kreditů: 0			
Doporučený ročník: 1 ZS			
<input checked="" type="checkbox"/>	FBI/007	Statut předmětu	A
Práce s informačními zdroji			
Počet kreditů: 0			
Doporučený ročník: 1 ZS			
<input checked="" type="checkbox"/>	UAI/648	Statut předmětu	A
Právní minimum pro informatiky			
Počet kreditů: 1			
Doporučený ročník: 1 ZS			
<input type="checkbox"/>	UAI/686	Statut předmětu	A

PŘIDAT JINÝ PŘEDMĚT

← **→** **ULOŽIT** **VYBRANÉ**

INFO **HOME** **MENU**

Obrázek 16 - Plánovač předmětů

Náplňované předměty

1. ročník ZS

FBI/007	Statut předmětu	A
Práce s informačními zdroji		
Počet kreditů: 0		
Doporučený ročník: 1 ZS		
UAI/648	Statut předmětu	A
Právní minimum pro informatiky		
Počet kreditů: 1		
Doporučený ročník: 1 ZS		
UAI/761	Statut předmětu	A
Úvod do SW aplikací		
Počet kreditů: 5		
Doporučený ročník: 1 ZS		
Podmiňuje předměty: UAI/695, UAI/695U		
UAI/795	Statut předmětu	C
Supercomputing		
Počet kreditů: 4		

INFO **HOME** **MENU**

Obrázek 17 - Zobrazení naplánovaných předmětů

5.8 Generování aplikace

Posledním krokem při psaní aplikace je její vygenerování na dané platformy pomocí příkazové řádky. Pro vygenerování aplikace bylo využito Apache Cordova, která je pro generování optimalizovaná. Před samotným generováním byl použit příkaz:

```
ionic cordova plugin rm cordova-plugin-console
```

Tento příkaz odebere z aplikace z aplikace konzoli, která se používá pro debugování.

Samotné generování pro platformy Android a iOS se liší. Aplikace pro platformu iOS musí být generována za pomoci Xcode, jenž běží pouze na operační systému MacOS. Jelikož se během práce nenaskytla možnost dostat se ke stroji, na kterém běží tento operační systém, nebylo možno aplikaci vygenerovat pro použití na iOS.

Při generování pro platformu Android je potřeba nejdříve založit keystore pro danou aplikaci, aby se zajistila její bezpečnost. Tento heslem chráněný keystore byl generován pomocí nástroje keytool a poté byl využíván pro podepsání aplikace. Později byl tento vygenerovaný keystore také používán pro aktualizování mobilní aplikace. Založení keystore se provedlo pomocí příkazu:

```
keytool -genkey -v -keystore my-release-key.keystore  
-alias kralova -keyalg RSA -keysize 2048 -validity 10000
```

Poté co byl vytvořen keystore, se mohlo přistoupit ke generování samotné aplikace. To bylo provedeno pomocí následujících příkazů:

```
ionic cordova build --prod --release android  
-----  
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1  
-keystore my-release-key.keystore app-release-  
unsigned.apk kralova  
-----  
zipalign -v 4 app-release-unsigned.apk  
Mobilni_aplikace_STAG.apk
```


Prvním příkazem byla nejdříve vygenerována nepodepsaná aplikace pomocí souboru pro generování config.xml, který obsahuje popis aplikace a lze ho upravit. Výsledný apk soubor se vygeneroval do složky platforms/android/build/outputs/apk. Následně bylo možné vygenerovanou aplikaci podepsat pomocí uloženého keystore a tím zajistit její validitu. Pro podepsání aplikace se využívá nástroje jarsigner, který poskytuje Android SDK a zajišťuje jej druhý příkaz. Když byla aplikace vygenerovaná a podepsaná byl použit nástroj Android studia zipalign, jenž optimalizuje mobilní aplikaci (22).

Po vygenerování je aplikace připravena pro instalaci do mobilního zařízení. Celý výsledný projekt byl nahrán pod MIT licenci na webový server GitLab.com⁹. Na této adrese se také nachází vygenerovaný .apk soubor pro instalaci aplikace na platformu Android a zároveň je zde dostupná i uživatelská dokumentace aplikace.

