

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Návrh uživatelského rozhraní kalendáře sdíleného mezi studenty a pedagogy pro organizaci doučování

Sabina Balaeva

Vedoucí: RNDr. Ladislav Serédi
Studijní program: Softwarové inženýrství a technologie
Květen 2023

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Balaeva** Jméno: **Sabina** Osobní číslo: **495556**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Návrh uživatelského rozhraní kalendáře sdíleného mezi studenty a pedagogy pro organizaci doučování

Název bakalářské práce anglicky:

User interface design of a planning calendar for tutoring lessons shared between students and teachers

Pokyny pro vypracování:

Provedte rešerši existujících kalendářů pro plánování doučování. Diskutujte výhody a nevýhody uživatelských rozhraní a způsobů využívání (UI/UX) jednotlivých řešení a navrhnete vlastní uživatelské rozhraní odstraňující nedostatky a omezení existujících kalendářů. Prozkoumejte technické možnosti vytvoření fungujícího modelu takového kalendáře a navrhnete pro ně vhodnou architekturu včetně způsobu tvorby UI, API a komunikačních protokolů mezi jednotlivé komponenty. Porovnejte možnosti vámi navrženého a stávajících řešení a naplánujte způsob a průběh uživatelského testování vašeho kalendáře.

V rámci nalezeného technického řešení implementujte prototyp Vaší aplikace s uceleným uživatelským rozhraním v takové formě, aby bylo možné provést uživatelské testování.

Naplánujte a proveďte formální uživatelské testování jak z hlediska studentů tak i učitelů. Výsledky testů vyhodnoťte a diskutujte je.

Seznam doporučené literatury:

Scheduling Meetings in Distance Learning, Jian Wang, Changyong Niu & Ruimin Shen

https://link.springer.com/chapter/10.1007/978-3-540-76837-1_63

Meeting Scheduling: Face-to-Face, Automatic Scheduler, and Email Based Coordination, Bongsik Shin & Kunihiko Higa,

https://www.tandfonline.com/doi/abs/10.1207/s15327744joc1502_3

Leveraging Mental Models in UX Design, Scott Benson, <https://www.toptal.com/designers/ux/mental-models-ux-design>

Identifying Usability Issues in Personal Calendar Tools, Dezhi Wu, International Conference on Networked Digital

Technologies, 2011, <https://link.springer.com/content/pdf/10.1007/978-3-642-22185-9.pdf>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

RNDr. Ladislav Serédi kabinet výuky informatiky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **17.02.2023**

Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **22.09.2024**

RNDr. Ladislav Serédi
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Studentka bere na vědomí, že je povinna vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studentky

Poděkování

Chtěla bych poděkovat RNDr. Ladislavu Serédi za vedení této práce, cenné rady a konzultace. Také bych chtěla poděkovat své rodině a přátelům za podporu během studia. Dále děkuji všem, kteří se zúčastnili uživatelského testování.

Prohlášení

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

V Praze, dne 25.5.2023

Abstrakt

Cílem této bakalářské práce je navržení a implementace uživatelského rozhraní kalendáře pro organizaci doučování mezi studenty a pedagogy. Nejprve je provedena rešerše existujících řešení. Na základě této rešerše jsou sestaveny funkční požadavky systému a vytvořen low-fidelity wireframe webové aplikace. Tento drátěný model čili wireframe je následně podroben uživatelskému testování, jehož výsledky jsou diskutovány. Serverová část aplikace je implementována pomocí Pocketbase. Pro klientskou část je použit React. Jako CSS framework je zvolen Tailwind v kombinaci s Flowbite. Kalendář je implementován pomocí kalendářové komponenty DayPilot. Výsledná aplikace je otestována uživateli a výsledky vyhodnoceny.

Klíčová slova: webová aplikace, uživatelské rozhraní, uživatelské testování, kalendář, Pocketbase, React, DayPilot, Tailwind, Flowbite

Vedoucí: RNDr. Ladislav Serédi

Abstract

The aim of this bachelor thesis is to design and implement a calendar user interface for organizing tutoring sessions between students and teachers. First, a search of existing solutions is conducted. Based on this research, the functional requirements of the system are compiled and a low-fidelity wireframe of the web application is created. This wireframe is then subjected to user testing, the results of which are discussed. The server side of the application is implemented using Pocketbase. For the client part, React is used. Tailwind is chosen as the CSS framework in combination with Flowbite. The calendar is implemented using DayPilot calendar component. The resulting application is tested by users and the results are evaluated.

