



Zadání bakalářské práce

| | |
|-----------------------------|--|
| Název: | Educhild – webová aplikace |
| Student: | Jan Stejskal |
| Vedoucí: | Ing. Jiří Hunka |
| Studijní program: | Informatika |
| Obor / specializace: | Webové a softwarové inženýrství, zaměření Softwarové inženýrství |
| Katedra: | Katedra softwarového inženýrství |
| Platnost zadání: | do konce letního semestru 2021/2022 |

Pokyny pro vypracování

Cílem této práce je realizace webové aplikace rozšiřující možnosti mobilní aplikace Educhild, která slouží jako pomůcka pro rodiče k motivaci plnění edukačních úkolů a limitaci jiných aktivit dítěte na mobilním zařízení s OS android. Webová aplikace by měla minimálně umožnit správu dětských uživatelů a pohodlnou tvorbu a sdílení kvízů.

Postupujte v těchto krocích:

- Analyzujte potřeby budoucích uživatelů webové aplikace Educhild
- Výslednou analýzu doplňte o rešerši stávajících (konkurenčních) řešení.
- Na základě analýzy proveďte řádný softwarový návrh včetně volby technologií.
- Implementujte prototyp webové aplikace, který následně podrobte vhodným testům.
- Na základě testů navrhnete úpravy a realizujete výslednou aplikaci
- Řádně vše zhodnoťte a navrhnete možná budoucí vylepšení



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Bakalářská práce

Educhild – webová aplikace

Jan Stejskal

Katedra softwarového inženýrství

Vedoucí práce: Ing. Jiří Hunka

12. května 2021

Poděkování

Rád bych na těchto řádcích poděkoval vedoucímu práce Ing. Jiřímu Hunkovi za cenné praktické rady a nápady, které mi poskytl v době tvorby této práce a také za důkladnou zpětnou vazbu, především ve fázích analýzy a návrhu uživatelského rozhraní. Stejně tak děkuji kolegyni Barboře Kyselové, která se značnou mírou podílela na výsledném systému a bez její spolupráce by systém vznikl v dosti omezené podobě.

Děkuji také zbylým členům týmu EduChild, konkrétně Petru Šímovi, Miloši Popovičovi, Marku Trzaskalíkovi, Tereze Langové, Danielu Matouškovi a Tomáši Hankovi. Děkuji za objemnou zpětnou vazbu při analýze a návrhu a za pomoc při řešení problémů.

Velký dík patří také všem čtyřem osobám, které mi pomáhali otestovat prototyp. Před testováním bylo dohodnuto, že zůstanou v anonymitě.

Děkuji mému bratrovi JUDr. Jakubovi Stejskalovi za jazykovou a stylistickou korekturu. Na závěr bych rád poděkoval rodině a přátelům za jejich podporu v době mého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principu při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 12. května 2021

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2021 Jan Stejskal. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Stejskal, Jan. *Educhild – webová aplikace*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Abstrakt

Práce se soustřeďuje na kompletní proces vývoje softwarového řešení, počínající analýzou a návrhem řešení s postupným přechodem k implementaci a k testování. Webová aplikace EduChild vznikla v návaznosti na potřeby sourozenecké aplikace na systému Android stejného jména. Mezi tyto potřeby zařadíme přehlednější a uživatelsky přívětivější činnosti jako je například správa uživatelských účtů či tvorba kvizových formulářů.

Naprostá většina analýzy a návrhu spatřila světlo světa jako součást řešení semestrální práce předmětu BI-SP2 vyučovaném na FIT ČVUT v Praze. Pro implementaci je využíván Javascriptový framework Vue.js, jeho nadstavba pro jednodušší tvorbu universal webových aplikací Nuxt.js a grafický framework Vuetify. Aplikace získává data pomocí REST API, které je součástí jiné bakalářské práce.

Klíčová slova Vue.js, Nuxt.js, Vuetify, Javascript, frontend, kvízy, správa uživatelských účtů

Abstract

This thesis focuses on complete process of software development lifecycle starting with analysis and draft with gradual transition to implementation and testing. Web application Educhild originated following the needs of its sibling application of the same name for Android. Included among those needs are the need for clear and more user friendly experience when dealing with actions e.g. user account management or creation of quiz forms.

Overwhelming majority of analysis and draft saw the light of the world as a part of solution for semestral work in subject BI-SP2 from FIT CTU in Prague. Javascript framework Vue.js, higher-level framework Nuxt.js which is built on top of Vue.js and makes developing universal web apps easier and graphical framework Vuetify. Application gets its data from REST API which is a part of another bachelor thesis.

Keywords Vue.js, Nuxt.js, Vuetify, Javascript, frontend, quizzes, user account management

Obsah

| | |
|---|----------|
| Úvod | 1 |
| 1 Analýza | 3 |
| 1.1 Mobilní aplikace EduChild | 3 |
| 1.2 Sběr podkladů | 3 |
| 1.3 Funkční požadavky | 4 |
| 1.3.1 Rodič | 4 |
| 1.3.2 Tvůrce kvízů | 6 |
| 1.4 Nefunkční požadavky | 6 |
| 1.5 Případy užití a scénáře | 7 |
| 1.5.1 Sledování dětské aktivity | 8 |
| 1.5.2 Správa uživatelských kvízů | 8 |
| 1.5.3 Tvorba kvízů | 9 |
| 1.5.4 Sdílení kvízů | 10 |
| 1.5.5 Obchod | 11 |
| 1.5.6 Správa aplikací | 12 |
| 1.5.7 Správa uživatelského účtu | 12 |
| 1.5.8 Správa dětských účtů | 13 |
| 1.5.9 Správa dětského času | 14 |
| 1.6 Analýza konkurenčních řešení | 15 |
| 1.6.1 Analýza aplikací zabývajících se tvorbou kvízů | 15 |
| 1.6.1.1 Google Forms | 15 |
| 1.6.1.2 Microsoft Forms | 16 |
| 1.6.1.3 Survio | 17 |
| 1.6.1.4 Fyrebox | 18 |
| 1.6.1.5 Kahoot! | 20 |
| 1.6.2 Analýza aplikací zabývajících se správou svěřených účtů | 20 |
| 1.6.2.1 Microsoft Family Safety | 20 |
| 1.6.2.2 Google Families | 21 |

| | | |
|----------|---|-----------|
| 1.6.3 | Shrnutí | 22 |
| 2 | Návrh | 23 |
| 2.1 | Task list | 23 |
| 2.1.1 | Uživatelé | 23 |
| 2.1.2 | Přihlášení uživatelé | 24 |
| 2.1.3 | Role Rodič | 25 |
| 2.2 | Návrh uživatelského rozhraní (Li-Fi prototyp) | 26 |
| 2.2.1 | Volba nástroje | 26 |
| 2.2.2 | Wireframes | 26 |
| 2.3 | Architektura | 28 |
| 2.3.1 | Single Page Application – SPA | 28 |
| 2.3.1.1 | Rekapitulace | 31 |
| 2.3.2 | Universal App | 32 |
| 2.3.2.1 | Rekapitulace | 34 |
| 2.3.3 | Výběr architektury | 34 |
| 2.4 | Volba technologií | 34 |
| 2.5 | Mockup (Hi-Fi prototyp) | 35 |
| 2.6 | Návrh API | 36 |
| 3 | Implementace | 37 |
| 3.1 | Nástroje | 37 |
| 3.2 | Tvorba Nuxt projektu a jeho struktura | 37 |
| 3.3 | Nastavení Firebase Auth | 39 |
| 3.4 | Axios a API | 40 |
| 3.5 | Překlady | 42 |
| 3.6 | GitLab CI/CD a Sentry | 43 |
| 3.7 | Problémy při implementaci | 44 |
| 3.7.1 | Persistence přihlášených uživatelů a přesměrování | 44 |
| 3.8 | Dokončená část | 45 |
| 4 | Testování | 47 |
| 4.1 | Co je uživatelské testování | 47 |
| 4.2 | Způsob testování | 48 |
| 4.3 | Testeři | 49 |
| 4.3.1 | První tester | 49 |
| 4.3.2 | Druhý tester | 49 |
| 4.3.3 | Třetí tester | 49 |
| 4.3.4 | Čtvrtý tester | 50 |
| 4.3.5 | Nielsenova křivka | 50 |
| 4.4 | Shrnutí poznatků z testování | 51 |
| 4.4.1 | Provedené změny a ukázky výsledného systému | 54 |
| 5 | Budoucí vylepšení | 57 |

| | |
|--|-----------|
| Závěr | 59 |
| Literatura | 61 |
| A Seznam použitých zkratk | 65 |
| B Případy užití a scénáře | 67 |
| B.1 Sledování dětské aktivity | 67 |
| B.2 Správa uživatelových kvízů | 68 |
| B.3 Tvorba kvízů | 69 |
| B.4 Sdílení kvízů | 70 |
| B.5 Obchod | 71 |
| B.6 Správa aplikací | 72 |
| B.7 Správa uživatelského účtu | 73 |
| B.8 Správa dětských účtů | 74 |
| B.9 Správa dětského času | 75 |
| C Wireframes | 77 |
| D Testovací scénáře | 89 |
| E Dotazník po testování | 93 |
| F Ukázky systému | 95 |
| G Obsah příloženého CD | 99 |

Seznam obrázků

| | | |
|-----|---|----|
| 1.1 | Vztah rolí | 9 |
| 1.2 | Ukázka nekonzistentního rozvržení Google Forms, kdy změna názvu formuláře probíhá vlevo nahoře místo očekávaného středu . . . | 17 |
| 1.3 | Ukázka špatně nastavené komponenty výběru v nástroji Survio . . | 19 |
| 2.1 | Wireframe obrazovky Moje kvízy | 27 |
| 2.2 | Jeden z prvních návrhů obrazovky Tvorba kvízu | 28 |
| 2.3 | Současný návrh obrazovky Tvorba kvízu | 29 |
| 2.4 | Návrh obrazovky Děti | 30 |
| 2.5 | Proces startu universal aplikace | 33 |
| 3.1 | Ukázka založení Nuxt aplikace | 38 |
| 3.2 | Vuex pattern | 40 |
| 3.3 | Ukázka konfigurace Firebase | 41 |
| 3.4 | Ukázka kódu metody onRequest | 42 |
| 3.5 | Ukázka základní a specializované servisní třídy | 43 |
| 3.6 | Ukázka kódu pro inject servisních tříd do context | 44 |
| 4.1 | Testovací scénář pro registraci | 48 |
| 4.2 | Nielsenova křivka | 51 |
| 4.3 | Výřez navigační listy v době testování | 52 |
| 4.4 | Detail kvízu s upraveným tlačítkem stažení | 55 |
| 4.5 | Stránka Moje Kvízy | 55 |
| 5.1 | Ukázka jednoduchého Unit testu pro komponentu zobrazení volného času | 58 |
| 5.2 | Graf stráveného času vývojem prototypu část 1 | 60 |
| 5.3 | Graf stráveného času vývojem prototypu část 2 | 60 |
| C.1 | Nastavení uživatele | 77 |

| | | |
|------|---|----|
| C.2 | Obrazovka správy dětí | 78 |
| C.3 | Panel tvorby rozvrhu | 79 |
| C.4 | Panel kontroly vyplněného kvízu | 80 |
| C.5 | Panel editace dětského účtu | 81 |
| C.6 | Panel přiřazení dítěte ke kvízu | 82 |
| C.7 | Obrazovka obchodu | 83 |
| C.8 | Detail kvízu v obchodě | 84 |
| C.9 | Detail kvízu před přiřazením | 85 |
| C.10 | Nahlášení kvízu | 86 |
| C.11 | Obrazovka seznamu uživatelových kvízů | 87 |
| C.12 | Tvorba kvízu - informace o kvízu | 87 |
| C.13 | Tvorba kvízu - vyplněné informace o kvízu | 88 |
| C.14 | Tvorba kvízu - otázka | 88 |
| F.1 | Detail kvízu s upraveným tlačítkem stažení | 95 |
| F.2 | Stránka Moje Kvízy | 96 |
| F.3 | Rozvrhy a nastavení volného času | 96 |
| F.4 | Přiřazení kvízů ve Správě dětí | 97 |
| F.5 | Stránka Nastavení | 97 |
| F.6 | Tvorba kvízu – vyplnění informací a výběr typu otázky | 98 |
| F.7 | Tvorba kvízu – vyplnění otázky | 98 |

Seznam tabulek

| | | |
|-----|--|----|
| 1.1 | Hodnocení konkurenčních aplikací | 22 |
|-----|--|----|

Úvod

S příchodem chytrých mobilních telefonů se změnil způsob, jakým děti tráví svůj volný čas. Asi každý z nás má zkušenost s dětmi, které tráví příliš mnoho času se svým mobilem. A tak vznikl, jako součást předmětu BI-SP1, nápad vytvořit aplikaci na mobilní telefony, která by zablokovala sociální sítě, hry a jiné volnočasové aplikace a odblokovala je až poté, co dítě splní vědomostní kvíz. S tím se bohužel vážou jistá úskalí, přičemž tím největším z nich je velikost. Přestože se v posledních letech změnila uhlopříčka mobilních zařízení do dříve nevídaných rozměrů, stále díky velikosti svých obrazovek nenabízí takové možnosti pro čisté a uživatelsky přívětivé aplikace jako notebook nebo stolní počítač. Mobil je skvělý na zobrazení věcí. V čem už bohužel moc neexceluje je schopnost na něm například tvořit složité formuláře nebo spravovat uživatelské účty. A obě tyto akce jsou stěžejním bodem naší aplikace.

Proto došlo k rozhodnutí, že mobilní aplikace se bude zaměřovat na prezentaci kvízů svým dětským uživatelům a blokování jejich volnočasových aplikací. Na tvorbu kvízových formulářů, správu dětských účtů nebo získávání kvízů se zaměří „sesterská“ webová aplikace, která může uživatelům nabídnout dostatek prostoru na pohodlné užívání. Na základě těchto rozhodnutí vzniklo zadání této bakalářské práce.

V této bakalářské práci se budu zabývat pouze klientskou částí webové aplikace, neboť serverová část bude sdílená s mobilní aplikací a je součástí bakalářských prací kolegů z BI-SP2. Důvodů, proč jsem si vybral toto téma je hned několik. Mám sice zkušenosti z vývoje frontendu bankovních aplikací, ale jedná se spíše o podporu a malé změnové požadavky. Toto téma mi nabízí možnost, jak si vyzkoušet vývoj aplikace od počátku implementace. Další důvod je ten, že si můžu vyzkoušet nové technologie, se kterými se v práci nesetkávám.

Co se týče osnovy této práce, začnu analýzou funkčních a nefunkčních požadavků, představím možné scénáře a use cases, se kterými se může uživatel setkat a samozřejmě neopomenou analýzu konkurence. Další kapitola se bude

ÚVOD

věnovat návrhu vystaveném na získaných datech z analýzy a výběru vhodných technologií. V následujících kapitolách přiblížím proces implementace a průběh uživatelského testování. Na závěr zhodnotím dosažené výsledky a vytyčím si možné úpravy a zlepšení do budoucna.

Analýza

V této kapitole představím průběh analýzy webové aplikace, způsob získávání informací a proces vypracování funkčních a nefunkčních požadavků ze získaných podkladů. Představím uživatelské scénáře. Následně realizuji analýzu konkurenčních řešení a na závěr představím proces volby technologií.

Zde je také vhodné upozornit, že většina procesu analýzy probíhala jako součást semestrálního projektu předmětu BI-SP2 vyučovaném na FIT ČVUT v Praze. V rámci této analýzy se mnou spolupracovala kolegyně Barbora Kyselová, která je členkou týmu kolem skupiny aplikací EduChild. Ostatními členy tohoto týmu jsou Petr Šíma, Miloš Popovič, Daniel Matoušek, Tomáš Hanko, Marek Trzaskalík a Tereza Langová, všichni studenti FIT ČVUT. Pokud se v práci budu dále odkazovat na tým, bude se jednat o tuto skupinu.

1.1 Mobilní aplikace EduChild

Na začátek je nutné popsat mobilní aplikaci EduChild, neboť se k některým jejím funkcím budu odkazovat v dalším textu.

Mobilní aplikace EduChild se snaží pomoci rodičům v tom, aby jejich, zejména malé, děti nestrávily většinu svého volna s mobilním telefonem v ruce. Toho se snaží docílit pomocí zámku nepovolených aplikací. Dítě má alokovaný volný čas, kdy si může na mobilu dělat, co chce. Po uplynutí tohoto času se zablokuje zakázané aplikace a pro jejich odemknutí je nutné absolvovat přiřazený kvíz v aplikaci EduChild.

1.2 Sběr podkladů

Prvotní sběr podkladů probíhal na pravidelných týmových online schůzkách. Mobilní aplikace měla v této době již hotovou analýzu¹ a proto počáteční

¹Ta byla provedena jako součást semestrální práce předmětu BI-SP1, který jsme já a Barbora Kyselová absolvovali s jiným týmem

analýza vychází z upravených požadavků mobilní aplikace. Nad těmito upravenými požadavky poté probíhali diskuse o tom, jak si jednotliví členové týmu představují funkčnost výsledné aplikace. Z těchto diskusí jsme² si dělali poznámky a v další iteraci analýzy jsme tyto poznámky adresovali.

1.3 Funkční požadavky

Webová aplikace EduChild bude obsahovat dvě role – rodiče a tvůrce kvízů (pedagogové). Roli tvůrce kvízů rozšiřuje role rodič, tedy veškeré možnosti, které bude mít tvůrce kvízů, bude mít k dispozici také rodič. Rodič navíc bude mít možnosti správy dětských účtů a další, které popíšu v následujících řádcích. Následující sekci jsem z důvodu přehlednosti proto rozdělil na tyto dvě role.

K jednotlivým požadavkům připisujeme i kategorii, kam daný požadavek spadá. Tyto kategorie vzešly z modelu *FURPS* [1][2]. Tento model byl vyvinut společností Hewlett-Packard v 90. letech 20. století. Název je akronymem zastupující 5 základních hledisek, skrze které se můžeme dívat na kvalitu softwaru. Těmito hledisky jsou:

- **F**unctionality (funkčnost),
- **U**sability (použitelnost),
- **R**eliability (spolehlivost),
- **P**erformance (výkon),
- **S**upportability (podpora).

1.3.1 Rodič

Tato role se vyznačuje tím, že její představitelé budou mít děti, které budou mít nainstalovanu mobilní aplikaci EduChild na svém telefonu. Rodiče potřebují mít k dispozici možnost kontrolovat mobilní aktivitu, nastavovat rozvrh a čas, kdy si dítě může na mobilu hrát bez přerušování. Rodiče často nemají tolik času, kolik by si přáli a proto uvítají možnost získat již před-připravené kvízy z obchodu, čímž se vyhnou nutnosti trávit čas jejich tvorbou.

Na základě těchto skutečností jsme vymezili následující funkční požadavky pro rodiče:

F1 - Správa blokováných aplikací

Popis: Uživatel bude moci svým podřízeným dětským účtům spravovat povolení pro jednotlivé aplikace nainstalované na telefonu dítěte.

²Já a Barbora Kyseřová

Kategorie: F (funkčnost)

Priorita: Střední

F2 - Správa volného času

Popis: Uživatel bude moci nastavit dětem délku volného času, kdy nebudou aplikace blokovány.

Kategorie: F

Priorita: Střední

F3 - Nastavování rozvrhu

Popis: Uživatel bude moci nastavit svým dětem rozvrh³. Možná bude tvorba nového rozvrhu, editace již vytvořených rozvrhů a smazání rozvrhu.

Kategorie: F

Priorita: Střední

F4 - Správa dětských kvízů

Popis: Uživatel bude mít kontrolu nad dětskými kvízy. Mezi tyto akce se řadí např. odebrání kvízu dítěti či jeho přiřazení.

Kategorie: F

Priorita: Vysoká

F5 - Zobrazení dětských statistik

Popis: Uživatel bude mít přehled o stavu jednotlivých kvízů jako je např. zda byl kvíz splněn či jak moc bylo dítě úspěšné. Zároveň bude mít k dispozici informace o dětské aktivitě na telefonu, jako např. jak mnoho času strávilo dítě nad různými aplikacemi.

Kategorie: F

Priorita: Střední

F6 - Správa dětského účtu

Popis: Uživatel bude moci editovat dětský účet. Prozatím se bude jednat o možnost změny obrázku a jména. Dětský účet bude možné i smazat.

Kategorie: F

Priorita: Vysoká

³Aneb kdy budou vybrané aplikace blokovány a nepůjdou odblokovat ani po absolvování kvízu např. v době školní výuky

1.3.2 Tvůrce kvízů

Na rozdíl od role rodiče se tvůrce charakterizuje absencí dětí a tedy i potřeby je nějak hlídat či spravovat, jak tráví svůj čas na mobilu. Webovou aplikaci navštěvuje za účelem tvorby kvízů. Může se jednat například o pedagoga na ZŠ, který tvoří kvízy jako doplněk k práci při vyučování.

Vážeme k němu následující funkční požadavky:

F7 - Správa uživatelského účtu

Popis: Uživatel bude mít možnost editovat svůj účet. V době psaní této práce se jedná o nastavení preferovaného jazyka webové aplikace. Dále může uživatel změnit své heslo nebo smazat svůj účet.

Kategorie: F

Priorita: Vysoká

F8 - Tvorba a editace kvízů

Popis: Uživatel bude mít k dispozici nástroj na pohodlné tvoření nových a editaci stávajících kvízů.

Kategorie: F

Priorita: Vysoká

F9 - Sdílení kvízů do obchodu

Popis: Uživatel bude mít možnost jím vytvořené kvízy sdílet do obchodu (část webové aplikace, kde bude možné získávat cizí kvízy). Toto sdílení může kdykoliv zrušit a svůj kvíz z obchodu odstranit.

Kategorie: F

Priorita: Nízká

F10 - Správa uživatelských kvízů

Popis: Uživatel bude mít přehled o všech svých kvízech. Jím vytvořené kvízy bude moci upravovat či mazat.

Kategorie: F

Priorita: Vysoká

1.4 Nefunkční požadavky

Webová aplikace se nachází v raném stadiu, kdy ještě nebyla spuštěna do provozu⁴. Proto jsou následující nefunkční požadavky spíše obecného charakteru:

⁴V době psaní textu

N1 - Bezpečnost aplikace a jejích dat:

Popis: Aplikace bude operovat s citlivými uživatelskými daty jako jsou např. email nebo profilové foto. Je proto důležité, aby tato data byla bezpečně uložena.

Kategorie: F

N2 - Jednoduchá rozšiřitelnost a údržba:

Popis: V případě nutnosti v budoucnu rozšířit aplikaci je nutné, aby tyto úpravy bylo možno realizovat snadno a rychle bez nutnosti výrazně změnit dosavadní systém.