⁹ Dostupné na <https://gitlab.com/kuriarare.anko/mobilni-aplikace-stag>

6. Diskuze

Pro komunikaci se serverem IS/STAG lze použít rozhraní SOAP nebo REST API. Bylo využito REST API, protože je považováno za modernější přístup. Pro vývoj aplikace byl zvolen přístup hybridního programování, jelikož aplikace především zobrazuje data, a tak není potřeba využívat speciálních funkcí jednotlivých platforem. Z důvodu osobní zkušenosti byl jako front-end framework využit Angular.

Z dotazníkového šetření v rámci stanovení funkčních požadavků vyplynulo, že studenti silně preferují funkci vyhledávání. Tato funkce není dostupná v žádné existující produkční aplikaci v aplikačním obchodě (existuje pouze jako součást jediné aplikace, která je momentálně v testovací verzi). Její implementace do aplikace je silnou stránkou celého řešení. Další z hlavních funkcí vyvíjené aplikace, byla funkcionalita plánování předmětů na budoucí semestry, která v žádné jiné existující aplikaci není.

Pro podporu funkce plánovače v aplikaci bylo nutné využít dokumentový databázový server. V současné verzi je tato podpora řešena formou SaaS. Pro budoucí rozvoj aplikace by bylo vhodné přenést takto ukládaná data na vlastní server univerzity a případně rozšířit jeho možnosti přímým napojením na IS/STAG, nebo tuto funkci implementovat přímo do informačního systému.

Jelikož je aplikace dostupná pouze pro studenty Jihočeské univerzity, bylo by vhodné ji rozšířit o možnost využití na více školách. Dalším možným rozšířením je mapa, která by zobrazovala budovy školy. V průběhu vývoje aplikace přibylo na webovém portálu IS/STAG oznamování zapsání známky. Tato funkce zatím není webovými službami IS/STAG podporována, ale jelikož je rozhraní REST stále rozšiřováno je možné, že tato funkce bude zavedena. Další funkcí, kterou webové služby IS/STAG neposkytují je předzápis a stejně jako u funkce oznamování je možné její přidání IS/STAGem v budoucnu.

7. Závěr

Tato bakalářská práce se zabývala vývojem mobilní aplikace pro zobrazování informací ze serveru IS/STAG. Pro zjištění požadovaných funkcí byl nejzásadnější názor studentů, který byl zjišťován pomocí rozeslaného dotazníku. Z funkčních požadavků definovaných na základě dotazníkového šetření nebyly do aplikace některé funkce implementovány, a to především z technických důvodů, jelikož serverové rozhraní IS/STAG tyto funkce (prozatím) nepodporuje (např. předzápis a oznámení zapsání známky).

Zásadní funkce, tedy funkce, které byly nejvíce požadované, byly ve výsledné aplikaci úspěšně implementovány. Jedná se o funkce: zobrazení rozvrhu, správa zkouškových termínů, zobrazení předmětů a známek studenta a vyhledávání.

Velkým přínosem této aplikace je možnost naplánovat si předměty na budoucí semestry. Funkce nabízí především předměty, které jsou povinné pro obor přihlášeného studenta, ale také umožňuje přidání jiných předmětů. Naplánovaný rozvrh se ukládá ve formátu JSON do NoSQL databáze na bázi MongoDB, poskytované službou restdb.io.

Využití Ionic frameworku se ukázalo jako správně zvolené, především díky využití komponent, které pomohly s tvorbou grafické části aplikace. Propojení grafické části s částí logickou bylo usnadněno použitím funkcionality two-way bindingu z frameworku Angular, avšak našli se i případy, kde muselo být zobrazování řešeno jiným způsobem. Průběžné i závěrečné testování aplikace proběhlo na základě scénářů podle uživatelské dokumentace.

Vzhledem k výše uvedenému lze cíle práce považovat za splněné, nicméně funkcionality REST API rozhraní IS/STAG (i samotného informačního systému) se nadále rozvíjí, což poskytuje prostor pro další rozvoj i pro mobilní aplikace s ním spojené.