Kategorie: S

N3 - Dostupnost a stabilita:

Popis: Webová aplikace by měla být dostupná svým uživatelům, kdykoli ji potřebují. Je proto nezbytné, aby aplikace byla dostatečně robustní a bez chyb, čehož se dá dosáhnout pomocí širokého pokrytí testy.

Kategorie: R

N4 - Komunikace přes standard REST⁵:

Popis: Komunikace s externími službami bude probíhat skrze technologii REST.

Kategorie: S

N5 - Lokalizace do více jazyků:

Popis: Aplikace bude od počátku podporovat dva jazyky (český a anglický). V budoucnu bude možné přidat i jiné lokalizace a aplikace na to musí být technicky připravena.

Kategorie: U

1.5 Případy užití a scénáře

Případ užití (anglicky *use case*) se při softwarové analýze používá k popsání interakcí mezi hercem (*actor*) a systémem. Hercem může být koncový uživatel nebo jiný externí systém. Jedná se o seznam akcí, které popisují, jak mezi sebou systém a herec interagují, aby dosáhli nějakého předem definovaného cíle. Nebere se v nich v potaz výsledný vzhled uživatelského rozhraní ani použité technické řešení.

Následující přehled neobsahuje veškeré případy užití, pouze stěžejní výběr. Veškeré případy použití jsou k dispozici v příloze B. Číslování případů užití odpovídá seznamu v příloze.

⁵Representational state transfer

1. ANALÝZA

1.5.1 Sledování dětské aktivity

UC-1 Zobrazení poslední dětské aktivity

Role: Rodič

Popis: Uživatel uvidí informace o dětské aktivitě na mobilu.

Scénář:

1. Systém zobrazí obrazovku pro výběr dítěte.
2. Uživatel zvolí dítě.
3. Systém zobrazí kartu dítěte s jeho poslední aktivitou.

UC-4 Zobrazení detailu vyplněného kvízu

Role: Rodič

Popis: Uživatel si zobrazí detail vyplněného kvízu.

Scénář:

1. Systém dodá obrazovku, kde bude seznam vyplněných kvízů.
2. Uživatel zvolí ze seznamu kvíz.
3. Systém zobrazí detail kvízu včetně správných odpovědí a odpovědí zadaných dítětem.

1.5.2 Správa uživatelových kvízů

UC-6 Odebrání kvízu z uživatelových kvízů

Role: Tvůrce kvízů

Popis: Uživatel odstraní kvíz.

Scénář:

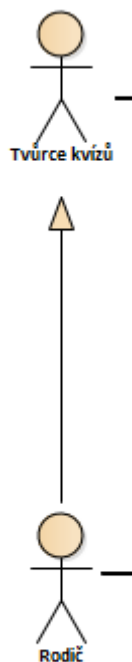
1. Systém poskytne obrazovku dostupných kvízů.
2. Uživatel najde v seznamu nebo si vyhledá kvíz, který chce odstranit.
3. Uživatel klikne na akci smazání kvízu.
4. Systém vyžaduje po uživateli potvrzení akce.
5. Uživatel potvrdí akci.
6. Systém odstraní kvíz z uživatelových kvízů. V případě, že kvíz byl vytvořen uživatelem, odstraní ho i z databáze.

UC-7 Vyhledávání kvízů v seznamu

Role: Tvůrce kvízů

Popis: Uživatel si na stránce dostupných kvízů vyhledá konkrétní kvíz podle vyhledávacích kritérií.

Scénář:



Obrázek 1.1: Vztah rolí

1. Systém dodá obrazovku s kvízy.
2. Uživatel zadá kritéria do vyhledávacího filtru.
3. Systém podle kritérií vyfiltruje příslušné kvízy.

1.5.3 Tvorba kvízů

UC-8 Vytvoření nového kvízu

Role: Tvůrce kvízů

Popis: Uživatel vytvoří vlastní kvíz.

Scénář:

1. Systém dodá obrazovku s kvízy.
2. Uživatel spustí akci tvorby nového kvízu.
3. Systém zobrazí obrazovku s formulářem pro tvorbu kvízu.
4. Uživatel vyplní povinné informace o kvízu.
5. Uživatel přidá alespoň jednu otázku.
6. Uživatel ukončí tvorbu kvízu.
7. Systém uloží kvíz.

1. ANALÝZA

UC-9 Vytvoření otázky

Role: Tvůrce kvízů

Popis: Uživatel při tvorbě kvízu vytvoří otázku.

Scénář:

1. Uživatel se nachází na stránce dostupných kvízů.
2. Uživatel spustí akci pro vytvoření nové otázky.
3. Systém zobrazí nový formulář pro výběr typu otázky.
4. Uživatel vybere typ otázky.
5. Systém zobrazí formulář pro daný typ otázky.
6. Uživatel vyplní příslušná pole (textem nebo přiloží obrázek).
7. Uživatel spustí akci tvorby nové otázky nebo akci ukončení tvorby kvízu.
8. Systém uloží otázku.

UC-13 Editace kvízu

Role: Tvůrce kvízů

Popis: Uživatel edituje jím dříve vytvořený kvíz.

Scénář:

1. Uživatel se nachází na stránce dostupných kvízů.
2. Uživatel spustí akci editace kvízu u záznamu vybraného kvízu.
3. Systém zobrazí obrazovku editace kvízu.
4. Uživatel upraví informace o kvízu a/nebo znění otázek. Může přidat či smazat otázky.
5. Uživatel ukončí tvorbu kvízu.
6. Systém uloží kvíz.

1.5.4 Sdílení kvízů

UC-14 Sdílení kvízu do obchodu

Role: Tvůrce kvízů

Popis: Uživatel sdílí kvíz s komunitou.

Scénář:

1. Uživatel se nachází na stránce dostupných kvízů.
2. Uživatel spustí akci sdílení u vybraného kvízu.
3. Systém přidá kvíz do obchodu.

UC-15 Sdílení kvízu při tvorbě

Role: Tvůrce kvízů

Popis: Uživatel na konci tvorby kvízu přidá kvíz do obchodu.

Scénář: Navazuje na případ užití *Vytvoření nového kvízu*.

1. Systém zobrazí obrazovku s možností sdílet kvíz s komunitou.
2. Uživatel potvrdí sdílení.
3. Systém kvíz přidá do obchodu.

1.5.5 Obchod

UC-17 Vyhledání kvízu v obchodě

Role: Tvůrce kvízů

Popis: Uživatel v obchodě vyhledá kvízy podle zadaných kritérií.

Scénář:

1. Uživatel se nachází na obrazovce obchodu.
2. Uživatel vyplní vyhledávací filtry podle svých potřeb.
3. Systém vyhledá kvízy odpovídající zadaným filtrům.

UC-20 Stažení kvízu

Role: Tvůrce kvízů

Popis: Uživatel si stáhne cizí kvíz do vlastní kolekce kvízů.

Scénář:

1. Uživatel se nachází na obrazovce detailu kvízu.
2. Uživatel spustí akci stažení kvízu.
3. Systém stáhne kvíz do uživatelova seznamu kvízů.

UC-21 Přidělení kvízu dítěti

Role: Rodič

Popis: Uživatel přiřadí cizí kvíz jednomu ze svých dětí.

Scénář:

1. Uživatel se nachází na obrazovce detailu kvízu.
2. Uživatel vyvolá akci přiřazení kvízu.
3. Systém nabídne děti.
4. Uživatel vybere, kterým dětem chce kvíz přiřadit.

1. ANALÝZA

5. Systém stáhne kvíz do uživateli kolekce a přiřadí ho vybraným dětem.

UC-23 Nahlášení kvízu

Role: Tvůrce kvízů

Popis: Uživatel shledá, že daný kvíz má nevhodný obsah a nahlásí ho.

Scénář:

1. Uživatel se nachází na stránce detailu kvízu.
2. Uživatel začne akci nahlášení.
3. Systém zobrazí formulář reportu.
4. Uživatel vyplní formulář a odešle ho.
5. Systém uloží hlášení.

1.5.6 Správa aplikací

UC-24 Přidání aplikace do zakázaných aplikací

Role: Rodič

Popis: Uživatel přidá aplikaci do zakázaných aplikací dítěte.

Scénář:

1. Uživatel se nachází na obrazovce správy dětí.
2. Uživatel vybere dítě.
3. Systém zobrazí kartu dítěte s aplikacemi.
4. Uživatel kliknutím na aplikaci označí, že ji chce zakázat.
5. Systém uloží volbu a graficky indikuje, že je aplikace zakázána.

1.5.7 Správa uživatelského účtu

UC-27 Změna hesla

Role: Tvůrce kvízů

Popis: Uživatel změní své přístupové heslo.

Scénář:

1. Uživatel se nachází na obrazovce nastavení.
2. Uživatel spustí akci změny hesla.
3. Systém zobrazí obrazovku s formulářem pro změnu hesla.
4. Uživatel vyplní formulář a ukončí změnu.
5. Systém uloží změnu hesla.

UC-29 Smazání účtu

Role: Tvůrce kvízů

Popis: Uživatel smaže svůj účet.

Scénář:

1. Uživatel se nachází na obrazovce nastavení.
2. Uživatel inicializuje akci zrušení účtu.
3. Systém vyžaduje po uživateli potvrzení akce.
4. Uživatel potvrdí akci.
5. Systém odešle kód na email uživatele.
6. Systém zobrazí formulář na vyplnění potvrzovacího kódu.
7. Uživatel zadá kód a odešle formulář.
8. Systém uživatele odhlásí, přesměruje na domovskou stránku a smaže jeho účet.

1.5.8 Správa dětských účtů

UC-30 Změna dětských údajů

Role: Rodič

Popis: Uživatel změní základní informace o dětském účtu.

Scénář:

1. Systém zobrazí obrazovku pro výběr dítěte.
2. Uživatel vybere dítě.
3. Systém zobrazí kartu dítěte.
4. Uživatel spustí akci editace dítěte.
5. Systém zobrazí editační formulář.
6. Uživatel vyplní formulář a odešle ho.
7. Systém uloží nové informace o dětském účtu.

UC-31 Smazání dětského účtu

Role: Rodič

Popis: Uživatel smaže dětský účet.

Scénář:

1. Uživatel se nachází na kartě dítěte.
2. Uživatel spustí akci editace dítěte.
3. Systém zobrazí editační panel.

1. ANALÝZA

4. Uživatel vyvolá akci smazání účtu.
5. Systém vyžaduje po uživateli potvrzení akce.
6. Uživatel potvrdí akci.
7. Systém odešle kód na email uživatele.
8. Systém zobrazí formulář na vyplnění potvrzovacího kódu.
9. Uživatel zadá kód a odešle formulář.
10. Systém smaže dětský účet a přesměruje uživatele na obrazovku se záložkami pro výběr dítěte.

1.5.9 Správa dětského času

UC-32 Přidání nového rozvrhu

Role: Rodič

Popis: Uživatel přidá dítěti nový rozvrh.

Scénář:

1. Uživatel se nachází na kartě dítěte.
2. Uživatel spustí akci přidání rozvrhu.
3. Systém zobrazí panel s formulářem pro tvorbu rozvrhu.
4. Uživatel vyplní formulář a uloží ho.
5. Systém uloží rozvrh a nastaví ho vybranému dítěti.

UC-33 Aktivace rozvrhu

Role: Rodič

Popis: Uživatel zapne dítěti vybraný rozvrh.

Scénář:

1. Uživatel se nachází na kartě dítěte.
2. Uživatel spustí akci aktivaci rozvrhu.
3. Systém aktivuje rozvrh.

UC-37 Správa volného času dítěte

Role: Rodič

Popis: Uživatel nastaví dítěti délku volného času.

Scénář:

1. Uživatel se nachází na kartě dítěte.
2. Uživatel změní hodnoty na formuláři volného času.
3. Systém uloží změny.

1.6 Analýza konkurenčních řešení

Než přejdu k procesu návrhu aplikace, je žádoucí provést analýzu konkurence. Jelikož aplikace nemá pouze jeden cíl (tvorba kvízů, správa dětských účtů) bude následující sekce rozdělena na dvě části. První část se bude zabývat konkurenčními řešeními z pohledu tvorby kvízů a správy kvízů. Druhá část se zaměří na aplikace, které dokáží kontrolovat svěřené účty (ať už se jedná o dětské účty nebo například pracovní). Na závěr vyhodnotím zjištěné informace.

Tato analýza se zaměřuje na UI⁶ a UX⁷ design. Části nadepsané jako *Výhody* a *Nevýhody* se vztahují k UI a UX. Díky tomu dostanu aktuální přehled o stavu aplikací zaměřujících se na toto téma. V důsledku toho se budu moci vyvarovat možných chyb při návrhu samotné webové aplikace. Na konci ještě jednotlivé aplikace ohodnotím podle mé zkušenosti s tím, jak se mi s nimi pracovalo. Hodnocení bude probíhat jako ve škole (1 - nejlepší, 5 - nejhorší).

1.6.1 Analýza aplikací zabývajících se tvorbou kvízů

1.6.1.1 Google Forms

Vyvinuto společností Google, *Google Forms* [3] je nástroj pro tvorbu kvízů, anket, průzkumů mínění a mnoha dalších formulářů. Je dostupný zdarma. Výsledné formuláře se automaticky ukládají na Google účet a jsou kdykoliv přístupné na Google Drive. To tedy znamená, že k používání je potřeba účet Google.

Zajímavé funkce:

- Import otázek z jiných formulářů,
- import videí,
- vložení obrázku z Google Search,
- možnost duplikace otázek,
- přidání hodnocení k jednotlivým otázkám,
- dělení formuláře na sekce podle uživatelova uvážení,
- funkce *Undo*, tedy možnost vrátit poslední provedený krok zpět,
- možnost označit otázku jako povinnou/nepovinnou,
- možnost přidat vlastní validaci otázek (uživatel může např. dodat validaci typu *Vyberte maximálně X možností.*),

⁶User interface

⁷User experience

1. ANALÝZA

- nastavení náhodného promíchávání otázek.

Výhody:

- Při označení odpovědi za správnou se zobrazí u příslušné odpovědi „“ pro indikaci správné odpovědi,
- aplikace využívá Material design,
- při zobrazení seznamu formulářů lze přepínat mezi dlaždicovým a seznamovým zobrazením.

Nevýhody:

- Při přiložení obrázku k otázce, nelze přiložený obrázek přiblížit.
- Pokud je ve formuláři vytvořeno více otázek, nelze se mezi nimi rychlým způsobem navigovat, je nutné se po formuláři pohybovat pomocí vertikálního posuvníku.
- Při prvním vstupu do tvorby formuláře není jasné, kde lze formulář přejmenovat. UI tvorby formuláře je soustředěno na střed obrazovky, ale přejmenování kvízu je nahoře vlevo (viz. obrázek 1.2).

Hodnocení: S aplikací se mi pracovalo dobře, UI je ve většině případů jednoduché na pochopení a přehledné. Jediné, co mě významně omezovalo, byla absence rychlého přechodu na konkrétní otázku. Hodnotím tedy známkou 2.

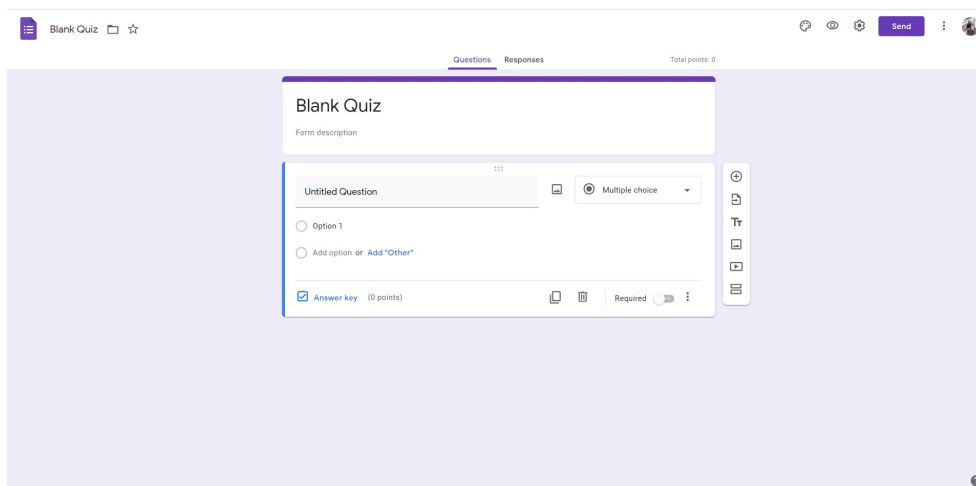
1.6.1.2 Microsoft Forms

Společnost Microsoft přišla s vlastním nástrojem pro tvorbu anket a testových formulářů. Microsoft Forms [4] je součástí balíčku *Office 365* od roku 2016.[5] Stejně jako Google Forms jsou veškeré jeho funkcionality přístupné zdarma, potřeba je pouze Microsoft účet. S tímto nástrojem jsem se seznámil již dříve, je totiž častou součástí bezkontaktní výuky.

Zajímavé funkce:

- Na základě zadané odpovědi lze formulář následně rozvést na relevantní části,
- ke každé odpovědi lze zadat zprávu, která se posléze zobrazí uživateli po vybrání dané odpovědi,
- možnost duplikace otázek,
- přidání bodování k otázce,

1.6. Analýza konkurenčních řešení



Obrázek 1.2: Ukázka nekonzistentního rozvržení Google Forms, kdy změna názvu formuláře probíhá vlevo nahoře místo očekávaného středu

- označení otázky jako povinné,
- možnost změnit náhodně pořadí odpovědí.

Výhody:

- Je možné přidat matematický zápis,
- u správné odpovědi se zobrazí grafická indikace správnosti v podobě textu a ikony zaškrtnutí.

Nevýhody:

- Po výběru typu otázky již nelze typ změnit, je nutné otázku smazat,
- přiložený obrázek nelze přiblížit (např. pro kontrolu),
- nelze přidat obrázek jako součást odpovědi.

Hodnocení: Ačkoli i s touto aplikací se mi pracovalo dobře, vnímám zde uvedené nedostatky jako více zásadní (zejména absence možnosti změnit otázku) a proto hodnotím známkou 3.

1.6.1.3 Survio

Survio je český projekt, který vznikl v Brně v roce 2012. Obsahuje několik desítek předem připravených vzorů dotazníků, samozřejmostí je i možnost

tvorby vlastního vzoru. Pro vytvoření dotazníku potřebujete mít založený účet [6].

Zajímavé funkce:

- Zajímavé typy otázek jako např. matice, sémantický diferenciál,
- možnost zapnout v dotazníku automatický posun na další otázku po vyplnění odpovědi,
- tisknutí nevyplněného dotazníku,
- sdílení dotazníku přes QR kód nebo přes tlačítko, které vložíte na svůj web.

Výhody:

- Umožňuje nastavit, jakým způsobem se zobrazí odpovědi – vedle sebe, pod sebou,
- možnost určit maximální a minimální počet odpovědí na otázku,
- u otevřené otázky lze nastavit, jaký druh odpovědi se očekává – textová, číselná, datum nebo datum,
- jedním kliknutím lze zkopírovat celý dotazník.

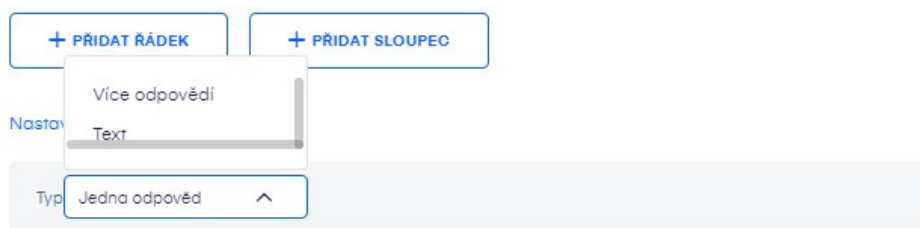
Nevýhody:

- Po přidání otázky už nelze změnit její typ, je potřeba celou otázku smazat a vytvořit novou,
- sekce *Nastavení* u otázek obsahuje komponentu výběru, která mimo možností zobrazuje také invazivní horizontální a vertikální posuvníky, které částečně zakrývají možnosti. Navíc to evokuje, že je k dispozici, že je k dispozici ještě více možností, i když tomu tak není (viz 1.3).

Hodnocení: Stejná nevýhoda jako u předchozí aplikace, zde ovšem k lepšímu hodnocení přispívá větší počet otázek na výběr a možnost určit, co za odpověď má následovat na otevřenou otázku. Hodnotím za 2.

1.6.1.4 Fyrebox

Fyrebox je nástroj pro tvorbu dotazníků určený pro firmy a organizace. Také je to první zástupce v této kapitole, který má zdarma dostupnou jen základní množinu funkcí, zbytek funkcí se skrývá za platební bránou. Pro vytvoření dotazníku je potřeba mít založený účet, lze se ale přihlásit také skrze Google nebo Facebook účet [7].



Obrázek 1.3: Ukázka špatně nastavené komponenty výběru v nástroji Survio

Zajímavé funkce:

- Při prvním přihlášení jsou k dispozici vzory k rychlému spuštění a vyzkoušení si, jak může vypadat dotazník,
- možnost vytvořit ryze obrázkový kvíz (bohužel jedna z prémiových funkcí).

Výhody:

- Možnost si detailně upravit vzhled kvízu např. obrázek na pozadí, barvy tlačítek, font, barvu textu aj.,
- nástroj zobrazuje po levé straně postranní panel, kde jsou umístěni zástupci jednotlivých sekcí kvízu a při jejich poklepání dojde k rychlému zobrazení požadované sekce.

Nevýhody:

- Při tvorbě kvízu je nutné vybrat typ kvízu z předem definovaných druhů (jedna odpověď, více odpovědí, otevřené otázky) a tím pádem nelze kombinovat různé druhy otázek v jednom kvízu,
- při upozornění uživatele na placenou funkci je text v naprosto odlišném a špatně čitelném fontu. Font jsem navíc nikde jinde v aplikaci nespatriřil, patří zde tedy nepatříčně,
- při nastavování základních hodnot kvízu se několikrát po sobě samo od sebe spustí krátké instruktážní video, které má uživatele upozornit na možnost si zobrazit náhled dotazníku. Pro uživatele to začne být dost rušivé, když se to stane počtvrté na jedné a té samé stránce.

Hodnocení: S touto aplikací se mi pracovalo asi nejhůře ze všech zde vyjmenovaných. Mnoho funkcí jsem neměl možnost vyzkoušet, neboť byly ukryté za platební bránou. Nemožnost mít v kvízu více druhů otázek je velmi svazující a proto v konečném důsledku hodnotím známkou 4.

1.6.1.5 Kahoot!

Kahoot! je platforma pro hraní, sdílení a tvorbu kvízů. Na trh vstoupil v roce 2013, od té doby se těší velké podpoře hlavně v amerických školách [8]. Lze se přihlásit přes jeden z Apple, Microsoft či Google účtů, popřípadě je možné založit svůj Kahoot! účet. Základní funkce jako je tvorba elementárních kvízů jsou součástí bezplatného plánu, ostatní jsou součástí dalších tří platebních plánů.

Zajímavé funkce:

- Na otázky lze nastavit časový limit,
- je možné prohledat veškeré otázky v systému pomocí klíčových slov,
- lze importovat otázku v Excel tabulce,
- lze přidat video ze serveru Youtube.

Výhody:

- Odpovědi jsou odlišeny pomocí barev a geometrických tvarů.

Nevýhody:

- Nelze přidat text a zároveň obrázek jako odpověď,
- při náhledu kvízu není k dispozici možnost ho přerušit, je nutné se vracet přes šipku zpět v prohlížeči.

Hodnocení: Již podle vzhledu je poznat, že aplikace je zaměřena na děti. Tvorba i vyplňování kvízů probíhalo v pořádku. Našel jsem malé nedostatky, ale v celkovém vyznění se mi s aplikací pracovalo dobře. Hodnotím známkou 2-.

1.6.2 Analýza aplikací zabývajících se správou svěřených účtů

1.6.2.1 Microsoft Family Safety

Microsoft nabízí Microsoft Family Safety jako své řešení pro rodiče, kteří potřebují ohlídat aktivitu svých potomků na digitálních zařízeních. Pro fungování potřebuje pouze účet Microsoft a zaštiťuje všechna zařízení Windows, zařízení Xbox a nyní už i mobilní zařízení Android a iOS [9]. Nabízí zdarma funkcionality jako např. sledování času u obrazovky, správu vhodného obsahu a další.

Zajímavé funkce:

- Nastavení času u obrazovky,
- blokování určitých kategoriích či přímo konkrétních webových stránek,
- týdenní hlášení o aktivitách.

Výhody:

- Přehledné, jednotlivé prvky UI na sebe logicky navazují.

Nevýhody:

- Nenalezl jsem. Nelíbí se mi mdlé barvy, ale to je pouze má osobní preference.

Hodnocení: Absence významných nevýhod indikuje moji spokojenost s používáním této aplikace. I když jsem aplikaci viděl poprvé, pohyb v ní byl intuitivní a plynulý. Hodnotím známkou 1.

1.6.2.2 Google Families

Google nabízí aplikaci Google Families, která funguje napříč všemi zařízeními s Google účtem. Aplikace nabízí klasické funkcionality jako např. kontrolu času u zařízení, kontrolu obsahu a mnoho dalšího. V rodinné skupině může být až 6 členů. Je k dispozici zdarma.

Zajímavé funkce:

- Omezení vyhledávacích filtrů dle obsahu

Výhody:

- Přehledný, jednoduchý design
- Material design

Nevýhody:

- Bez výhrad.

Hodnocení: Stejně jako s jejím konkurentem od Microsoftu, i s touto aplikací jsem byl spokojený při jejím užívání. Google se drží svého zaběhlého a ozkoušeného designu a funguje to. Hodnotím také za 1.

1.6.3 Shrnutí

Výše zmíněné body mi pomohly si ucelit představu o tom, jak chci, aby se uživatelé v naší aplikaci cítili. Některé vyzorované výhody jsme posléze začlenili do návrhu a implementace. Jednalo se například o možnost vložení textu i obrázku do otázek a odpovědí. Dalším důležitým poznatkem byla například nutnost pro uživatele mít přístup k rychlé navigaci mezi otázkami. Zároveň

| Aplikace | Hodnocení |
|-------------------------|-----------|
| Google Forms | 2 |
| Microsoft Forms | 3 |
| Survio | 2 |
| Fyrexbox | 4 |
| Kahoot! | 2- |
| Microsoft Family Safety | 1 |
| Google Families | 1 |

Tabulka 1.1: Hodnocení konkurenčních aplikací

jsme věnovali pozornost tomu, aby nedošlo k opakování nevýhod jako například nemožnost změnit typ otázky po jejím vytvoření či schopnost přiblížení vloženého obrázku. Na závěr zde ještě shrnu výsledná hodnocení v tabulce 1.1.

Návrh

V této kapitole se budu věnovat problematice návrhu uživatelského rozhraní, výběru architektury a v poslední části popíšu vybrané technologie.

2.1 Task list

Ještě před samotným návrhem uživatelského rozhraní je vhodné si vymodelovat, které akce a zobrazovací prvky budou uživatelům viditelné v závislosti na jejich roli. K tomuto slouží task list. Jedná se o seznam akcí a obrazovek, který nám popisuje uživatelské rozhraní z hlediska klienta. V mém případě je postaven na již vytvořeném seznamu případů užití z kapitoly 1.5. Jednotlivé body jsem seskupil do tří oddílů podle toho, kdo je může vidět či je spustit: uživatelé, přihlášení uživatelé a uživatelé s rolí rodič.

2.1.1 Uživatelé

Akce

- Přihlásit se
- Registrovat se
- Zvolit zobrazovací jazyk
- Kontaktovat tvůrce aplikace
- Získat informací o aplikaci

Zobrazení

- Přihlašovací dialog pomocí emailu a hesla
- Přihlašovací dialog přes Google

2. NÁVRH

- Registrační dialog
- Dialog obnovy ztraceného hesla
- Formulář pro kontakt tvůrců
- Zobrazit informace o aplikaci

2.1.2 Přihlášení uživatelé

Akce

- Navigace
 - Obchod
 - Kvízy
 - Nastavení
 - Odhlášení
- Tvorba kvízu
- Editace kvízu
- Filtrování kvízů
- Sdílení kvízu
- Odebrání kvízu
- Aktualizace kvízu
- Změna preferovaného jazyka
- Změna hesla
- Změna role
- Smazání účtu
- Stahování kvízu
- Ohodnocení kvízu
- Nahlášení kvízu

Zobrazení

- Dashboard uživatelových kvízů
- Formulář úpravy kvízu
- Formulář tvorby kvízu
- Seznam dostupných kvízů
- Detail kvízu
- Detail otázek kvízu
- Panel nahlášení kvízu

2.1.3 Role Rodič

Akce

- Navigace
 - Děti
- Přiřazení kvízu
- Zakázání aplikace
- Povolení aplikace
- Přidání rozvrhu
- Odebrání rozvrhu
- Editace rozvrhu
- Editace volného času
- Editace dětského účtu
- Smazání dětského účtu

Zobrazení

- Dětský dashboard
- Panel editace rozvrhu
- Panel statistik
- Panel přiřazení kvízu

2.2 Návrh uživatelského rozhraní (Li-Fi prototyp)

Prvním krokem po získání údajů z analýzy bylo vytvoření *papírového modelu*, lze také použít označení wireframes. Na základě získaných dat jsme si vytyčili základní principy, kterými jsme se při návrhu snažili do co nejvyšší možné míry řídit. Tyto principy jsou:

- Uživatel musí vždy vědět, kde se nachází.
- Aplikace musí být přehledná. To znamená, že GUI⁸ navrhujeme tak, aby bylo na první pohled jasné, kde uživatel najde jednotlivé informace, akce a funkce systému.
- Snažíme se předejít chybám uživatele. Toho se dá docílit například zvýrazněním určitých ovládacích prvků pomocí barev či úpravou jejich rozměrů. Dále je důležité, aby popisky ovládacích prvků byly stručně a výstižně popsány. U některých prvků lze přidat ikony, které uživatelům pomohou s rychlejší orientací.

Při tvorbě papírového modelu jsme se⁹ inspirovali z několika zdrojů. Jedním z nich byla provedená analýza konkurenčních řešení, dalším zdrojem potom byly grafické návrhy Android aplikace Educhild a v neposlední řadě se jednalo o vypracované scénáře. Z návrhů pro Android jsme se inspirovali hlavně proto, aby pro uživatele, kteří již mají mobilní aplikaci, byl přechod na webovou aplikaci co nejsnadnější. Každý vytvořený papírový model jsme představili na pravidelných týmových schůzkách, kde byl prostor pro jednotlivé členy týmu vznést dotazy či připomínky. Tyto připomínky jsme poté adresovali v další iteraci návrhu.

2.2.1 Volba nástroje

Pro tvorbu wireframes existuje na internetu mnoho kvalitních nástrojů. Pro nás byly důležitými požadavky na tento nástroj především rychlá křivka učení, možnost sdílet kompletní modely a také aby měl nástroj verzi zdarma. Na základě těchto kritérií jsme nakonec vybrali nástroj Figma [10]. Výhodou pro nás bylo, že zbytek týmu s ním již měl zkušenost a používal ho pro tvorbu grafického návrhu pro mobilní aplikaci.

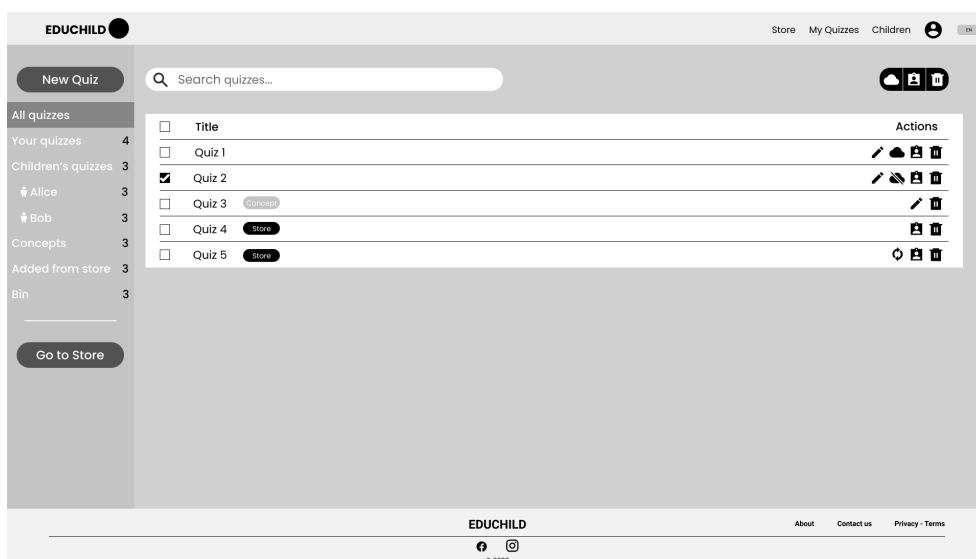
2.2.2 Wireframes

V této podkapitole představím některé výsledky práce na návrhu. Nebudu popisovat všechny návrhy, neboť by se potom tato kapitola příliš natáhla. Všechny návrhy budou dostupné v příloze C. Na začátek této sekce upozorním,

⁸Graphical User Interface

⁹Já a Barbora Kyseřová

2.2. Návrh uživatelského rozhraní (Li-Fi prototyp)



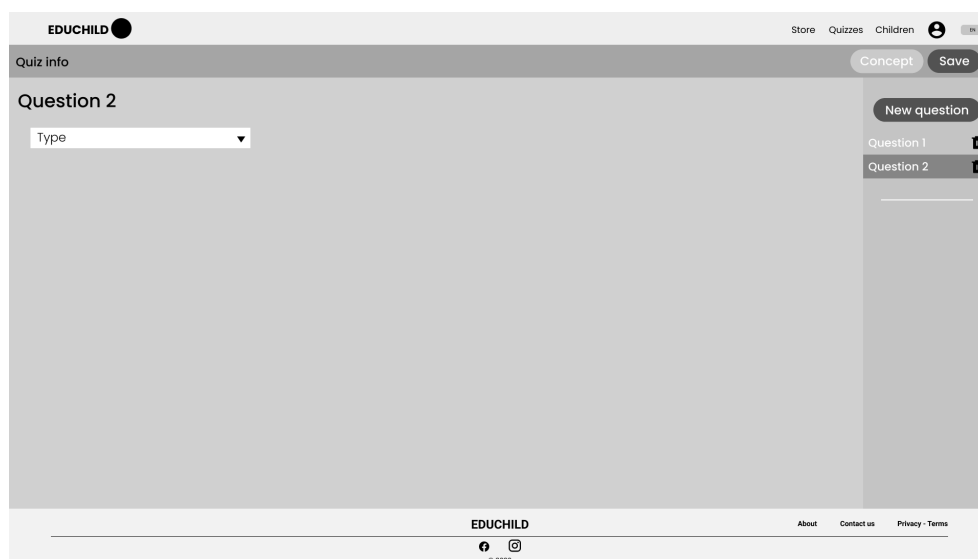
Obrázek 2.1: Wireframe obrazovky Moje kvízy

že jednotlivé wireframes jsou v angličtině. Důvodem proto je, že aplikace bude od začátku dostupná v anglickém i českém jazyce a při navrhování jsme se tak dohodli.

Moje kvízy Tato obrazovka je první ze dvou stěžejních obrazovek aplikace. Budou na ni mít přístup obě role uživatelů. Návrh je možné vidět na obrázku 2.1. Při návrhu jsme se snažili využít co nejvíce horizontálního místa. Podél levé strany se nachází postranní panel sloužící k navigaci mezi předem definovanými skupinami kvízů. Čísla u jednotlivých skupin dávají uživateli okamžitě na vědomí, kolik daná skupina obsahuje kvízů. Většinu obrazovky zabírá potom seznam kvízů v právě zvolené skupině. U kvízu potom můžeme provést jednu z akcí, které jsou na konci řádků. Samozřejmostí je nápověda po najetí myši nad ikonu. Pokud chceme provést hromadnou akci, označíme kvízy a vybereme jednu z akcí v pravém horním rohu, které se dynamicky objevili po zaškrtnutí prvního kvízu. Je zde vidět velká inspirace emailovými klienty jako například Gmail od společnosti Google.

Tvorba kvízu Tato obrazovka byla nejčastěji přepracována ze všech. Jeden z prvotních návrhů, jak je vidět na obrázku 2.2, neefektivně pracoval se šířkou obrazovky a obsahoval příliš mnoho nevyužitého místa, které by mohlo v uživateli evokovat, že se nějaká část aplikace nenačetla správně. Po extenzivních úpravách jsme se dopracovali k současné verzi na obrázku 2.3. Obrazovka je rozdělena na tři sloupce. V levém sloupci je náhled aktuální otázky tak, jak

2. NÁVRH



Obrázek 2.2: Jeden z prvních návrhů obrazovky Tvorba kvízu

bude vypadat v mobilní aplikaci. V prostředním sloupci je formulář současně otázky a poslední část vpravo obsahuje seznam otázek. Seznam je interaktivní, po kliknutí na záznam systém uživatele přenesse na vybraný záznam.

Děti Nejzásadnější obrazovka pro uživatele – rodiče, obrazovka správy dětí (obrázek 2.4) je tvořena panelem, kde každé dítě má vlastní záložku. Obsah záložky je dělen na různě velké řádky, které postupně obsahují panel se statistikami kvízů a užívání telefonu, panel s rozvrhy a volným časem, panel aplikací a sekci přiřazených kvízů. Šlo nám o to, aby veškeré akce, které může rodič v souvislosti s dítětem provést, byly dostupné z jednoho místa.

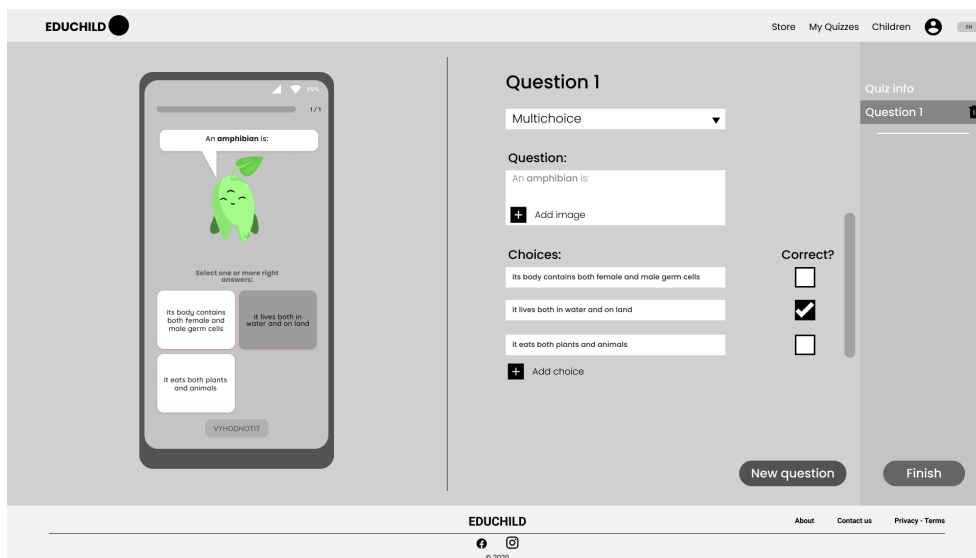
2.3 Architektura

2.3.1 Single Page Application – SPA

Jednou z možností, jak tvořit webové aplikace, je využít architektury SPA – Single Page Application (do češtiny lze přeložit jako „jednostránková aplikace“, ale je zvykem nepřekládat tento termín).

Tradiční webová aplikace funguje tak, že většina aplikační logiky probíhá na serveru a po vykonání se výsledek odešle klientovi. To se označuje jako SSR¹⁰ (v následujícím textu budu často tyto termíny zaměňovat). Na rozdíl od toho, SPA pracuje tak, že po prvotním načtení aplikace se většina práce

¹⁰Server-Side Rendering



Obrázek 2.3: Současný návrh obrazovky Tvorba kvízů

vykoná v klientově prohlížeči a po serveru se vyžadují pouze data. Při prvotním načtení aplikace od serveru obdrží potřebný HTML¹¹ soubor s uživatelským rozhraním (proto „jednostránková“), JS¹² popisující logiku aplikace a CSS¹³ styly. Každé další překreslení je už v režii prohlížeče, který reaguje na dodaná data získaná skrze AJAX¹⁴ požadavky.

Podrobně popíši rozdíl na příkladu (volně převzato z [11]): Mějme v naší webové aplikaci stránku, která vykresluje obrázek se stromem a domečkem. SSR aplikace vykreslí celý obrázek na serveru. SPA si nechá na serveru vygenerovat vzor a nechá si ho poslat, společně s vybranými barvami, kterými potom vybarví vzor. Prozatím mezi těmito dvěma rozdílnými aplikacemi není poznat větší rychlostní rozdíl. Co se ale stane, když například chceme změnit styl stromu? SSR aplikace odešle požadavek na server, který znovu celý obrázek s novým typem stromu vygeneruje. U SPA se odešle požadavek na nový typ, zde ale server rozpozná, že se nemění celá obsah, ale jen část obsahu a odešle pouze změněný obsah, v našem případě nový strom. Aplikace poté nahradí starý strom novým a zbytek obsahu nemění. Tím, že vykreslování stránky probíhá u klienta (v prohlížeči), může být stránka dynamicky překreslena, namísto toho, aby server vykresloval znovu celou stránku.

Jak lze nahlédnout v [12], po úvodním načtení si aplikace ukládá získaná data do lokálního úložiště. To má za následek, že aplikace může v omezené

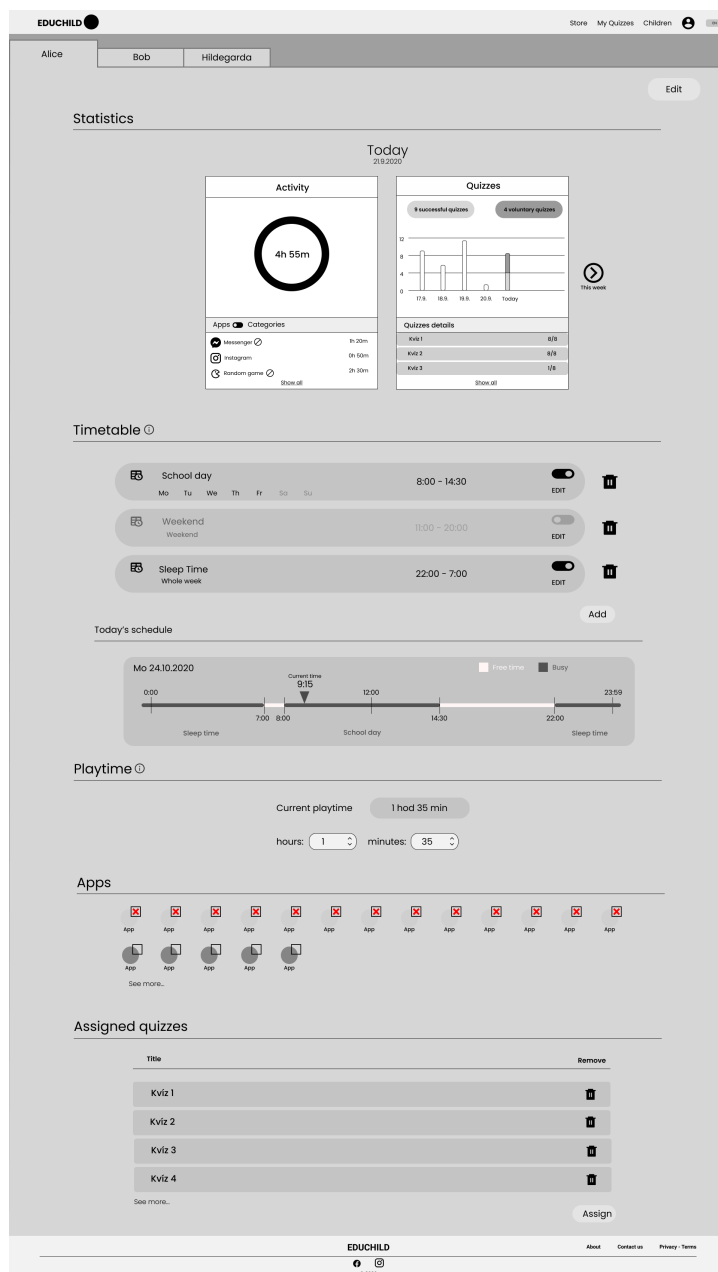
¹¹HyperText Markup Language

¹²JavaScript

¹³Cascading Style Sheets

¹⁴Asynchronous JavaScript and XML

2. NÁVRH



Obrázek 2.4: Návrh obrazovky Děti

míře fungovat i když není dostupné připojení k internetu. Po obnovení konektivity se lokální úložiště synchronizuje se serverem a obnoví se plná funkčnost aplikace.

Další výhodou SPA je rychlejší vývoj a snazší údržba. Z důvodu oddělené architektury (separace backendových služeb a frontendového zobrazení) mohou vývojáři vyvíjet či experimentovat nezávisle na architektuře backendu. Zároveň u případných chyb se relativně rychle zjistí, zda je chyba důsledkem špatného kódu na frontendu či na backendu.

K nevýhodám této architektury jistě patří její problémy s SEO¹⁵. Google a jiné vyhledávače měli velké problémy s indexováním SPA aplikací. Boti nebyli schopni vykreslovat dynamický obsah ani spustit JavaScript a tudíž vše, co v aplikaci viděli, byla prázdná HTML stránka. I když se v posledních letech tento stav poněkud zlepšil, Google sstále přiznává, že ne vždy se podaří SPA správně indexovat [12].