8. Seznam použité literatury

1. CENTRUM INFORMATIZACE A VÝPOČETNÍ TECHNIKY. Zájemci: Základní informace. *IS/STAG - Informační systém studijní agendy* [online]. Plzeň: Západočeská univerzita v Plzni. Poslední změna 07.04.2020. [cit. 2020-05-06]. Dostupné z: <https://is-stag.zcu.cz/zajemci/>
2. CENTRUM INFORMATIZACE A VÝPOČETNÍ TECHNIKY. Přístupy k systému: Další klienti. *IS/STAG - Informační systém studijní agendy* [online]. Plzeň: Západočeská univerzita v Plzni. Poslední změna 25.06.2019. [cit. 2020-05-06]. Dostupné z: https://is-stag.zcu.cz/zajemci/pristupy/dalsi_klienti.html
3. CENTRUM INFORMATIZACE A VÝPOČETNÍ TECHNIKY. Přístupy k systému: Windows klient. *IS/STAG - Informační systém studijní agendy* [online]. Plzeň: Západočeská univerzita v Plzni. Poslední změna 24.05.2017. [cit. 2020-05-07]. Dostupné z: https://is-stag.zcu.cz/zajemci/pristupy/tlusty_klient.html
4. CENTRUM INFORMATIZACE A VÝPOČETNÍ TECHNIKY. Přístupy k systému: Portálové rozhraní. *IS/STAG - Informační systém studijní agendy* [online]. Plzeň: Západočeská univerzita v Plzni. Poslední změna 24.05.2017. [cit. 2020-05-07]. Dostupné z: <https://is-stag.zcu.cz/zajemci/pristupy/portal.html>
5. CENTRUM INFORMATIZACE A VÝPOČETNÍ TECHNIKY. Přístupy k systému: Webové služby. *IS/STAG - Informační systém studijní agendy* [online]. Plzeň: Západočeská univerzita v Plzni. Poslední změna 24.05.2017. [cit. 2020-05-07]. Dostupné z: <https://is-stag.zcu.cz/zajemci/pristupy/webservices.html>
6. CENTRUM INFORMATIZACE A VÝPOČETNÍ TECHNIKY. Portál IS/STAG: Referenční příručka. *Referenční příručka IS/STAG* [online]. Plzeň: Západočeská univerzita v Plzni, ©2010-2014 [cit. 2020-05-07]. Dostupné z: https://is-stag.zcu.cz/NapovedyBuild/PortalNapovedy/build/pdf-zcu/Referencni_prirucka_IS-STAG.pdf

7. CENTRUM INFORMATIZACE A VÝPOČETNÍ TECHNIKY. Webové služby nad IS/STAG: Referenční příručka. *Referenční příručka IS/STAG* [online]. Plzeň: Západočeská univerzita v Plzni, ©2010-2014 [cit. 2020-05-06]. Dostupné z: <https://is-stag.zcu.cz/NapovedyBuild/PortalNapovedy/build/pdf-zcu/Web-Services-IS-STAG.pdf>
8. UNIVERZITA HRADEC KRÁLOVÉ. Aktuální seznam služeb: REST. *IS/STAG - Webové služby* [online]. Hradec Králové: Univerzita Hradec Králové [cit. 2020-05-06]. Dostupné z: https://stag-demo.uhk.cz/ws/web?pp_locale=cs&selectedTyp=REST&pp_reqType=render&pp_page=serviceList
9. DRIFTY CO. *Hybrid vs. Native* [online]. Shorewood: Drifty Co., 2018 [cit. 2020-05-08]. Dostupné z: https://cdn2.hubspot.net/hubfs/3776657/Ionic-HybridNative_eBook_Jan_2018_v8.pdf
10. SONOO, Jaiswal. Ionic History. *JavaTpoint* [online]. JavaTpoint, c2011-2018 [cit. 2020-05-08]. Dostupné z: <https://www.javatpoint.com/ionic-history>
11. DRIFTY CO. Guide: Overview. *Ionic Framework: Ionic Documentation* [online]. Shorewood: Drifty Co. Poslední změna 02.04.2020. [cit. 2020-05-08]. Dostupné z: <https://ionicframework.com/docs>
12. LYNCH, Max. Announcing Capacitor 1.0. In: *Ionic Framework: The Ionic Blog* [online] [online]. Shorewood: Drifty Co. Poslední změna 2.4.2020. [cit. 2020-05-08]. Dostupné z: <https://ionicframework.com/blog/announcing-capacitor-1-0/>
13. GOOGLE. Displaying data in views. *AngularJS: Docs* [online]. Mountain View: Google, ©2010-2020 [cit. 2020-05-08]. Dostupné z: <https://angular.io/guide/displaying-data>