Jinou, ale ne méně závažnou nevýhodou jsou úniky paměti („memory leaks“). Jelikož je SPA de facto jednou stránkou, která se nikdy neobnovuje, tak čím déle aplikace běží, tím horší tyto úniky jsou. Mohou způsobit zpomalení aplikace a také například zvýšit energetické nároky zařízení. Je proto důležité, aby se tomuto tématu věnovala dostatečná pozornost při vývoji, aby se tomuto problému dokázalo předejít.

První načtení SPA aplikace trvá déle než u SSR aplikací. Důvodem pro tento pomalý start je nutnost stahovat objemné JS frameworky, které jsou nezbytné pro hladký chod aplikace. Existují postupy, jak zkrátit délku času potřebného pro start, například minimalizace skriptů či dynamické načítání obrázků.

SPA jsou více zranitelné vůči XSS¹⁶. XSS umožňuje útočnickům vkládat vlastní skripty do aplikací. Uživatel předpokládá, že kód přišel z důvěrného zdroje (v tomto případě z aplikace) a spustí ho. Skript poté může přistupovat ke cookies, tokenům a jiným senzitivním informacím. Skript může navíc přepsat obsah HTML stránky a získat tak od uživatele ještě více privátních informací. Tato zranitelnost je způsobena tím, že prohlížeč má k dispozici veškerou logiku aplikace v JS souborech poslaných ze serveru. Jednou z možností, jak tomuto zabránit nebo alespoň ztížit, je rozdělit JS soubory, které klient obdrží do více izolovaných celků. Útočník poté nebude mít moci přistupovat ke všem zdrojovým kódům najednou.

2.3.1.1 Rekapitulace

Mezi hlavní výhody SPA patří:

- Rychlost aplikace a tedy i rychlé a bezproblémové UI,
- funguje i offline,

¹⁵Search engine optimization

¹⁶Cross-Site Scripting

2. NÁVRH

- rychlý vývoj - separace backendu a frontendu,
- snadná údržba.

Nevýhodami potom jsou:

- Problémy s SEO,
- úniky paměti,
- dlouhý první start,
- zranitelnost vůči XSS.

2.3.2 Universal App

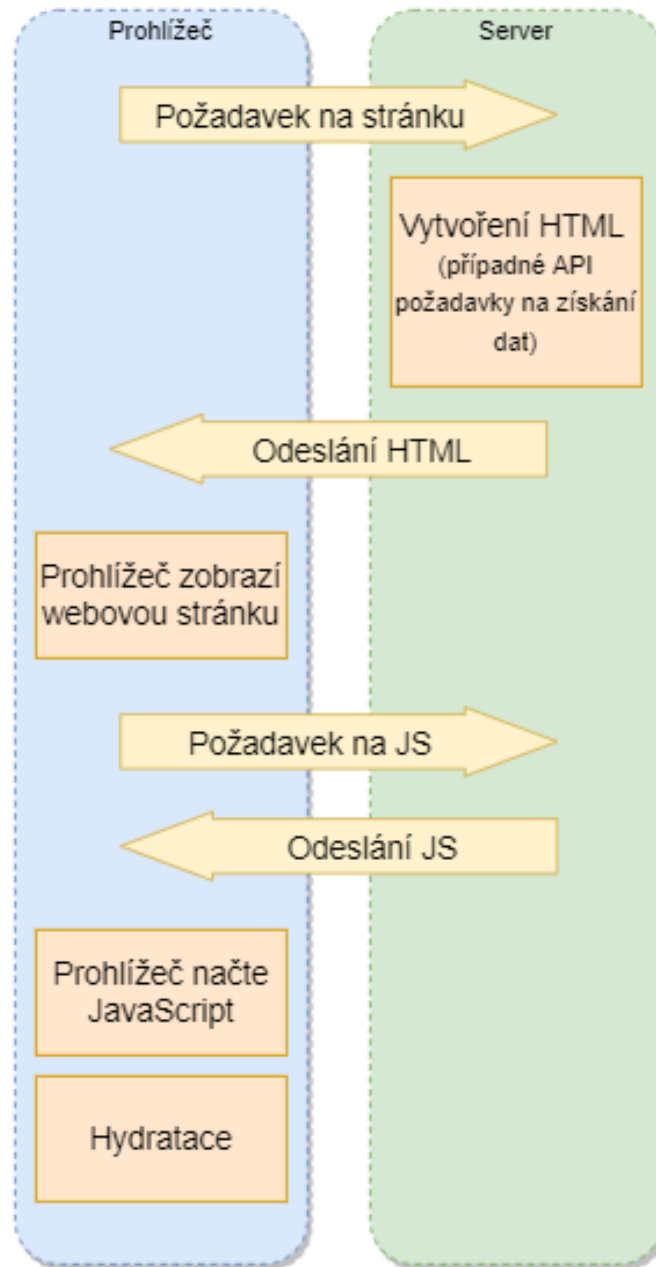
Druhou možností je nechat server, aby vygeneroval stránku. V zásadě se jedná o stejný princip, který využívá tradiční webová aplikace. Několik let zpátky by se dokonce jednalo o naprosto stejný princip. V dnešní době už ale jsou k dispozici řešení, které dokáží využít toho nejlepšího z obou světů. Tedy mít k dispozici výhody SPA a zároveň výhody SSR. Zároveň tím se tím také zbavíme některých nevýhod SPA aplikací. Mimo označení universal se můžeme setkat s označením *isomorphic*.

Universal aplikace funguje tak, že prvotní načtení aplikace se provede na serveru a uživateli se pošle HTML soubor, který se zobrazí v prohlížeči. Poté se odešle potřebný JavaScript. Toto už probíhá na pozadí, uživatel už má v tuto chvíli zobrazenou stránku. JS se poté načte a „hydratuje“ (v angličtině se používá výraz *hydrate*) stránku, tedy aplikace je v tuto chvíli plně funkční a chová se nyní jako SPA aplikace. Tím se neguje jedna z nevýhod SPA aplikací, kterou je pomalé iniciální načtení. Proces je zakreslen na obrázku 2.5[13].

Výhodou je bezesporu lepší provázanost s vyhledávači, kteří nebudou narážet na problémy při indexování aplikace. Plusovým bodem je také fakt, že nemusíme brát v potaz, jak silné je uživatelské zařízení, které bude zobrazovat naši aplikaci. U SPA aplikací toto činí problém, kdy při objemných aplikacích se můžou startovací časy nepříjemně prodloužit. Tím, že obsah generujeme již na serveru a posíláme výsledné HTML, odstraňujeme zátěž na cílové zařízení.

Jedním z handicapů universal aplikace je nutnost obstarat dostatečně výkonný server. V případě oblíbenosti aplikace bude nutné zajistit více serverů pro hladký chod celé aplikace, což se může prodražit.

Další nevýhodou je potom (částečná) nefunkčnost některých JS frameworků na straně serveru. Tvůrci těchto frameworků si tento problém našťěstí v posledních letech uvědomili a poskytli nástroje, jak se mu vyhnout. Přesto je ale důležité být obezřetný, neboť to může snížit rychlost vývoje aplikace.



Obrázek 2.5: Proces startu universal aplikace

2.3.2.1 Rekapitulace

Mezi hlavní výhody universal aplikace patří:

- Rychlé počáteční načtení aplikace,
- dobrá podpora SEO,
- rychlý vývoj - stejný kód vykonávaný na serveru i na klientovi,
- nemusí brát ohled na výpočetní sílu uživatelského zařízení,
- lepší podpora sociálních sítí než SPA.

Nevýhodami potom jsou:

- Cena serverů,
- nefunkčnost některých částí frameworků.

2.3.3 Výběr architektury

Po pečlivém zvážení všech bodů pro a proti jsem se rozhodl aplikaci vyvíjet jako universal. Klíčová pro mě byla dobrá podpora vyhledávačů, neboť pro naši rozvíjející aplikaci je důležité, abychom se potencionálním uživatelům ukazovali co nejčastěji. Zároveň s tím figuruje lepší podpora pro sociální sítě, kdy u SPA aplikací velmi často špatně nebo úplně nefungují náhledy při sdílení stránky. Jelikož nejsem zkušený vývojář v jazyce JavaScript, tak pro mě negativní body jako je zranitelnost vůči XSS a možné úniky paměti mohou představovat problém a to je taky jeden z důvodů, proč jsem nakonec nevybral SPA. Nevýhoda universal aplikací v tom, že potřebují výkonnější servery než jiné typy architektury není v tomto případě tak důležitá, neboť mobilní aplikace EduChild již disponuje jedním takovýmto serverem a pro počáteční běh aplikace bude dostačovat.

2.4 Volba technologií

Před započítím této práce jsem ještě neměl zkušenosti s vývojem v jazyce JavaScript a tedy ani potřebnou znalost některého z jeho mnoha frameworků určeného pro webový vývoj. Proto jsem potřeboval najít framework, který:

- Má dostatečnou podporu, ať už se jedná o dokumentaci či o komunitní fóra,
- má podporu v dostupných IDE¹⁷ (vývojových prostředích),

¹⁷Integrated Development Environment

- je možné s ním snadno vyvíjet universal aplikace,
- umožňuje jednoduše testovat výsledný produkt.

Za možné kandidáty se jevily hlavně frameworky či knihovny Angular od Google [14], React od Facebooku [15] a Vue.js od Evana You [16]. Po prozkoumání daných možností jsem se nakonec rozhodl pro framework Vue.js.

K rozhodnutí mi pomohli hlavně diplomová práce *Frontend skladového systému* [17] od Ing. Oldřicha Malce a poté článek *My experience with Angular 2, React and Vue* [18], jehož autorem je Rajdeep Chandra.

Vue.js vzniklo v roce 2014 jako vedlejší projekt Evana You. Od té doby vzrostlo na popularitě a díky příspěvkům komunity se dnes dokáže měřit s frameworky jako jsou Angular či React. Jednotlivé Vue komponenty se vyvíjí jako tzv. „single file components” (komponenta v jednom souboru), kdy soubory s koncovkou `.vue` jsou rozděleny na tři části – HTML část, script část a jako poslední část pro CSS styly.

Společně s Vue.js budu používat framework Nuxt.js, který se označuje jako „meta framework pro universální aplikace” [19]. Je vystavěn právě nad Vue.js a významně ulehčuje konfiguraci a nastavení universální aplikace. Je využíván aplikacemi jako například GitHub nebo GitLab.

Jako grafickou knihovnu jsem zvolil Vuetify, framework dodržující pravidla Material Designu. Umožňuje snadno tvořit bohaté a poutavé aplikace za použití předem připravených komponent.

2.5 Mockup (Hi-Fi prototyp)

Hi-Fi prototyp (často označovaný jako „mockup”) se liší od Lo-Fi prototypu tím, že se jedná o interaktivní zobrazení, které je co možno nejvíce podobné výslednému produktu, co se týče designu a funkčnosti. Je založen na Lo-Fi prototypu a zkoumá se na něm nejen uživatelské rozhraní jako takové, ale také i uživatelská zkušenost, především co se týče interakcí uživatele se systémem a jestli je chování systému předvídatelné a snadno pochopitelné pro běžného uživatele (volně citováno z [20]).

Typicky tento prototyp obsahuje předem připravená data, neboť v této fázi vývoje ještě není implementováno API či jiná aplikační logika. Úkolem je na tomto prototypu odladit důležité části UI. Pro mě to navíc znamenalo první seznámení s vybranými technologiemi, a také proto mi tato část práce zabrala relativně mnoho času. Výsledkem poté byly prototypy hlavních obrazovek, přihlašovacích a registračních dialogů. Ty poté byly prezentovány zbytku týmu a na základě jejich poznámek a připomínek byl prototyp následně upraven do stávající podoby.

2.6 Návrh API

API¹⁸ můžeme popsat jako prostředníka mezi dvěma aplikacemi, který umožňuje aplikacím mezi sebou komunikovat. V tomto konkrétním případě se jedná o webovou aplikaci a její server se všemi potřebnými daty. Jednotlivé požadavky mají svůj pevně daný formát, který je určen, jehož nesplnění vyústí v chybu.

Pro dorozumívání se serverem byl vybrán REST standart, což znamená, že ke zdrojům (datům) je přístupováno a je možné je měnit skrze HTTP¹⁹ protokol (konkrétně v našem případě přes metody GET, POST, PUT a DELETE) a odpovídající URL²⁰ (webovou adresu).

API pro aplikaci pro Android bylo již ve vývoji v době, kdy jsem byl přijat do týmu, proto jsem se na vývoji zprvu nepodílel. Po vytvoření a následném schválení návrhu uživatelského rozhraní jsem sepsal seznam požadavků, které jsem potřeboval pro to, abych mohl začít implementovat spojení webové aplikace se serverem. Společně s kolegy Petrem Šímou a Milošem Popovičem jsme poté na schůzce seznam ještě jednou prošli a vyjasnili si nedostatky či nedostatečně formulované požadavky. API a server mají v našem týmu na starosti již zmiňovaní kolegové, kteří se tomuto tématu věnují hlouběji ve svých bakalářských pracích.

¹⁸Application Programming Interface

¹⁹Hypertext Transfer Protocol

²⁰Uniform Resource Locator

Implementace

Po kapitole návrhu následuje kapitola věnující se implementaci aplikace. Zde nejprve popíšu, jaké nástroje jsem při vývoji využíval, poté se budu věnovat zajímavým bodům implementace, které podle mě stojí za zmínku. Na závěr popíšu některé problémy, kterým jsem při vývoji čelil a uvedu, které části návrhu se mi podařilo dokončit v době psaní této práce.

3.1 Nástroje

Jak již bylo dříve zmíněno v kapitole 2.4, vybranými technologiemi jsou Vue.js, Nuxt a grafický framework Vuetify. Pro zdárné fungování bylo nejprve potřeba nainstalovat Node.js [21], což je JavaScriptové běhové prostředí. Verze Node.js nesmí být menší než 10.13. Společně s Node.js je při instalaci nutné zahrnout i správce balíčků npm²¹ [22].

Aplikace je vyvíjena na operačním systému Windows ve vývojovém prostředí WebStorm od JetBrains [23]. Pro WebStorm jsem se rozhodl, neboť se s produkty firmy JetBrains setkávám běžně při svém studiu a práci a tudíž mám již představu, jaké funkce mi nabízí a kde je můžu najít.

3.2 Tvorba Nuxt projektu a jeho struktura

Projekt vytvoříme pomocí příkazu `npx create-nuxt-app <jméno projektu>`. Systém se nás zeptá na jméno projektu, dá nám na výběr mezi JavaScriptem a TypeScriptem, dovolí nám vybrat si UI framework podle naší záliby a mnoho dalšího, viz obrázek 3.1. Poté, co dobehne instalace vybraných možností máme vytvořený projekt, který je okamžitě připraven ke spuštění.

Rád bych se zde pozastavil a zmínil strukturu Nuxt projektu. V následujícím seznamu jsou popsány jednotlivé složky a jejich význam.

²¹Node Package Manager

3. IMPLEMENTACE

```
janst@Stejsky-Aspire5 MINGW64 /d/janst/Documents/FIT/BAP
$ npx create-nuxt-app test

create-nuxt-app v3.6.0
[?] Generating Nuxt.js project in test
[?] Project name: test
[?] Programming language: JavaScript
[?] Package manager: Npm
[?] UI framework:
  Bootstrap Vue
  Buefy
  Chakra UI
  Element
  Framevuerk
  Oruga
  Tachyons
> Tailwind CSS
  Windi CSS
  Vant
  View UI
  Vuesax
  Vuetify.js
  None
  Ant Design Vue
(Move up and down to reveal more choices)
```

Obrázek 3.1: Ukázka založení Nuxt aplikace

- **layouts** Složka obsahuje soubory, které popisují rozložení aplikace.
- **pages** Tato složka obsahuje stránky (někdy označujeme jako „views”) aplikace. Mezi těmito stránkami by měl přepínat router, když uživatel naviguje aplikací. V běžné Vue.js aplikaci by to znamenalo vytvořit konfigurační soubor a do něj manuálně přidávat jednotlivé stránky, jak je postupně vytváříme, aby fungovaly v aplikaci přechody mezi stránkami. Nuxt ovšem zajistí, že ke každému .vue souboru je automaticky vygenerována přechodová konfigurace a tudíž se tímto vývojář nemusí zabývat.
- **components** Složka obsahující komponenty. Nuxt zde navíc zajišťuje auto-import komponent do stránek, které jednotlivé komponenty využívají.
- **assets** Zde jsou ukládány CSS styly, obrázky nebo typografické fonty.

- **static** Do této složky patří soubory, které s největší pravděpodobností nebudou již dále měněny. Každý takovýto soubor v tomto adresáři je poté dostupný skrze kořenovou URL.
- **store** Při velkém množství komponent je časté, že mezi sebou komponenty musejí sdílet stav. Pro tento problém je v klasické Vue.js aplikaci k dispozici Vuex, Nuxt ho integruje v základním nastavení. Jedná se o state management pattern, někdo označuje také jako knihovnu, který centralizuje stav aplikace do jednoho místa, ke kterému mají přístup všechny komponenty v aplikaci. Má vlastní pravidla, která zaručují, že případné změny stavu se smí konat pouze skrze mutace, které jsou definovány v souborech v této složce. Tyto mutace poté mění stav aplikace, což způsobí překreslení komponent, který daný stav využívají. Komponenty mohou změnit stav pouze tím způsobem, že vyvolají definovanou akci, která může (ale nemusí) vyvolat požadavek na server a posléze vyvolá mutaci. Vzor tohoto chování je zachycen na obrázku 3.2 [24].
- **plugins** V této složce se nachází soubory, které rozšiřují (přizpůsobují) stažené pluginy.

V poslední řadě se chci zmínit o konfiguračním souboru celého Nuxt projektu `nuxt.config.js`. Úpravou tohoto souboru lze docílit inicializace pluginů, nastavení přídavných modulů a mnoho dalšího. Úprava konfigurace je přímočará, především díky kvalitní dokumentaci [25].

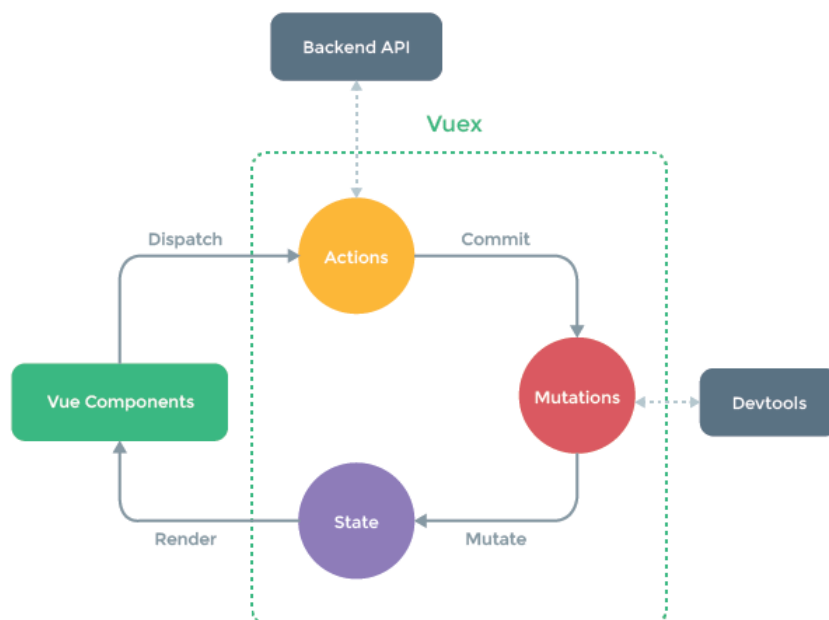
3.3 Nastavení Firebase Auth

Firebase je BaaS²². Jedná se o platformu určenou k usnadnění vývoje mobilních a webových aplikací. Obsahuje několik modulů, kde každý plní rozdílnou funkci (například Firebase Auth se stará o autentizaci uživatelů, Firebase Firestore poskytuje databázi atd.).

Pro Nuxt existuje modul Nuxt Firebase [26]. Po nainstalování pomocí `npm` příkazu je poté potřeba založit projekt na Firebase [27]. Tento projekt tvoří jakýsi kontejner pro veškeré aplikace, které potřebují přístup ke stejným zdrojům. Dalším krokem po založení projektu je založení aplikace v projektu. Po vybrání typu aplikace Firebase vygeneruje konfigurační kód, který je potřeba vložit do Nuxt projektu (viz obrázek 3.3).

V konfiguračním souboru projektu je poté nutné specifikovat, které moduly bude má aplikace inicializovat. Nespecifikované moduly nejsou importovány a tím se snižuje velikost balíčků a tím se zvyšuje výkon aplikace.

²²Backend-as-a-Service



Obrázek 3.2: Vuex pattern

3.4 Axios a API

Axios je HTTP klient, který se stará o odesílání a přijímání HTTP požadavků. Stejně jako Firebase má vlastní modul pro framework Nuxt [28]. Tento modul je využíván pouze pro posílání požadavků na privátní API (Firebase do aplikace přináší vlastní API).

Přístup k tomuto privátnímu API je chráněn skrze definované hlavičky X-Authozation-Firebase a X-API-KEY. Pro snadné vložení těchto hlaviček jsem využil jednu z pomocných funkcí modulu a tou je funkce `onRequest`, která se zavolá při každém odchozím požadavku. Zde poté kontroluji, zda je uživatel přihlášen a pokud je, tak vložím do hlavičky jeho token, který je uložen ve Vuex modulu. Ukázka je k dispozici na obrázku 3.4.

Po nastavení Axios jsem čelil dalšímu problému a to, jak implementovat volání na API. Rozhodl jsem se, že začnu tím, že implementuji základní třídu `Service`. Třída obsahuje 4 metody odpovídající nejpoužívanějším typům HTTP požadavků: GET, POST, PUT a DELETE. Specializované servisní třídy poté zavolají metody základní třídy s danou URL a případnými nutnými daty. Výsledek je vidět na obrázku 3.5.

```
firebase: {
  config: {
    apiKey: '<api_key>',
    authDomain: '<auth_domain>',
    databaseURL: '<database_url>',
    projectId: '<project_id>',
    storageBucket: '<storage_bucket>',
    messagingSenderId: '<messaging_sender_id>',
    appId: '<app_id>',
    measurementId: '<measurement_id>',
  },
  services: {
    auth: {
      persistence: 'local',
      initialize: {
        onAuthStateChangedAction: 'Auth/onAuthStateChangedAction',
        subscribeManually: false,
      },
      ssr: true,
    },
  },
},
```

Obrázek 3.3: Ukázka konfigurace Firebase

Další věc, se kterou jsem nebyl ještě spokojen, byla nutnost importu do jednotlivých komponent. Proto jsem využil možnosti frameworku Nuxt a provedl inject („vlození“) jednotlivých servisních tříd do objektu context. Tento objekt je k dispozici napříč celou aplikací a obsahuje dodatečné objekty například objekt app, který obsahuje instance všech pluginů. Tímto jsem zpřístupnil servisní třídy napříč celou aplikací bez nutnosti používat import direktivu v každém souboru. Na plugin je možno nahlédnout na obrázku 3.6.

```
export default ({ $axios, store }) => {
  $axios.onRequest( callback: (config : AxiosRequestConfig ) => {
    if (
      store.state.Auth.user &&
      !config.headers.common['X-Authorization-Firebase']
    ) {
      $axios.setHeader(
        name: 'X-Authorization-Firebase',
        value: `${store.getters['Auth/getAuthorizationToken']}`
      )
    }
    $axios.setHeader( name: 'Content-Type', value: 'application/json')
    $axios.setHeader( name: 'X-API-KEY', value: '<api_key>')
    config.paramsSerializer = (params) => {
      return qs.stringify(params, options: { arrayFormat: 'repeat' })
    }
  })
}
```

Obrázek 3.4: Ukázka kódu metody onRequest

3.5 Překlady

Dalším úkolem byla implementace podpory více jazyků. Použil jsem k tomu modul i18n pro Nuxt [29]. Ten umožňuje zapisovat překlady do JavaScriptových souborů nebo do JSON souborů.

I18n umožňuje specifikovat překládat i množná čísla, je tedy velmi sofistikovaný. Přesto pro přeložení velkého kusu textu se mi jeví jako nevhodný a proto jsem se rozhodl, že tento modul budu používat převážně pro překlady malého rozsahu například překlady notifikací, formulářových polí a popisků.

Pro překlady velkého textového rozsahu jako například informace na úvodní stránce budu využívat CMS²³ Nuxt Content. Pomocí Markdown souborů můžu zobrazit jakýkoliv obsah, dokonce mohu využívat i moje vlastní Vue komponenty. Tento CMS sice neposkytuje žádné možnosti překladu, ale hodlám to řešit způsobem, kdy pro každý jazyk bude k dispozici množina Mar-

²³Content Management System

```

export class Service {
  constructor($axios, resource) {
    this.$axios = $axios
    this.resource = resource
  }

  get(url, config: {} = {}) {
    return this.$axios.get( url `/${this.resource}/${url}`, config)
  }

  put(url, payload, config: {} = {}) {
    return this.$axios.put( url `/${this.resource}/${url}`, payload, config)
  }

  post(url, payload, config: {} = {}) {
    return this.$axios.post( url `/${this.resource}/${url}`, payload, config)
  }

  delete(url) {
    return this.$axios.delete( url `/${this.resource}/${url}`)
  }
}

```

```

import { Service } from '../Service'

export class ParentService extends Service {
  constructor($axios) {
    super($axios, resource: 'parents/me')
  }

  async getChildren() {
    try {
      const children = await super.get( url: 'children/' )
      return children._embedded ? children._embedded.children : []
    } catch (e) {
      console.log(e)
      return []
    }
  }

  getParent() {
    return super.get( url: '' )
  }

  addChild(child) {
    return super.post( url: 'children/', payload: {
      ...child,
    })
  }

  updateParent(parent) {
    return super.put( url: '', payload: { ...parent })
  }

  deleteParent() {
    return super.delete( url: '' )
  }
}

```

Obrázek 3.5: Ukázka základní a specializované servisní třídy

kdown souborů a na základě vybraného jazyka zobrazím daný soubor.

3.6 GitLab CI/CD a Sentry

CI²⁴ je automatizace integrace změn v kódu aplikace napříč více vývojáři. Kód se automaticky sloučí do jednoho centrálního úložiště, kde je poté kód podroben testům [30]. CD²⁵ je proces vypouštění nového softwaru, který využívá automatizované testy k ověření funkčnosti kódu a poté je daný software automaticky vypuštěn na produkční prostředí [31].

Prozatím využíváme CI/CD funkcionalitu, kterou nabízí GitLab [32]. S nastavením této funkcionality mi pomohl Miloš Popovič, který již dříve nastavoval CI/CD pro backend. Po každém přidání kódu do definované větve v repozitáři aplikace se provede kompilace aplikace a pokud je úspěšná, nasadí se pomocí programu Docker [33] na testovací server.

Každá aplikace by měla mít implementované zapisování chyb, neboť to výrazně ulehčí vývojářům najít chybný kus kódu, když mají k dispozici zápisy

²⁴Continuous Integration

²⁵Continuous Deployment

3. IMPLEMENTACE

```
export default ({ app: { $axios } }, inject) => {
  const serviceMap = new Map()
  serviceMap.set('QuizService', new QuizService($axios))
  serviceMap.set('ApplicationService', new ApplicationService($axios))
  serviceMap.set('CategoryService', new CategoryService($axios))
  serviceMap.set('ParentService', new ParentService($axios))
  serviceMap.set('ParentCreationService', new ParentCreationService($axios))
  serviceMap.set('QuestionService', new QuestionService($axios))
  serviceMap.set('TimetableService', new TimetableService($axios))
  serviceMap.set('ImageService', new ImageService($axios))
  serviceMap.set('ChildService', new ChildService($axios))
  serviceMap.set('AuthorService', new AuthorService($axios))
  serviceMap.set('AuthService', new AuthService())
  serviceMap.set('AccountService', new AccountService($axios))
  serviceMap.set('LanguageService', new LanguageService($axios))

  for (const [key, value] of serviceMap.entries()) {
    inject(key, value)
  }
}
```

Obrázek 3.6: Ukázka kódu pro inject servisních tříd do context

o chybě, o tom kde a za jakých podmínek nastala. Aplikaci jsem proto pomocí modulu Sentry [34] napojil na stejnojmenný logovací systém [35]. Nastavení bylo přímočaré, v projektu stačilo přidat konfigurační objekt do souboru `nuxt.config.js` a aplikace od té doby loguje chyby, které nastanou při jejím běhu.

3.7 Problémy při implementaci

3.7.1 Persistence přihlášených uživatelů a přesměrování

Pokud chceme ochránit některé stránky od neautentizovaných uživatelů, musíme vytvořit middleware. To je soubor funkcí, které se spustí před vykreslením stránek. Takovýto middleware jsem vytvořil, bohužel jeho fungování se nedalo popsat jako správné. Při každém obnovení stránky došlo k navrácení uživatele na domovskou stránku. Middleware se totiž spouští na serveru, který ovšem

nemá ponětí o přihlášeném uživateli, respektive nelze na něm využít Firebase modul.

Tento problém se mi podařilo vyřešit až tím, že při každé změně autentizace uživatele je do cookie uložen uživatelský token. Ke cookies již server přístup má a proto dokáže autentizovat uživatele i bez přístupu k Firebase modulu.

3.8 Dokončená část

Bohužel se mi nepodařilo vyvinout aplikaci v plném rozsahu, který je popisován v kapitolách 1 a 2. Hlavním z faktorů je ten, že v oblasti vývoje webových aplikací je toto moje první aplikace a mnoho věcí jsem se musel nově naučit. To mělo za následek, že po kontrole komponent, které vznikly na počátku, jsem musel přistoupit k refaktoringu, neboť byly moc složité a často byly špatně rozšiřitelné. Nicméně vytvořená aplikace splňuje zadání v tom, že se jedná o funkční prototyp, na jehož základech se dá dobře budovat v dalším vývoji aplikace.

Poznámka: Nebudu se v následujícím textu vyjadřovat k designu celé aplikace, neboť bylo v týmu dohodnuto, že tento úkol bude následovat až toto léto. Nynější vzhled aplikace je proto pouze „prototyp“ a nelze ho považovat za finální verzi.

Nicméně i přesto jsem dokázal vyvinout prototyp v základní podobě, s víceméně kompletní funkcionalitou, která je popsána v dřívějších kapitolách. V oblasti autentizace se jedná o komponenty přihlášení, přihlášení přes Google účet, registrace a komponentu zapomenutého hesla.

Co se týče tvorby a editace kvízů je tato oblast z většiny hotová. Chybí komponenta pro zobrazení aktuálního kvízu na mobilním rozhraní a zatím nelze uložit editovaný kvíz jako nový kvíz s jiným jménem. Mimo tyto výjimky je ovšem tato část hotová.

V oblasti obchodu je funkční zobrazení populárních a doporučených kvízů, stejně jako tak je funkční vyhledávání. Karta detailu kvízu a detailu otázek je také připravena. Co zatím v této oblasti není dokončené, je nahlašování kvízů, neboť na to dosud není hotový API endpoint (také plánován na léto), stejně tak je na tom doporučování kvízů dětem na kartě kvízu.

Část správy dětí obsahuje funkční komponenty zobrazení, tvorby, editace a mazání kvízu. S tím se pojí i panel úpravy volného času. Připraven je také seznam přiřazených kvízů a jejich přiřazování. Editace dítěte a jeho mazání je rovněž připravena. Chybí ovšem celá komponenta statistik dítěte.

Stránka Nastavení uživatelského účtu je připravená v celé předem definované podobě, kromě části detailů uživatele, kdy po dohodě s týmem zůstal pouze checkbox pro změnu role uživatele, protože se prozatím v aplikaci nepočítá s možností měnit si svoje jméno a email, popřípadě profilovou fotku, neboť se s těmito daty prozatím nijak npracuje.

3. IMPLEMENTACE

Domovská stránka je prozatím „holá“ z důvodu, že zatím nebylo řešeno, co bude zobrazeno a není proto ani vymyšlen text.

Zdrojový kód práce je dostupný na <https://gitlab.fit.cvut.cz/stejsj26/educhild-web>.

Testování

V této kapitole popíší průběh uživatelského testování, jakým způsobem probíhalo a jaké jsme z něj vyvodili důsledky. Na závěr poté shrnu změny, které byly provedeny na základě výsledků testování.

Důvodem, proč bylo provedeno pouze uživatelské testování, byl nedostatek času, který mi v této fázi práce zbýval. Bylo tedy na mě se rozhodnout, kterému způsobu testování (uživatelské vs. testování funkčnosti prototypu) dám přednost. Většinu času ve fázi implementace zabralo vývoj samotného vzhledu prototypu a menší část byla strávena připojením k API. Jelikož API je sdílené mezi mobilní i webovou aplikací, probíhá již základní testování na mobilní aplikaci. Z tohoto důvodu jsem se nakonec přiklonil k provedení uživatelského testování, abych i poslední část aplikace podrobil testu. Více se k tomuto tématu vyjadřuji v kapitole 5.

4.1 Co je uživatelské testování

Proces uživatelské testování spočívá v tom, že pozorujeme jednu osobu (testera) při jeho interakci s aplikací, prototypem či jenom designovým návrhem. Tester při této interakci provádí běžné úkoly spojené s danou aplikací či daným prototypem.

Tento proces můžeme přirovnat situaci, kdy provádíme skupinu turistů našim městem. Turisté si budou všimnout více věcí, i těch, které vy osobně běžně přehlídíte. Mají totiž na vaše město jiný úhel pohledu a věci, které se vám zdají přirozené a zřejmé, se jim mohou jevit jako nepřirozené a nepochopitelné [36].

Úkolem tohoto typu testování je odhalit slabiny a chyby v návrhu uživatelského rozhraní prototypu jako například špatné rozmístění ovládacích prvků nebo absence vysvětlujících popisků.

4.2 Způsob testování

Testování probíhalo ve dnech 3.-4. května 2021 v domácím prostředí u testerů doma. K dispozici byl připravený notebook se spuštěnou webovou aplikací. Společně s testerem a mnou byla přítomna i Barbora Kyselová, která se na testování podílela.

Během testování jsem byl testerům k dispozici jako moderátor, tedy když se v aplikaci zasekli a nevěděli, jak pokračovat, tak jsem poradil, kam se vrátit, popřípadě co udělat. Zároveň Barbora dělala poznámky k průchodu scénářů jednotlivými testery. U prvního testera jsme měli role prohozené, tedy, že Barbora byla v roli moderátorky a já dělal poznámky, ale v dalších instancích testů jsme se rozhodli si naše role prohodit, neboť jsme společně po prvním testu vyhodnotili, že Barbora umí lépe zaznamenávat informace o testu a já naopak lépe komunikuji s testery.

Testování u jednotlivých testerů probíhalo vždy ve stejném formátu. Začalo se krátkým seznámením testera, co je cílem aplikace a k čemu slouží toto testování, neboť ani jeden z testerů se nikdy předtím na testování softwaru nepodílel. Následně byl testerům předložen jeden ze šesti připravených scénářů (viz obrázek 4.1, také dostupné v příloze D).

Testovací scénář 1

Registrace

Uvedení do děje:

Jste rodičem dvou dětí ve školním věku. Aniče je 12 let, Bořkovi bude brzy 9 let. Děti tráví většinu dne na mobilu a kdykoliv se je pokusíte přesvědčit, aby si zkusili jít hrát jiným způsobem, skončí to nezdarem.

Od známých se dozvíte o jisté aplikaci, která by Vám v tomto směru mohla pomoci. Aplikace blokuje volný čas, který děti na mobilech tráví. Přifazuje jim k vyplnění znalostní kvízy a podle jejich úspěšnosti jim poté zpřístupňuje ostatní aplikace, na kterých tráví svůj čas běžně. Rozhodnete se, že tuto aplikaci vyzkoušíte. Stáhnete ji svým dětem do mobilů a během používání zjistíte, že aplikace také disponuje webem, prostřednictvím kterého můžete snadněji spravovat mobilní aktivitu dětí. Rozhodnete se jej také vyzkoušet.

Úkoly:

- Registrujte se ve webové aplikaci

Obrázek 4.1: Testovací scénář pro registraci

Tyto scénáře obsahují krátkou přípravu uvedení do děje, aby se tester dokázal vžít do situace, kterou se snažíme nasimulovat. Na konci scénáře je poté stručně popsán úkol, který po testerovi chceme splnit. Stručně je zde klíčové slovo. Nechceme úkol popisovat příliš podrobně, abychom testerovi neúmyslně neporadili a tím ovlivnili pozdější výsledky testů. Případnou nápovědu od moderátora dostane tester až v tom případě, že je viditelně ztracen a neví, jak pokračovat v plnění úkolu.

Po absolvování všech scénářů dostal tester na výběr, zda chce sám vyplnit Dotazník po testování (viz příloha E) či zda si přeje o průběhu testu s námi diskutovat a my si jeho odpovědi zaznamenáme.

4.3 Testeři

Z důvodu probíhající pandemie onemocnění Covid-19 [37] byli jako testeři vybráni lidé z mého blízkého okolí, kteří ale patří do cílových skupin. Celkově byla aplikace otestována čtyřmi testery.

4.3.1 První tester

- Jméno: Eva S., 49 let
- Povolání: vrchní sestra
- Děti: dvě dospělé děti, dva synovci ve školním věku (13 a 10 let)

Tester má vysokoškolské vzdělání, vlastní děti nejsou cílovou skupinou pro aplikaci, ale synovci jsou v cílové skupině. Tester je řadovým uživatelem PC.

4.3.2 Druhý tester

- Jméno: Libor S., 52 let
- Povolání: radiologický asistent
- Děti: dvě dospělé děti, dva synovci ve školním věku (13 a 10 let)

Tester má vysokoškolské vzdělání, vlastní děti nejsou cílovou skupinou pro aplikaci, ale synovci jsou v cílové skupině. Tester je řadovým uživatelem PC.

4.3.3 Třetí tester

- Jméno: Milena F., 49 let
- Povolání: učitelka ZŠ
- Děti: dvě dospělé děti

Tester má vysokoškolské vzdělání. Tester je řadovým uživatelem PC. Jako učitelka je jednou z možných skupin, na které je aplikace zaměřena.

4.3.4 Čtvrtý tester

- Jméno: Lucie S., 28 let
- Povolání: učitelka ZŠ
- Děti: dívka 0,5 roku

Tester má vysokoškolské vzdělání. Tester je řadovým uživatelem PC. Jako učitelka je jednou z možných skupin, na které je aplikace zaměřena.

4.3.5 Nielsenova křivka

Mohlo by se zdát, že uživatelské testování pouze se čtyřmi uživateli není příliš vypovídající o kvalitě návrhu uživatelského rozhraní prototypu. Ovšem podle Jakoba Nielsena [38] není potřeba pro uživatelské testování velký počet osob. Společně s Thomasem Landauerem zkoumali počet nalezených chyb při uživatelském testování napříč širokou škálou projektů a nakonec přišli s následující rovnicí:

$$N(1 - (1 - L)^n)$$

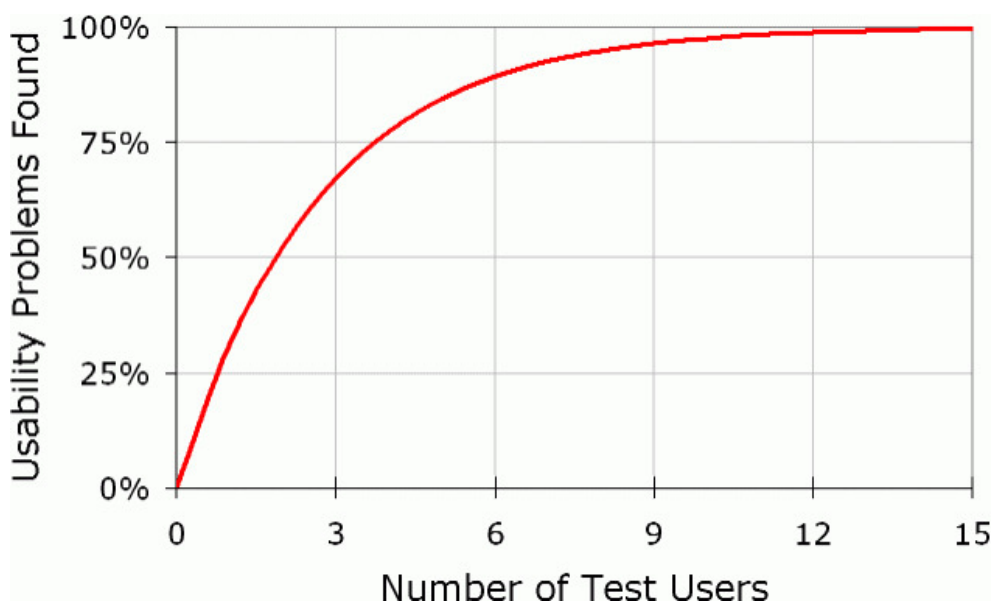
Zde N označuje celkový počet chyb v uživatelském návrhu a L označuje poměr objevených chyb při testování jednoho uživatele. Běžná hodnota L je rovna 31%, jak bylo zjištěno napříč širokou škálou testovaných subjektů. Hodnota n určuje počet uživatelů, kteří se na testu podílejí. Při zakreslení do grafu nám vznikne křivka jako na obrázku 4.2, kde osa y představuje hodnotu N a hodnota n je zakreslena na ose x .

Jak je vidět, již při testování jedním uživatelem výrazně stoupne počet odhalených chyb. Při testování dalších uživatelů se odhalují další chyby, ale již ne tak často, neboť dochází u uživatelů k částečnému překryvu, kdy více uživatelů například objeví stejný nedostatek. To znamená, že s narůstajícím počtem uživatelů se dozvídáme o čím dál tím méně nových problémech.

Při počtu uživatelů $n = 4$ a hodnotě $L = 31\%$ dostáváme následující rovnici

$$N(1 - (1 - 0,31)^4) \doteq N(0,773)$$

Z toho vyplývá, že při tomto uživatelském testování se podařilo objevit přibližně 77% chyb v návrhu uživatelského rozhraní. Důležité je si uvědomit, že po změně uživatelského rozhraní v důsledku tohoto testování bude nutné nový návrh podrobit dalšímu uživatelskému testování, aby se přišlo na nové chyby.



Obrázek 4.2: Nielsenova křivka

4.4 Shrnutí poznatků z testování

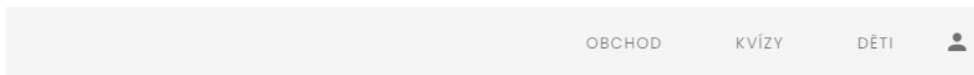
- **Scénář 1 - Registrace**

Tento scénář (viz 4.1) se setkal s problémy u každého testera. Původní návrh rozhraní umístil tlačítko Registrace na modální okno, které se otevře kliknutím na tlačítko Přihlášení. Úvaha za tímto rozhodnutím byla taková, že veškeré akce související s autentizací budou shluknuté na jednom místě a bude tak jednoduché pro uživatele vědět, kde tyto akce najít. Tato úvaha se ukázala jako chybná. Testeři sice po chvíli uvažování skutečně vybrali tlačítko přihlašování, ale už si neuvědomili, že je potřeba kliknout následně na tlačítko otevírající okno registračního formuláře. To poté vedlo k situacím, kdy testeři vyplňovali přihlašovací formulář, i když ještě neměli vytvořený účet a vedlo to poté k nutnému zásahu moderátora.

Řešení Z diskuzí s testery po testování se došlo k závěru, že bude nutné přidat samostatné tlačítko na vyvolání registračního formuláře přímo do navigační lišty aplikace. Tím se oba úkony od sebe viditelně oddělí a uživatelé nebudou muset přemýšlet, jestli se registrace schovává pod tlačítkem přihlášení.

- **Scénář 2 - Získání kvízů**

Tento scénář byl zaměřen na stránku Obchod, kde může uživatel zdarma získat kvízy a po získání kvízu na jeho následné přiřazení dítěti. Ze všech scénářů byl nejvíce problémový. Hlavním důvodem pro tuto skutečnost byl fakt, že popisky v navigační liště byly příliš obecné jako *Obchod* či *Kvízy* (viz obrázek 4.3) a uživatelé nevěděli, co si (zejména pod pojmem Obchod) představit.



Obrázek 4.3: Výřez navigační listy v době testování

Důsledkem toho se poté uživatelé snažili získat nové kvízy na stránce Kvízy, tato stránka ale slouží pro již získané kvízy nebo pro tvorbu nových. Po pomoci od moderátora se správně přesunuli na stránku Obchodu, kde jim již vyhledání a výběr adekvátního kvízu nečinilo potíže. Po otevření detailu kvízu byla většina testerů zmatená, neboť nemohli najít, tlačítko stáhnutí, které nebylo nijak barevně odlišeno a nemělo textový popis. To činilo obtíže především „starší“ generaci uživatelů, kteří nejsou tolik zvyklí na ovládací prvky bez popisku.

Řešení Z diskuzí se později usneslo, že popisky v navigační liště doznají úprav, aby byly více konkrétní, vysvětlující (např. Obchod se změní na

Kvízy ke stažení, Kvízy se přejmenují na Moje kvízy atd.). Další změnu dozná karta kvízu, kde se tlačítko stáhnutí přesune k ostatním tlačítkům do spodní části karty a k tlačítku bude přidán popisek a vhodná odlišující barva.