14. MROZEK, Jakub. Začínáme s AngularJS. *Zdroják, o tvorbě webových stránek a aplikací* [online]. Devel.cz, 30.11.2012 [cit. 2020-05-08]. Dostupné z: <https://www.zdrojak.cz/clanky/zaciname-s-angularjs/>
15. THE APACHE SOFTWARE FOUNDATION. Platform Support. *Apache Cordova: Documentation* [online]. Forest Hill: The Apache Software Foundation, ©2015 [cit. 2020-05-08]. Dostupné z: <https://cordova.apache.org/docs/en/latest/guide/support/index.html>
16. GRIFFITH, Chris. What is Apache Cordova? In: *Ionic Framework* [online]. Shorewood: Drifty Co., ©2020 [cit. 2020-05-08]. Dostupné z: <https://ionicframework.com/resources/articles/what-is-apache-cordova>
17. DRIFTY CO. Guide: Previewing. *Ionic Framework: Ionic Documentation* [online]. Shorewood: Drifty Co. Poslední změna 02.04.2020. [cit. 2020-05-08]. Dostupné z: <https://ionicframework.com/docs/developing/previewing>
18. DRIFTY CO. Native: HTTP. *Ionic Framework: Ionic Documentation* [online]. Shorewood: Drifty Co. [cit. 2020-05-08]. Dostupné z: <https://ionicframework.com/docs/native/http>
19. THE APACHE SOFTWARE FOUNDATION. Inappbrowser. *Apache Cordova: Documentation* [online]. Forest Hill: The Apache Software Foundation, ©2015 [cit. 2020-05-08]. Dostupné z: <https://cordova.apache.org/docs/en/9.x/reference/cordova-plugin-inappbrowser/>
20. GOOGLE. API Reference: DomSanitizer. *AngularJS: Docs* [online]. Mountain View: Google, ©2010-2020 [cit. 2020-05-08]. Dostupné z: <https://angular.io/api/platform-browser/DomSanitizer>
21. GOOGLE. API Reference: NgForOf. *AngularJS: Docs* [online]. Mountain View: Google, ©2010-2020 [cit. 2020-05-08]. Dostupné z: <https://angular.io/api/common/NgForOf>

22. DRIFTY CO. Guide: Android Play Store Deployment. *Ionic Framework: Ionic Documentation* [online]. Shorewood: Drifty Co. Poslední změna 02.04.2020. [cit. 2020-05-08]. Dostupné z: <https://ionicframework.com/docs/deployment/play-store>

9. Seznam obrázků

Obrázek 1 - Logo IS/STAG.....	12
Obrázek 2 - Záložka Prohlížení.....	15
Obrázek 3 - Rozbor URL adresy při dotazu na webový server IS/STAG.....	16
Obrázek 4 - Odpověď na dotaz ve formátu JSON.....	17
Obrázek 5 - Logo Ionic Frameworku	21
Obrázek 6 - Struktura aplikace	23
Obrázek 7 - Diagram přihlašování	26
Obrázek 8 - Volání komponenty s inputy.....	29
Obrázek 9 - Příklad ukládaného JSONu.....	36
Obrázek 12 - Komponenta rozvrh	37
Obrázek 11 - Domovská stránka přihlášeného uživatele.....	37
Obrázek 10 - Domovská stránka nepřihlášeného uživatele.....	37
Obrázek 13 - Menu aplikace.....	37
Obrázek 14 - Vyhledávání vyučujících	38
Obrázek 15 - Stránka obsahující správu zkouškových termínů	38
Obrázek 16 - Plánovač předmětů	38
Obrázek 17 - Zobrazení naplánovaných předmětů.....	38

10. Seznam grafů

Graf 1- Nejdůležitější funkce dle odpovědí studentů	6
Graf 2- Výsledky dotazníku na funkce mobilní aplikace	11

11. Seznam tabulek

Tabulka 1 - Zúčastněné fakulty a studenti.....	5
--	---

12. Přílohy