- **Scénář 3 - Tvorba kvízu**

Při tomto scénáři nedocházelo k větším problémům, uživatelé na stránce kvízů správně lokalizovali tlačítko tvorby kvízů a bez větších obtíží kvíz vytvořili. Malé problémy, kterých jsme si všimli, jsou poté políčka pro vyplnění jména a popisu kvízu. Testeři se nás v několika případech ptali, zda jako jméno mají použít jméno jednoho z dětí, kterému bude kvíz přiřazen. Podobný problém nastal u popisu, kam nevěděli, co přesně mají napsat.

Dále jsme vyzorovali občasné malé zaváhání při výběru typu otázky. Bylo zjevné, že uživatelé si při výběru neuměli představit, jak některé typy fungují. Posledním malým problémem byla poté tvorba odpovědi u otevřených typů otázek, kde někteří testeři vymysleli otázku, na kterou chtěli zadat víceslovnou odpověď. To ovšem aplikace v nynější verzi ne-

umožňuje, u otevřených typů otázek je nutné odpovědět jedním slovem nebo číslem

Řešení Z výsledků tohoto scénáře jsme určili nutnost změn u některých popisků a také přidání popisků nových. Konkrétně ve formuláři informací o kvízu bude kolonka Jméno přejmenována na

Jméno kvízu a kolonka Popis dostane vysvětlivku, co se po uživateli chce vyplnit. Zároveň dojde k přidání popisků k jednotlivým typům otázek, které vysvětlují jak daný typ funguje. Poslední úprava v závislosti na tomto scénáři se týká přidání varování pro otevřené typy otázek, že je možné vyplnit pouze jedno slovo či číslo.

- **Scénář 4 - Správa času**

Vytvoření rozvrhu a upravení délky volného času byla pro většinu testerů složitá záležitost, pouze jeden z testujících subjektů splnil úkol bez nutnosti zásahu od moderátora. Testeři totiž nevěděli, co přesně znamená pojem volný čas a rozvrh. Při návrhu rozhraní se brala v potaz skutečnost, že většina nových uživatelů se nejdříve s danými pojmy seznámí při používání mobilní aplikace, to ovšem neplatilo pro testery v tomto testování. Jednalo se tedy především o naši chybu při tvorbě scénářů, že jsme zapomněli vysvětlit důležité pojmy.

Řešení Kromě nutnosti upravit scénáře pro budoucí iteraci testování jsme došli k závěru, že bude vhodné doplnit jednotlivé sekce na stránce Správa dětí o krátký popis, k čemu vlastně tyto sekce slouží.

- **Scénář 5 - Odebrání kvízů**

Odebrání přiřazených kvízů odhalilo jeden zásadní problém. Většina testerů se snažila odebrat kvízy pomocí tlačítka pro kompletní odstranění kvízu na stránce Moje kvízy, na což je musel včas upozornit moderátor. Pouze jeden tester bez problému odebral kvízy z karty dítěte na stránce Správa dětí. Tato inkonzistence – některé akce dostupné na stránce Moje kvízy a některé zase pouze dostupné ve Správě dětí – představuje problém, který je potřeba vyřešit.

Řešení Komponentu akcí s kvízy bude nutné přepracovat, aby umožnila i odebrání kvízů na stránce Moje kvízy.

- **Scénář 6 - Editace kvízu** Tento scénář byli schopni všichni testeři splnit bez větších problémů. Pouze jeden tester hledal tlačítko editace na stránce Správa dětí, kde tato akce není dostupná. Opět se tak vracíme k předchozímu bodu, kterým je inkonzistence dostupných akcí v závislosti na které stránce se uživatel nachází.

Řešení Stejně jako v předchozím bodu dojde k přepracování komponenty akcí, aby umožňovala více akcí na stránce Správa dětí.

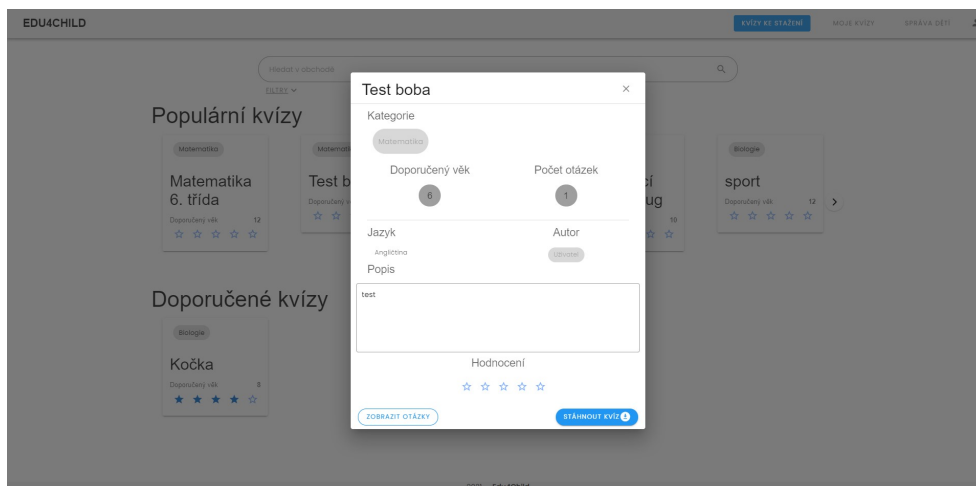
4.4.1 Provedené změny a ukázky výsledného systému

Konkrétně jsem po testování provedl následující změny v prototypu:

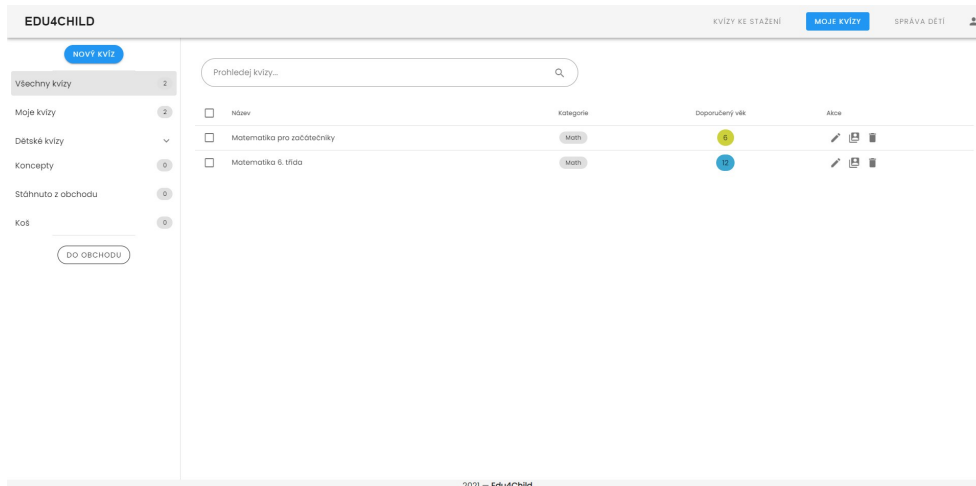
- Přidání samostatného tlačítka registrace na systémovou lištu aplikace (viz Scénář 1).
- Změnu popisků tlačítek v systémové liště, aby lépe popisovali, k čemu slouží (viz Scénář 2).
- přidání popisku a barvy tlačítka stáhnutí na kartě detailu kvízu (viz Scénář 2 a obrázek 4.4).
- Přidání vysvětlujících popisků k jednotlivým typům otázek a přejmenování textových polí na formuláři tvorby kvízu (viz Scénář 3).
- Sekce na stránce Správa dětí nyní obsahují krátkou informaci, k čemu mají sloužit (viz Scénář 4).
- V záložkách dětských kvízů na stránce Moje kvízy lze nyní kvízy rovnou odebírat (viz Scénář 5).
-

Po těchto změnách jsem dospěl k výsledné verzi prototypu a realizoval tím výslednou aplikaci. Ukázky výsledné aplikace jsou viditelné na obrázcích 4.4 a 4.5. Více ukázek je dostupných v příloze F.

4.4. Shrnutí poznatků z testování



Obrázek 4.4: Detail kvízu s upraveným tlačítkem stažení



Obrázek 4.5: Stránka Moje Kvízy

Budoucí vylepšení

Jak jsem již popisoval dříve v kapitole 3.8, prototyp se nepodařilo vyvinout v kompletní podobě, která byla zachycena v kapitole Návrhu.

Proto jedním z nejdůležitějších bodů budoucího vývoje bude dokončení aplikace v celém plánovaném rozsahu, což znamená například vývoj komponenty pro ukazování kvízů (jak budou vypadat v mobilní aplikaci) či přidání dětských statistik.

Další, stejně důležitý bod, je přidání Unit testů. Do finální verze aplikace jsem je z časové lhůty nestihl přidat a je to něco, co je potřeba napravit ještě před prvotním vypuštěním aplikace. Na obrázku 5.1 je vidět ukázka jednoho takového Unit testu, který vznikl za pomoci frameworku Jest [39].

Dále bych si přál do aplikace přidat podporu nočního (tmavého) módu. Je běžné, že moderní aplikace nabízejí uživatelům možnost si vybrat alespoň z těchto dvou módů. Důvod, proč nebyla podpora nočního módu zahrnuta již v prototypu, je ten, že aplikace ještě nemá hotový návrh grafického designu. Ten bude po dohodě s kolegy probíhat v létě 2021.

Přidání možnosti přihlášení přes Facebook je dalším krokem k vylepšení komfortu uživatele. Aplikace již podporuje přihlášení skrze uživatelem vytvořený účet nebo přihlášení přes Google účet.

Pro rychlý budoucí vývoj bude nutné vytvořit kvalitní dokumentaci jednotlivých komponent a servisních tříd, ve které budou zachyceny informace o jednotlivých „eventech“, které komponenty spouštějí.

5. BUDOUCÍ VYLEPŠENÍ

```
import { mount, createLocalVue } from '@vue/test-utils'
import Vuetify from 'vuetify'
import PlaytimePanel from '../components/children/playtime/PlaytimePanel'

describe( name: 'PlaytimePanel.vue', fn: () => {
  const sampleData = {
    childObject: {
      name: 'Anička',
      playTimePerDay: 65,
      dateOfBirth: '2009-05',
      autoBlockingAfterInstallation: true,
      blockSocialCategoryByDefault: true,
      blockGameCategoryByDefault: true,
      modified: new Date(),
    },
  },
}

let playtimePanel
beforeEach( fn: () => {
  const vuetify = new Vuetify()
  const localVue = createLocalVue()

  playtimePanel = mount(PlaytimePanel, options: {
    mocks: {
      $t: (i) => i,
      localePath: (i) => i,
    },
    localVue,
    vuetify,
    propsData: sampleData,
  })
})

it( name: 'Playtime renders correctly', fn: () => {
  expect(playtimePanel.findComponent({ ref: 'playTime' }).text()).toMatch(
    expected: '01:05'
  )
})
})
```

Obrázek 5.1: Ukázka jednoduchého Unit testu pro komponentu zobrazení volného času

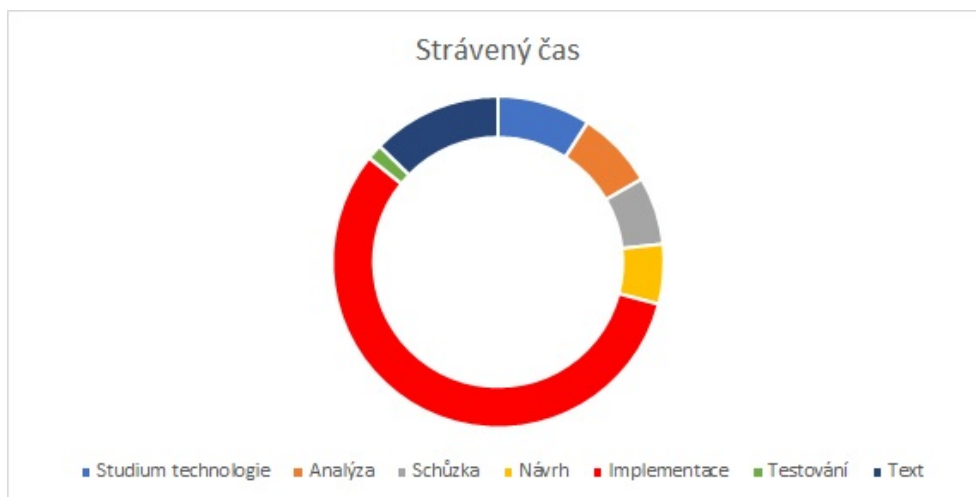
Závěr

Cílem práce bylo vyvinout prototyp webové aplikace pro podporu mobilní aplikace EduChild. Cíl práce se podařilo splnit a nyní mám k dispozici stabilní prototyp, který se dá v budoucnu rozvíjet.

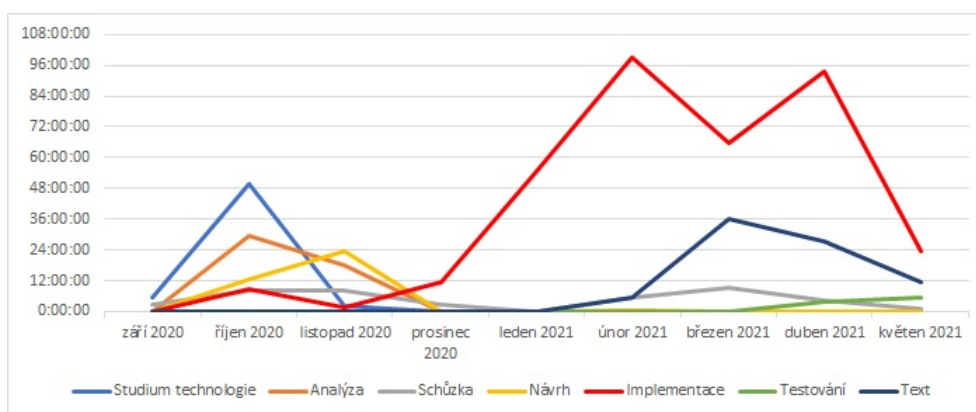
Z hlediska osobního rozvoje pro mne tento projekt představuje obrovský přínos. Během vývoje prototypu jsem se musel naučit pracovat s jazykem JavaScript a vyzkoušel si vývoj s frameworkem Vue.js. Tento framework jsem si oblíbil a chtěl bych s ním pracovat i nadále. Vyzkoušel jsem si od počátku návrh aplikace pro reálné použití a zjistil jsem, že ačkoliv studuji již 4 roky softwarové inženýrství, tak jsem se při vývoji neustále musel naučit nové znalosti a metodiky v oblasti vývoje aplikací. Naučil jsem se efektivně pracovat s novými nástroji jako je například Figma nebo Clockify [40]. V nástroji Clockify jsem si zapisoval čas, strávený činností na této práci. Jeho souhrny jsou k dispozici na obrázcích 5.2 a 5.3.

Je zde vidět, že čas strávený analýzou a návrhem je zastíněn časem stráveným implementací. To je hlavně kvůli tomu, že návrh jsme prováděli společně s Barborou Kyselovou a v důsledku toho jsme toho stihli více vytvořit za kratší čas. Moje absence předchozích zkušeností s vývojem webové aplikace se poté odrazila na množství času stráveného učením se novým technologiím a samotnou implementací.

Testování nezabírá mnoho času, neboť kvůli epidemiologické situaci nebylo tak rozsáhlé, jak bych si přál. Před zveřejněním aplikace určitě bude muset proběhnout mnohem extenzivnější testování, nejen na úrovni uživatelského rozhraní, ale také testování funkčností aplikace.



Obrázek 5.2: Graf stráveného času vývojem prototypu část 1



Obrázek 5.3: Graf stráveného času vývojem prototypu část 2

Literatura

- [1] FURPS model. [online], 2021, [cit. 2021-03-20]. Dostupné z: <https://en.wikipedia.org/wiki/FURPS>
- [2] Metoda FURPS. [online], 2019, [cit. 2021-03-20]. Dostupné z: <https://cs.wikipedia.org/wiki/FURPS>
- [3] Google Forms. [online], 2021, [cit. 2021-03-20]. Dostupné z: <https://www.google.com/forms/about/>
- [4] Microsoft Forms. [online], 2021, [cit. 2021-03-20]. Dostupné z: <https://www.forms.office.com/>
- [5] Microsoft Forms history. [online], 2021, [cit. 2021-03-20]. Dostupné z: https://en.wikipedia.org/wiki/Microsoft_Forms
- [6] Survio About. [online], 2021, [cit. 2021-03-20]. Dostupné z: <https://www.survio.com/cs/o-nas>
- [7] Fyrebox. [online], 2021, [cit. 2021-03-20]. Dostupné z: <https://www.fyrebox.com/>
- [8] Kahoot! popularity in US schools. [online], 2018, [cit. 2021-03-20]. Dostupné z: <https://techcrunch.com/2018/01/18/education-quiz-app-kahoot-says-its-now-used-in-50-of-all-us-classrooms-70m-users-overall/>
- [9] Microsoft Family Safety launches on Android and iOS. [online], 2020, [cit. 2021-03-20]. Dostupné z: https://www.business-standard.com/article/technology/microsoft-launches-family-safety-app-for-ios-and-android-for-digital-safety-120072900366_1.html
- [10] Figma. [online], 2021, [cit. 2021-03-28]. Dostupné z: <https://www.figma.com/>

- [11] What Is a Single Page Application and Why Do People Like Them so Much? [online], [cit. 2021-04-02]. Dostupné z: <https://www.bloomreach.com/en/blog/2018/07/what-is-a-single-page-application.html>
- [12] SPA vs MPA: The definitive guide for decision makers. [online], 2021, [cit. 2021-04-02]. Dostupné z: <https://www.mindk.com/blog/single-page-applications-the-definitive-guide/>
- [13] Universal Mode. [online], 2021, [cit. 2021-04-03]. Dostupné z: <https://www.vuemastery.com/courses/scaling-vue-with-nuxt-js/universal-mode>
- [14] Angular. [online], 2021, [cit. 2021-04-03]. Dostupné z: <https://angular.io/>
- [15] React. [online], 2021, [cit. 2021-04-03]. Dostupné z: <https://reactjs.org/>
- [16] Vue.js. [online], 2021, [cit. 2021-04-03]. Dostupné z: <https://vuejs.org/>
- [17] Malec, O.: Frontend skladového systému, kapitola Návrh funkcí a architektury reálné aplikace. Diplomová práce. Czech Technical University in Prague, Faculty of Information Technology, 2019, s. 28–37.
- [18] My experience with Angular 2, React and Vue. [online], 2021, [cit. 2021-04-03]. Dostupné z: <https://rajrock38.medium.com/my-experience-with-angular-2-react-and-vue-fb654e3ecf33>
- [19] Meta framework Nuxt.js. [online], 2021, [cit. 2021-04-03]. Dostupné z: <https://en.wikipedia.org/wiki/Nuxt.js>
- [20] High-fidelity prototyping: What, When, Why and How? [online], 2021, [cit. 2021-04-23]. Dostupné z: <https://blog.prototypr.io/high-fidelity-prototyping-what-when-why-and-how-f5bbde6a7fd4>
- [21] Node.js. [online], 2021, [cit. 2021-04-25]. Dostupné z: <https://nodejs.org/en/>
- [22] npm. [online], 2021, [cit. 2021-04-25]. Dostupné z: <https://www.npmjs.com/>
- [23] JetBrains Webstorm. [online], 2021, [cit. 2021-04-25]. Dostupné z: <https://www.jetbrains.com/webstorm/>
- [24] Vuex. [online], 2021, [cit. 2021-04-24]. Dostupné z: <https://vuex.vuejs.org/>

-
- [25] NuxtJS documentation. [online], 2021, [cit. 2021-04-24]. Dostupné z: <https://nuxtjs.org/docs/2.x/get-started/installation>
- [26] Nuxt Firebase. [online], 2021, [cit. 2021-04-25]. Dostupné z: <https://firebase.nuxtjs.org/>
- [27] Firebase. [online], 2021, [cit. 2021-04-25]. Dostupné z: <https://console.firebase.google.com/>
- [28] Nuxt Axios. [online], 2021, [cit. 2021-04-25]. Dostupné z: <https://axios.nuxtjs.org/>
- [29] Nuxt I18n. [online], 2021, [cit. 2021-04-25]. Dostupné z: <https://i18n.nuxtjs.org/>
- [30] Vuex. [online], 2021, [cit. 2021-04-25]. Dostupné z: <https://www.atlassian.com/continuous-delivery/continuous-integration>
- [31] What is Continuous Deployment? [online], 2021, [cit. 2021-04-25]. Dostupné z: <https://www.atlassian.com/continuous-delivery/continuous-deployment>
- [32] GitLab. [online], 2021, [cit. 2021-04-25]. Dostupné z: <https://about.gitlab.com/>
- [33] Docker. [online], 2021, [cit. 2021-04-25]. Dostupné z: <https://www.docker.com/>
- [34] Nuxt Sentry. [online], 2021, [cit. 2021-04-25]. Dostupné z: <https://sentry.nuxtjs.org/>
- [35] Sentry. [online], 2021, [cit. 2021-04-25]. Dostupné z: <https://sentry.io/welcome/>
- [36] Krug, S.: Don't make me think, kapitola Chapter 9. Usability testing on 10 cents a day. New Riders, třetí vydání, 2014, ISBN 0-321-96551-5, s. 106–107.
- [37] Pandemie onemocnění Covid-19 v České republice. [online], 2021, [cit. 2021-05-08]. Dostupné z: https://cs.wikipedia.org/wiki/Pandemie_covidu-19_v_%C4%8Cesku
- [38] Why You Only Need to Test with 5 Users. [online], 2000, [cit. 2021-05-12]. Dostupné z: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>
- [39] Jest. [online], [cit. 2021-05-12]. Dostupné z: <https://jestjs.io/>
- [40] Clockify. [online], 2021, [cit. 2021-05-09]. Dostupné z: <https://clockify.me/>

Seznam použitých zkratk

REST Representational state transfer

UI User Interface

UX User Experience

GUI Graphical User Interface

HTML HyperText Markup Language

CSS Cascading Style Sheets

JS JavaScript

AJAX Asynchronous JavaScript and XML

SEO Search engine optimization

XSS Cross-Site Scripting

SPA Single Page Application

SSR Server-Side Rendering

IDE Integrated Development Environment

API Application Programming Interface

URL Uniform Resource Locator

NPM Node Package Manager

BaaS Backend-as-a-Service

HTTP Hypertext Transfer Protocol

A. SEZNAM POUŽITÝCH ZKRATEK

CMS Content Management System

CI Continuous Integration

CD Continuous Deployment

Případy užití a scénáře

B.1 Sledování dětské aktivity

UC-1 Zobrazení poslední dětské aktivity

Popis: Uživatel uvidí informace o dětské aktivitě na mobilu.

Scénář:

1. Systém zobrazí obrazovku pro výběr dítěte.
2. Uživatel zvolí dítě.
3. Systém zobrazí kartu dítěte s jeho poslední aktivitou.

UC-2 Zobrazení stráveného času

Popis: Uživatel uvidí přehled stráveného času.

Scénář:

1. Systém zobrazí obrazovku pro výběr dítěte.
2. Uživatel zvolí dítě.
3. Systém zobrazí kartu dítěte s informacemi, jak dítě trávilo čas na mobilu.

UC-3 Zobrazení vyplněných kvízů

Popis: Uživatel uvidí přehled úspěšnosti dítěte v přiřazených kvízech.

Scénář:

1. Systém zobrazí obrazovku pro výběr dítěte.
2. Uživatel zvolí dítě.
3. Systém zobrazí kartu dítěte se statistikami úspěšnosti v kvízech.

UC-4 Zobrazení detailu vyplněného kvízu

Popis: Uživatel si zobrazí detail vyplněného kvízu.

Scénář:

1. Systém dodá obrazovku, kde bude seznam vyplněných kvízů.
2. Uživatel zvolí ze seznamu kvíz.
3. Systém zobrazí detail kvízu včetně správných odpovědí a odpovědí zadaných dítětem.

B.2 Správa uživatelových kvízů

UC-5 Zobrazení mých kvízů

Popis: Uživatel má přehled o kvízech, které může přiřadit.

Scénář:

1. Uživatel indikuje systému, že se chce dostat na seznam kvízů.
2. Systém dodá obrazovku, kde bude seznam uživatelových kvízů.

UC-6 Odebrání kvízu z uživatelových kvízů

Popis: Uživatel odstraní kvíz.

Scénář:

1. Systém poskytne obrazovku dostupných kvízů.
2. Uživatel najde v seznamu nebo si vyhledá kvíz, který chce odstranit.
3. Uživatel klikne na akci smazání kvízu.
4. Systém vyžaduje po uživateli potvrzení akce.
5. Uživatel potvrdí akci.
6. Systém odstraní kvíz z uživatelových kvízů. V případě, že kvíz byl vytvořen uživatelem, odstraní ho i z databáze.

UC-7 Vyhledávání kvízů v seznamu

Popis: Uživatel si na stránce dostupných kvízů vyhledá konkrétní kvíz podle vyhledávacích kritérií.

Scénář:

1. Systém dodá obrazovku s kvízy.
2. Uživatel zadá kritéria do vyhledávacího filtru.
3. Systém podle kritérií vyfiltruje příslušné kvízy.

B.3 Tvorba kvízů

UC-8 Vytvoření nového kvízu

Popis: Uživatel vytvoří vlastní kvíz.

Scénář:

1. Systém dodá obrazovku s kvízy.
2. Uživatel spustí akci tvorby nového kvízu.
3. Systém zobrazí obrazovku s formulářem pro tvorbu kvízu.
4. Uživatel vyplní povinné informace o kvízu.
5. Uživatel přidá alespoň jednu otázku.
6. Uživatel ukončí tvorbu kvízu.
7. Systém uloží kvíz.

UC-9 Vytvoření otázky

Popis: Uživatel při tvorbě kvízu vytvoří otázku.

Scénář:

1. Uživatel se nachází na stránce dostupných kvízů.
2. Uživatel spustí akci pro vytvoření nové otázky.
3. Systém zobrazí nový formulář pro výběr typu otázky.
4. Uživatel vybere typ otázky.
5. Systém zobrazí formulář pro daný typ otázky.
6. Uživatel vyplní příslušná pole (textem nebo přiloží obrázek).
7. Uživatel spustí akci tvorby nové otázky nebo akci ukončení tvorby kvízu.
8. Systém uloží otázku.

UC-10 Změna otázky

Popis: Uživatel si při tvorbě či editaci kvízu upraví otázku.

Scénář:

1. Uživatel se přesune na detail otázky.
2. Systém zobrazí detail otázky.
3. Uživatel změní otázku.
4. Systém uloží otázku.

UC-11 Smazání otázky

Popis: Uživatel při tvorbě či editaci kvízu smaže otázku.

Scénář:

B. PŘÍPADY UŽITÍ A SCÉNÁŘE

1. Uživatel vyvolá akci smazání otázky.
2. Systém smaže otázku.

UC-12 Náhled kvízu při tvorbě

Popis: Uživatel při tvorbě či editaci kvízu uvidí, jak bude kvíz rozložen na mobilním zařízení.

Scénář:

1. Uživatel se nachází na obrazovce tvorby kvízu.
2. Systém zobrazí vzhled aktuálního kvízu v mobilní aplikaci.

UC-13 Editace kvízu

Popis: Uživatel edituje jím dříve vytvořený kvíz.

Scénář:

1. Uživatel se nachází na stránce dostupných kvízů.
2. Uživatel spustí akci editace kvízu u záznamu vybraného kvízu.
3. Systém zobrazí obrazovku editace kvízu.
4. Uživatel upraví informace o kvízu a/nebo znění otázek. Může přidat či smazat otázky.
5. Uživatel ukončí tvorbu kvízu.
6. Systém uloží kvíz.

B.4 Sdílení kvízů

UC-14 Sdílení kvízu do obchodu

Popis: Uživatel sdílí kvíz s komunitou.

Scénář:

1. Uživatel se nachází na stránce dostupných kvízů.
2. Uživatel spustí akci sdílení u vybraného kvízu.
3. Systém přidá kvíz do obchodu.

UC-15 Sdílení kvízu při tvorbě

Popis: Uživatel na konci tvorby kvízu přidá kvíz do obchodu.

Scénář: Navazuje na případ užití *Vytvoření nového kvízu*.

1. Systém zobrazí obrazovku s možností sdílet kvíz s komunitou.
2. Uživatel potvrdí sdílení.
3. Systém kvíz přidá do obchodu.

UC-16 Odebrání kvízu z obchodu

Popis: Uživatel chce odstranit kvíz z obchodu.

Scénář:

1. Uživatel se nachází na obrazovce dostupných kvízů.
2. Uživatel zahájí akci zrušení sdílení u vybraného kvízu.
3. Systém odebere kvíz z obchodu.

B.5 Obchod

UC-17 Vyhledání kvízu v obchodě

Popis: Uživatel v obchodě vyhledá kvízy podle zadaných kritérií.

Scénář:

1. Uživatel se nachází na obrazovce obchodu.
2. Uživatel vyplní vyhledávací filtry podle svých potřeb.
3. Systém vyhledá kvízy odpovídající zadaným filtrům.

UC-18 Zobrazení detailních informací o kvízu

Popis: Uživatel si v obchodě zobrazí detail vybraného kvízu.

Scénář:

1. Uživatel se vyskytuje na stránce obchodu.
2. Uživatel poklepe na položku kvízu.
3. Systém zobrazí obrazovku s detailem kvízu.

UC-19 Zobrazení kvízových otázek

Popis: Uživatel si na detailu kvízu zobrazí příslušné otázky.

Scénář: Navazuje na případ užití *Zobrazení detailních informací o kvízu*.

1. Uživatel zahájí akci zobrazení otázek.
2. Systém zobrazí detaily jednotlivých otázek.

UC-20 Stažení kvízu

Popis: Uživatel si stáhne cizí kvíz do vlastní kolekce kvízů.

Scénář:

1. Uživatel se nachází na obrazovce detailu kvízu.
2. Uživatel spustí akci stažení kvízu.
3. Systém stáhne kvíz do uživatelova seznamu kvízů.

UC-21 Přidělení kvízu dítěti

Popis: Uživatel přiřadí cizí kvíz jednomu ze svých dětí.

Scénář:

1. Uživatel se nachází na obrazovce detailu kvízu.
2. Uživatel vyvolá akci přiřazení kvízu.
3. Systém nabídne děti.
4. Uživatel vybere, kterým dětem chce kvíz přiřadit.
5. Systém stáhne kvíz do uživateli kolekce a přiřadí ho vybraným dětem.

UC-22 Hodnocení kvízu

Popis: Uživatel ohodnotí kvíz, který právě prohlíží.

Scénář:

1. Uživatel se nachází na obrazovce detailu kvízu.
2. Uživatel ohodnotí kvíz.
3. Systém uloží hodnocení.

UC-23 Nahlášení kvízu

Popis: Uživatel shledá, že daný kvíz má nevhodný obsah a nahlásí ho.

Scénář:

1. Uživatel se nachází na stránce detailu kvízu.
2. Uživatel začne akci nahlášení.
3. Systém zobrazí formulář reportu.
4. Uživatel vyplní formulář a odešle ho.
5. Systém uloží hlášení.

B.6 Správa aplikací

UC-24 Přidání aplikace do zakázaných aplikací

Popis: Uživatel přidá aplikaci do zakázaných aplikací dítěte.

Scénář:

1. Uživatel se nachází na obrazovce správy dětí.
2. Uživatel vybere dítě.
3. Systém zobrazí kartu dítěte s aplikacemi.
4. Uživatel kliknutím na aplikaci označí, že ji chce zakázat.

5. Systém uloží volbu a graficky indikuje, že je aplikace zakázána.

UC-25 Odebrání aplikace ze zakázaných aplikací

Popis: Uživatel odebere aplikaci do zakázaných aplikací dítěte.

Scénář:

1. Uživatel se nachází na obrazovce správy dětí.
2. Uživatel vybere dítě.
3. Systém zobrazí kartu dítěte s aplikacemi.
4. Uživatel kliknutím na zakázanou aplikaci označí, že ji chce povolit.
5. Systém uloží volbu a graficky indikuje, že je aplikace povolena.

B.7 Správa uživatelského účtu

UC-26 Změna základních údajů

Popis: Uživatel změní své údaje v účtu.

Scénář:

1. Uživatel se nachází na obrazovce nastavení.
2. Uživatel změní své informace.
3. Uživatel ukončí úpravu informací.
4. Systém uloží informace.

UC-27 Změna hesla

Popis: Uživatel změní své přístupové heslo.

Scénář:

1. Uživatel se nachází na obrazovce nastavení.
2. Uživatel spustí akci změny hesla.
3. Systém zobrazí obrazovku s formulářem pro změnu hesla.
4. Uživatel vyplní formulář a ukončí změnu.
5. Systém uloží změnu hesla.

UC-28 Změna typu účtu

Popis: Uživatel změní svůj typ účtu (svoji roli).

Scénář:

1. Uživatel se nachází na obrazovce nastavení.
2. Uživatel vybere žádoucí typ účtu.

3. Systém uloží uživatelskou volbu.

UC-29 Smazání účtu

Popis: Uživatel smaže svůj účet.

Scénář:

1. Uživatel se nachází na obrazovce nastavení.
2. Uživatel inicializuje akci zrušení účtu.
3. Systém vyžaduje po uživateli potvrzení akce.
4. Uživatel potvrdí akci.
5. Systém odešle kód na email uživatele.
6. Systém zobrazí formulář na vyplnění potvrzovacího kódu.
7. Uživatel zadá kód a odešle formulář.
8. Systém uživatele odhlásí, přeměruje na domovskou stránku a smaže jeho účet.

B.8 Správa dětských účtů

UC-30 Změna dětských údajů

Popis: Uživatel změní základní informace o dětském účtu.

Scénář:

1. Systém zobrazí obrazovku pro výběr dítěte.
2. Uživatel vybere dítě.
3. Systém zobrazí kartu dítěte.
4. Uživatel spustí akci editace dítěte.
5. Systém zobrazí editační formulář.
6. Uživatel vyplní formulář a odešle ho.
7. Systém uloží nové informace o dětském účtu.

UC-31 Smazání dětského účtu

Popis: Uživatel smaže dětský účet.

Scénář:

1. Uživatel se nachází na kartě dítěte.
2. Uživatel spustí akci editace dítěte.
3. Systém zobrazí editační panel.
4. Uživatel vyvolá akci smazání účtu.

5. Systém vyžaduje po uživateli potvrzení akce.
6. Uživatel potvrdí akci.
7. Systém odešle kód na email uživatele.
8. Systém zobrazí formulář na vyplnění potvrzovacího kódu.
9. Uživatel zadá kód a odešle formulář.
10. Systém smaže dětský účet a přesměruje uživatele na obrazovku se záložkami pro výběr dítěte.

B.9 Správa dětského času

UC-32 Přidání nového rozvrhu

Popis: Uživatel přidá dítěti nový rozvrh.

Scénář:

1. Uživatel se nachází na kartě dítěte.
2. Uživatel spustí akci přidání rozvrhu.
3. Systém zobrazí panel s formulářem pro tvorbu rozvrhu.
4. Uživatel vyplní formulář a uloží ho.
5. Systém uloží rozvrh a nastaví ho vybranému dítěti.

UC-33 Aktivace rozvrhu

Popis: Uživatel zapne dítěti vybraný rozvrh.

Scénář:

1. Uživatel se nachází na kartě dítěte.
2. Uživatel spustí akci aktivaci rozvrhu.
3. Systém aktivuje rozvrh.

UC-34 Deaktivace rozvrhu

Popis: Uživatel vypne dítěti vybraný rozvrh.

Scénář:

1. Uživatel se nachází na kartě dítěte.
2. Uživatel vyvolá akci deaktivace rozvrhu.
3. Systém deaktivuje rozvrh.

UC-35 Editace rozvrhu

Popis: Uživatel upraví existující rozvrh.

Scénář:

B. PŘÍPADY UŽITÍ A SCÉNÁŘE

1. Uživatel se nachází na kartě dítěte.
2. Uživatel spustí akci editaci rozvrhu.
3. Systém zobrazí panel s editačním formulářem.
4. Uživatel vyplní formulář a klikne na tlačítko uložení.
5. Systém uloží úpravy rozvrhu.

UC-36 Smazání rozvrhu

Popis: Uživatel smaže rozvrh.

Scénář:

1. Uživatel se nachází na kartě dítěte.
2. Uživatel zahájí akci smazání rozvrhu.
3. Systém smaže rozvrh.

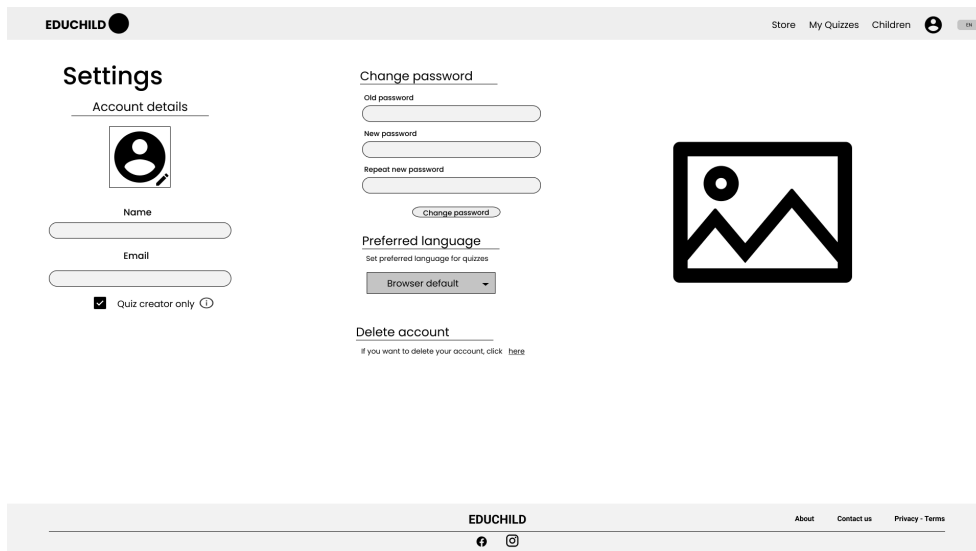
UC-37 Správa volného času dítěte

Popis: Uživatel nastaví dítěti délku volného času.

Scénář:

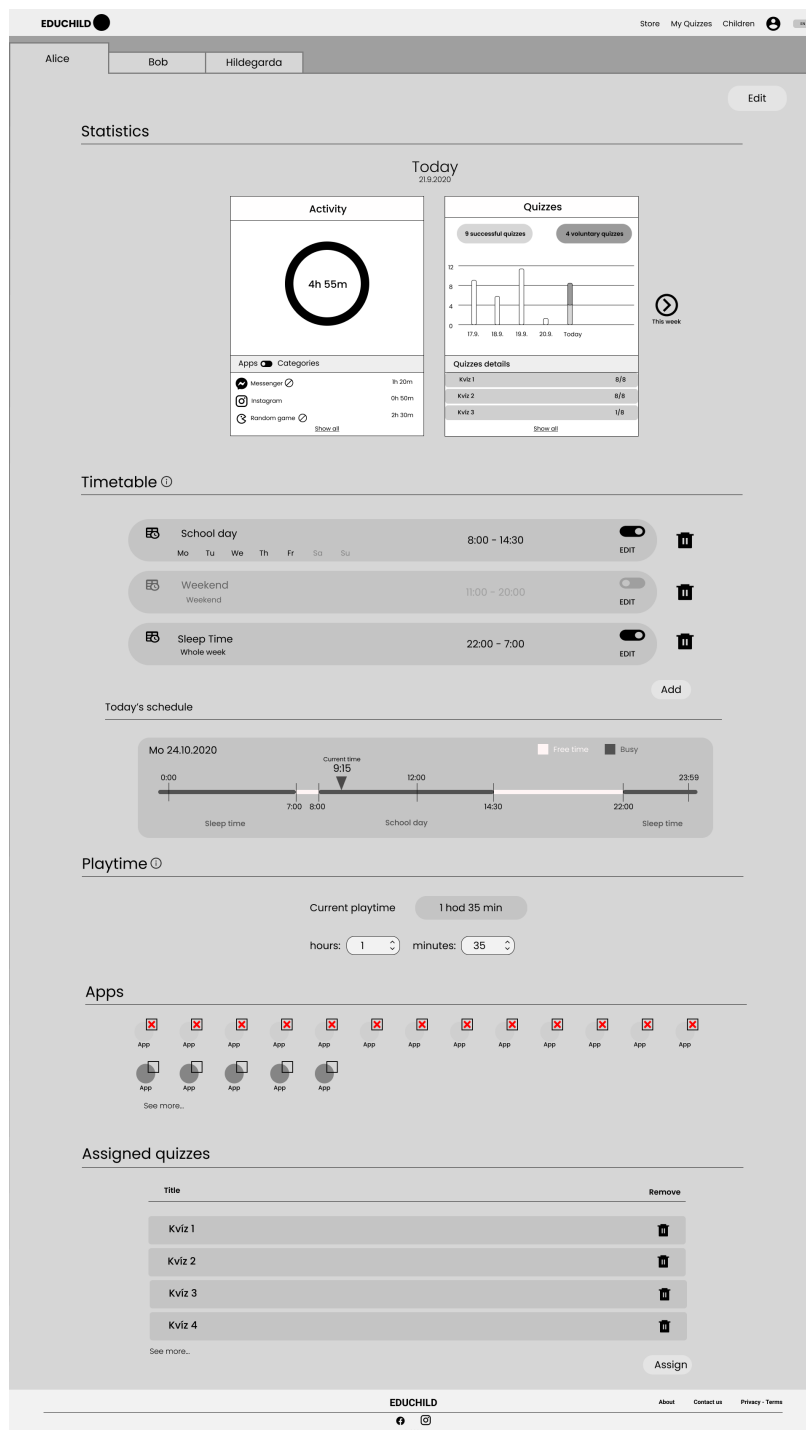
1. Uživatel se nachází na kartě dítěte.
2. Uživatel změní hodnoty na formuláři volného času.
3. Systém uloží změny.

Wireframes



Obrázek C.1: Nastavení uživatele

C. WIREFRAMES



Obrázek C.2: Obrazovka správy dětí

Creating Timetable

Name

Time

Set the time, when all choosed apps will be blocked

from: to:

Days

Choose days, when this timetable will be active

Mo Tu We Th Fr

Sa Su

Confirm

Obrázek C.3: Panel tvorby rozvrhu

C. WIREFRAMES

Quiz title Correctly answered: 3/5

Quiz info

Name: Obecné znalosti 1
Category: All, Xyz
Age: 5 - 10
Description: Bla bla bla bla bla, xyz
uifiausfiarfs hhhc.

Question 1




Type: Open
Question: Kolik dní má rok?
Answer: 365 ✓

Question 2

Type: Multichoice
Question: Které zvíře je savec?
Answer:
Kočka ✓
Pes ✓
Žralok
Moucha ✓

Question 3

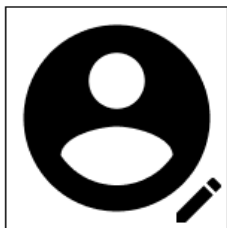
Type: Pairs
Question: Přiřaď název k vozidlu.

| | | | |
|---|---|---------|---|
|  | → | Traktor | ✓ |
|  | → | Auto | ✗ |
|  | → | Kolo | ✗ |

Question 1 ✓
Question 2 ✗
Question 3 ✗
Question 4 ✓
Question 5 ✓

Obrázek C.4: Panel kontroly vyplněného kvízu

Bob's account detail



Name

Date of birth

Delete account

Obrázek C.5: Panel editace dětského účtu



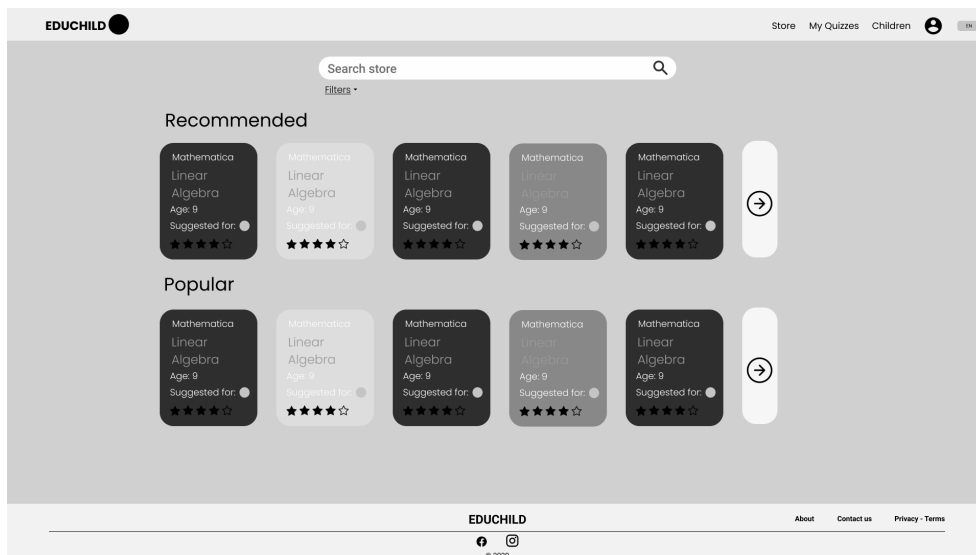
Assign quiz to:

 Alice

 Bob

Assign

Obrázek C.6: Panel přiřazení dítěte ke kvízu



Obrázek C.7: Obrazovka obchodu

Quiz 1 ★★★★★☆



Category

Biology

Mathematica

IT

Physics

Recommended age

9

of questions

22

Suggested for



[Assign](#)

Language

Čeština

Created by

Users

Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Augue quis nunc, enim nunc leo a eget in. Viverra.

[See questions...](#)

Report quiz

Obrázek C.8: Detail kvízu v obchodě

Check the quiz content before downloading! ⓘ

Quiz 1

Category

Biology Mathematics IT Physics

Recommended age # of questions

9 22

Language Created by

Čeština Users

Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Augue quis nunc, enim nunc leo a eget in. Viverra.

Question 1

Type: Open

Question: Kolik dní má rok?

Answer: 365

Question 2

Type: Multichoice

Question: Které zvíře je savec?

Answer:

Kočka

Pes


Žralok


Moucha


Question 3

Type: Pairs

Question: Přिřaď název k vozidlu.

 → Traktor

 → Kolo

 → Auto

Skip review Report quiz Ok

Obrázek C.9: Detail kvízu před přiřazením

Report quiz

Help us understand what's happening with this quiz.

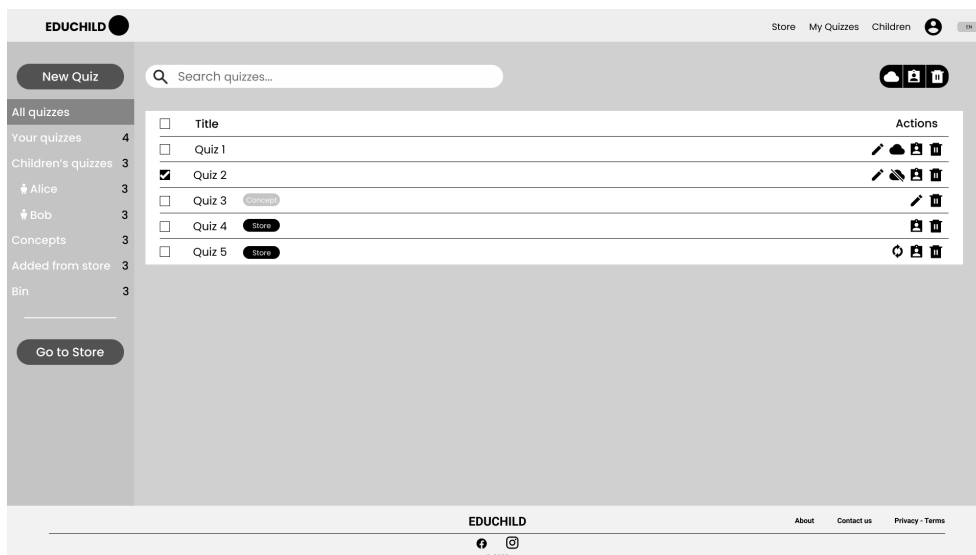
How would you describe it?

Quiz contains: (Select one or more)

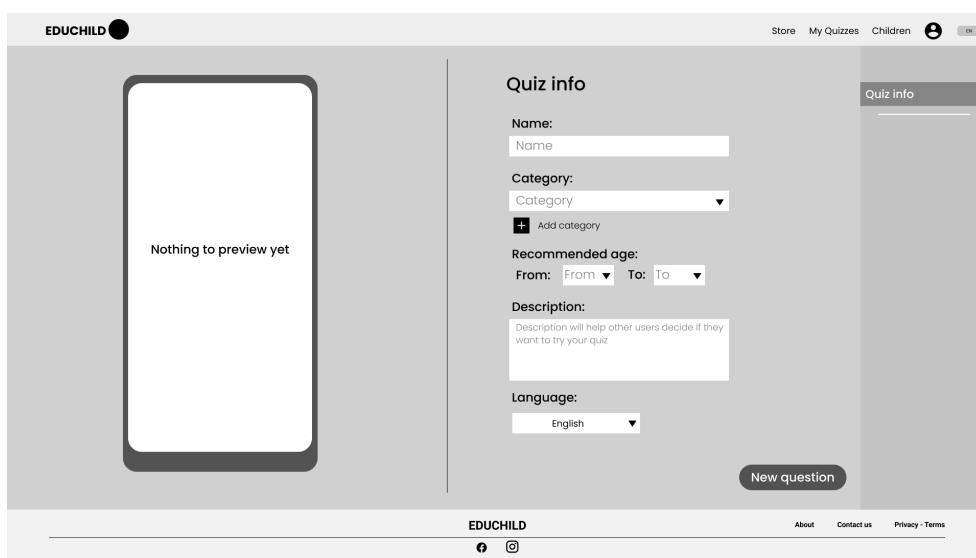
- sexual content
- offensive content
- political content
- violence
- false information
- other

Describe the problem in your own words

Obrázek C.10: Nahlášení kvízu

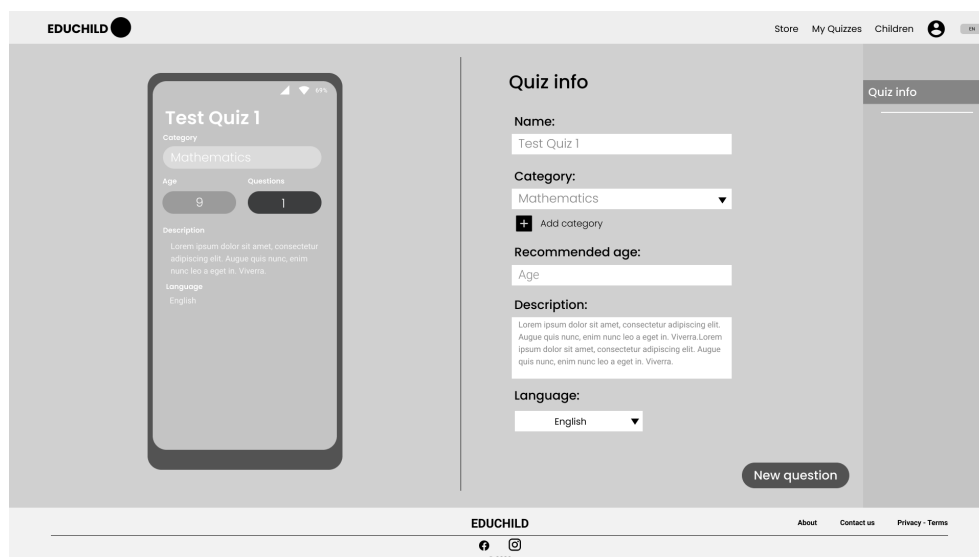


Obrázek C.11: Obrazovka seznamu uživatelských kvízů

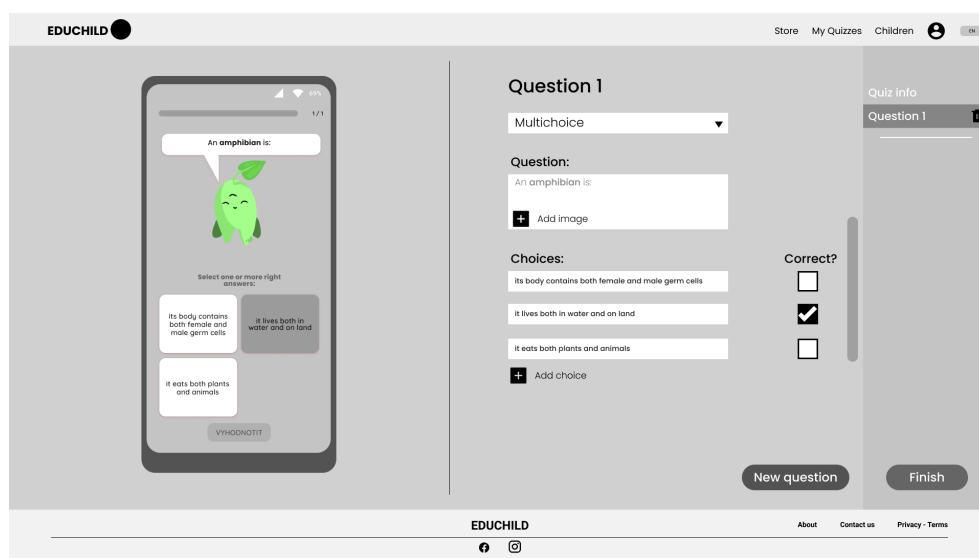


Obrázek C.12: Tvorba kvízu - informace o kvízu

C. WIREFRAMES



Obrázek C.13: Tvorba kvízu - vyplněné informace o kvízu



Obrázek C.14: Tvorba kvízu - otázka

Testovací scénáře

Testovací scénář 1

Registrace

Uvedení do děje:

Jste rodičem dvou dětí ve školním věku. Aničce je 12 let, Bořkovi bude brzy 9 let. Děti tráví většinu dne na mobilu a kdykoliv se je pokusíte přesvědčit, aby si zkusili jít hrát jiným způsobem, skončí to nezdarem.

Od známých se dozvíte o jisté aplikaci, která by Vám v tomto směru mohla pomoci. Aplikace blokuje volný čas, který děti na mobilech tráví. Přiřazuje jim k vyplnění znalostní kvízy a podle jejich úspěšnosti jim poté zpřístupňuje ostatní aplikace, na kterých tráví svůj čas běžně. Rozhodnete se, že tuto aplikaci vyzkoušíte. Stáhnete ji svým dětem do mobilů a během používání zjistíte, že aplikace také disponuje webem, prostřednictvím kterého můžete snadněji spravovat mobilní aktivitu dětí. Rozhodnete se jej také vyzkoušet.

Úkoly:

- Registrujte se ve webové aplikaci

Testovací scénář 2

Získání kvízů

Uvedení do děje:

Pokračujeme ze scénáře 1. Přihlásili jste se ke svému účtu a zjišťujete, že stejně jako v mobilní aplikaci lze také na webu dětem přiřadit kvízy k vyplnění. Rozhodnete se dětem nějaký kvíz vybrat.

Úkoly:

- Přiřaďte dětem kvíz (věnujte pozornost věku vašich dětí)

Testovací scénář 3

Tvorba kvízů

Uvedení do děje:

Od děje scénáře 2 uběhl den. Vy už nyní máte více času než včera a proto Vás napadne, že se podíváte, zda lze vytvářet kvízy vlastní.

Zjišťujete, že ano, a rozhodnete se tedy vytvořit kvíz pro Aničku, neboť se jí ten včerejší moc nepovedl.

Úkoly:

- Vytvořte vlastní kvíz
- Vytvořený kvíz přiřaďte Aničce

Testovací scénář 4

Správa času

Uvedení do děje:

Od děje scénáře 3 již uběhlo několik dní. Z mobilní aplikace víte, že lze každému dítěti přesně nastavit, na jak dlouho a v které dny bude jejich čas na mobilním zařízení omezen. Rozhodnete se funkcionalitu vyzkoušet na webu a podle vlastního uvážení nastavíte dobu neomezeného používání mobilu denně pro obě děti.

Úkoly:

- Omezte denní čas dětí na jejich zařízeních

Pokračování děje:

Zároveň si uvědomujete, že by bylo vhodné omezit jejich aktivitu v době, kdy se od nich očekává, že nebudou používat mobil - například když jsou ve škole nebo když mají spát.

Úkoly:

- Omezte dětskou aktivitu na zařízeních

Testovací scénář 5

Odebrání kvízů

Uvedení do děje:

Anička již všechny vámi přiřazené kvízy splnila. Aby se jí nehromadily již vypracované (a tedy pro ni zbytečné) kvízy, rozhodnete se jí je odebrat.

Úkoly:

- Odeberte Aničce kvízy

Testovací scénář 6

Uvedení do děje:

Při letmém nahlédnutí Bořkovi přes rameno v době plnění kvízu jste si všiml/a, že Vámi vytvořený kvíz obsahuje faktickou chybu. Nechcete vaše děti špatně učit a proto se rozhodnete chybu opravit.

Úkoly:

- Opravte chybu v kvízu (poznámka: Vyberte libovolný kvíz a upravte ho)

Dotazník po testování

Diskuze po uživatelském testování

1. Jak složitá pro Vás byla práce se systémem?

Velmi jednoduchá 1 - 2 - 3 - 4 - 5 Velmi složitá

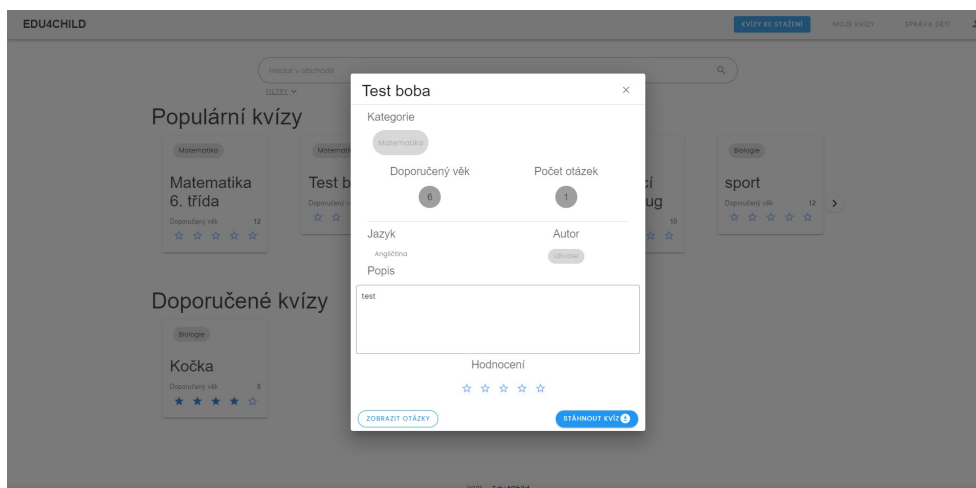
2. Co se Vám v systému líbilo / připadalo Vám dobře vyřešené?

3. Co se Vám naopak nelíbilo?

4. Nastal během testu moment nebo situace, kdy jste nevěděli, jak dál?

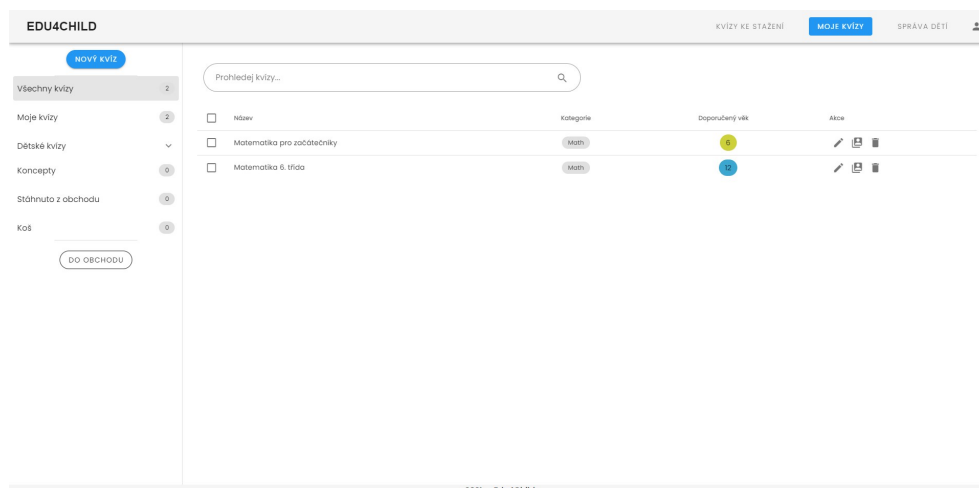
5. Máte k testování/aplikaci jakékoliv připomínky či dotazy?

Ukázky systému

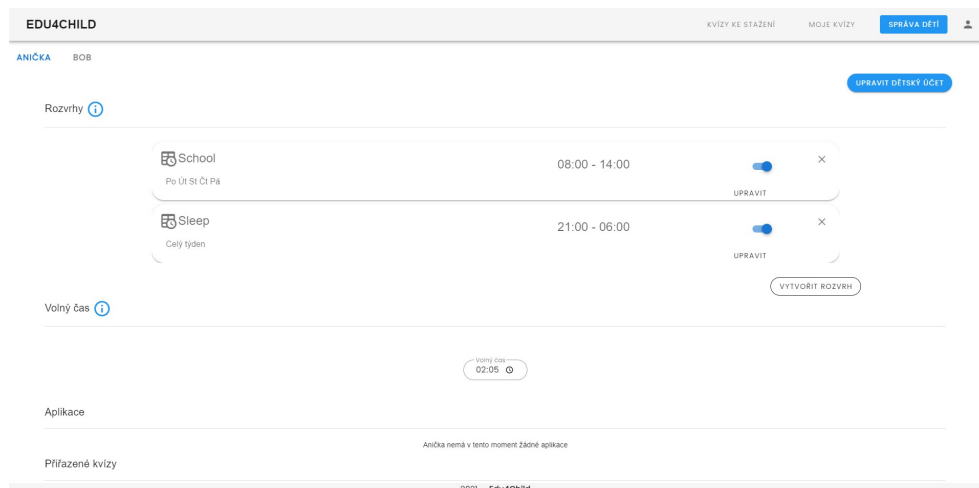


Obrázek F.1: Detail kvízu s upraveným tlačítkem stažení

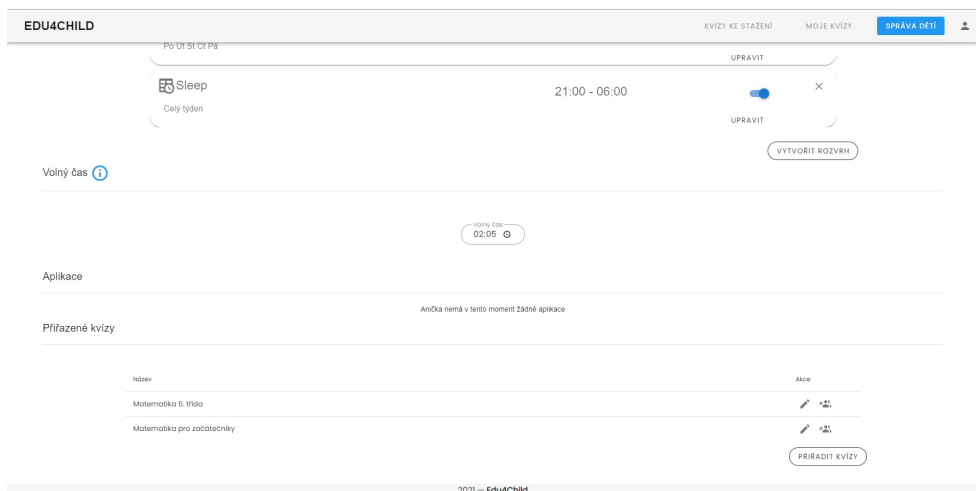
F. UKÁZKY SYTÉMU



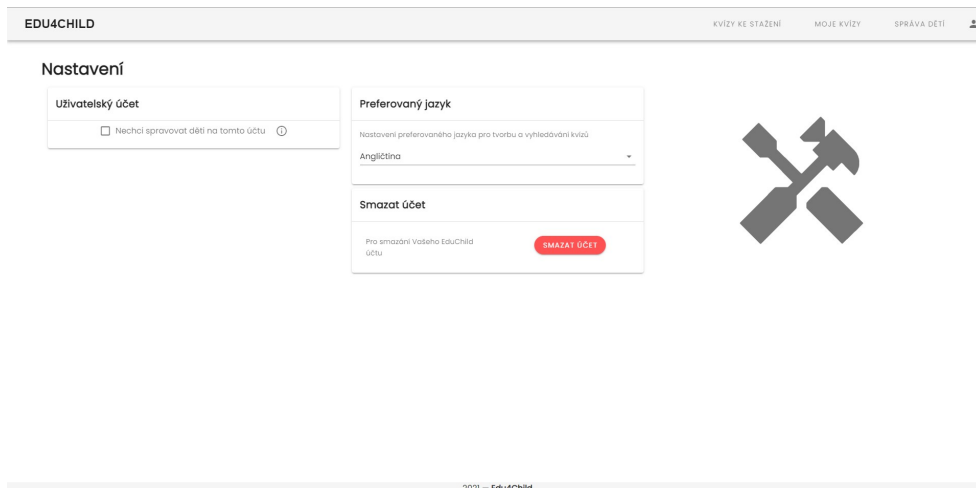
Obrázek F.2: Stránka Moje Kvízy



Obrázek F.3: Rozvrhy a nastavení volného času



Obrázek F.4: Přiřazení kvízů ve Správě dětí



Obrázek F.5: Stránka Nastavení

F. UKÁZKY SYTÉMU

The screenshot shows the 'Informace o kvízu' (Quiz Information) form in the EDU4CHILD system. The form includes the following fields:

- Jméno kvízu:** Text input field.
- Kategorie:** Dropdown menu.
- Věk:** Text input field with the value '6' entered.
- Popis:** Text area with a character count of 0 / 200.
- Jazyk kvízu:** Dropdown menu.
- Otázka #1:** A section for defining the question type, with options: 'Otevřená - Číselná' (Answered as number), 'Otevřená - Textová' (Answered as plain text), and 'Více možností' (Multiple choice). The 'Více možností' option is selected.

Navigation buttons at the top include 'KVÍZY KE STAŽENÍ', 'MOJE KVÍZY', and 'SPRÁVA DĚTI'. A 'DOKONČIT' (Finish) button is located at the bottom right of the form.

Obrázek F.6: Tvorba kvízu – vyplnění informací a výběr typu otázky

The screenshot shows the 'Otázka #1' (Question #1) form in the EDU4CHILD system. The form includes the following fields:

- Typ otázky:** Dropdown menu with 'Více možností' (Multiple choice) selected.
- Otázka:** Text area containing 'Testovací otázka' with a character count of 16 / 200.
- Možnost:** Two rows for defining answer options. The first row has 'Možnost' (0 / 200) and a checked 'Správně' (Correct) checkbox. The second row has 'Možnost' (0 / 200) and an unchecked 'Správně' checkbox.
- + PŘIDAT MOŽNOST:** Button to add more options.
- Vysvětlení:** Text area for providing an explanation with a character count of 0 / 200.

A blue '+ NOVÁ OTÁZKA' (New Question) button is located at the bottom center. A 'DOKONČIT' (Finish) button is located at the bottom right of the form.

Obrázek F.7: Tvorba kvízu – vyplnění otázky

Obsah přiloženého CD

| | | |
|--|----------------------------------|---|
| | readme.txt | stručný popis obsahu CD |
| | src | |
| | _ impl.txt | odkaz na zdrojové kódy implementace |
| | _ thesis..... | zdrojová forma práce ve formátu \LaTeX |
| | text | text práce |
| | _ BP_STEJSKAL_JAN_2021.pdf | text práce ve formátu PDF |
| | appendix | přílohy |
| | _ testovaci_scenare.pdf | scénáře uživatelského testování |
| | _ dotaznik.pdf | dotazník po uživatelském testování |
| | _ wireframes.txt..... | odkaz na modely návrhu uživatelského rozhraní |