Keywords: web application, user interface, user testing, calendar, Pocketbase, React, DayPilot, Tailwind, Flowbite

Title translation: User interface design of a planning calendar for tutoring lessons shared between students and teachers

Obsah

1 Úvod	3	6 Technologie	25
2 Rešerše	5	6.1 Backend	25
2.1 Seznámení s problémem	5	6.1.1 Json Server Api	25
2.2 Navržené řešení	5	6.1.2 Pocketbase	27
2.3 Analýza hotových řešení	5	6.1.3 BaaS	28
2.3.1 Google Calendar	6	6.2 Frontend	28
2.3.2 Appointlet	6	6.2.1 React	28
2.3.3 TutorBird	6	6.2.2 Vue	28
2.3.4 Calendly	7	6.2.3 Srovnání	29
2.3.5 FelSight	7	6.3 CSS frameworky	30
2.3.6 Doučuji.eu	7	6.3.1 Vlastnosti CSS frameworků	30
3 Analýza	9	6.3.2 Bootstrap	30
3.1 Vytvoření osoby	9	6.3.3 Tailwind	30
3.2 Funkční požadavky	10	6.3.4 Flowbite	30
3.3 Nefunkční požadavky	11	6.4 Vybrané technologie	31
3.4 Diagram případů užití	12	7 Implementace	33
4 Návrh uživatelského rozhraní	13	7.1 Backend	33
4.1 Mentální model	13	7.1.1 Konfigurace	33
4.1.1 Sjednocení mentálních modelů	13	7.1.2 AuthStore	33
4.2 Tři zákony použitelnosti	14	7.1.3 CRUD operace	35
4.3 Drátěný model	14	7.2 Frontend	36
4.3.1 Balsamiq	14	7.2.1 React Router	37
4.4 Mapa interakce	15	7.2.2 Vykreslování elementů	37
5 Testování návrhu uživatelského rozhraní	17	7.2.3 JSX	38
5.1 Způsob testování	18	7.2.4 Komponenta	38
5.2 Uživatelské testování použitelnosti učitelů	18	7.2.5 Životní cyklus a stav	40
5.2.1 Testovací scénář	18	7.2.6 Podmíněné vykreslování	41
5.2.2 Hodnocení náročnosti splnění úkolů	18	7.3 Implementace kalendáře	41
5.2.3 Čas strávený na úkolu	19	7.3.1 DayPilot	41
5.2.4 Celková spokojenost	19	8 Testování aplikace	45
5.2.5 Požadavky na změnu	20	8.1 Testování aplikace učitelů	45
5.2.6 Celkové zhodnocení	20	8.1.1 Testovací scénář	45
5.3 Uživatelské testování použitelnosti studenty	21	8.2 Testování aplikace studenty	48
5.3.1 Testovací scénář	21	8.2.1 Testovací scénář	48
5.3.2 Hodnocení náročnosti splnění úkolů	21	9 Závěr	53
5.3.3 Čas strávený na úkolu	21	Literatura	55
5.3.4 Celková spokojenost	22	A Seznam zkratk	59
5.3.5 Požadavky na změnu	22	B Ukázka low fidelity prototypu	61
5.3.6 Celkové zhodnocení	23	C Ukázka výsledné aplikace	69
		D Instalace a spuštění aplikace	79

Obrázky

3.1 Diagram případů užití	12	C.17 Hodnocení - pohled učitele . . .	77
4.1 Mapa interakce	15	C.18 Hodnocení - pohled studenta . .	77
5.1 Procento nalezených chyb testovanými uživateli	17	C.19 Ohodnocení učitele	78
6.1 Pocketbase dashboard	28	C.20 Profil učitele	78
7.1 Integrace DayPilot komponenty	42	D.1 Odkaz na GitLab repozitář se zdrojovým kódem aplikace	79
B.1 Pohled pro nepřihlášeného uživatele	61		
B.2 Registrovací stránka	62		
B.3 Přihlašovací stránka	62		
B.4 Nabídky doučování	63		
B.5 Vytvoření předmětu	63		
B.6 Nastavení dostupnosti	64		
B.7 Rezervované termíny	64		
B.8 Rezervované termíny v kalendáři	65		
B.9 Vybrání slotu pro rezervaci . . .	65		
B.10 Hodnocení učitele	66		
B.11 Platby	66		
B.12 Učitelé	67		
B.13 Studenti	67		
B.14 Profil učitele	68		
B.15 Profil studenta	68		
C.1 Pohled pro nepřišlášeného uživatele	69		
C.2 Přihlašovací stránka	69		
C.3 Nabídky doučování	70		
C.4 Předměty	70		
C.5 Vytvoření předmětu	71		
C.6 Nastavení dostupnosti	71		
C.7 Nadcházející události	72		
C.8 Rezervované události v kalendáři - pohled učitele	72		
C.9 Rezervované události v kalendáři učitele - pohled studenta	73		
C.10 Vybrání slotu pro rezervaci . . .	73		
C.11 Detail události	74		
C.12 Rezervované události v kalendáři - pohled studenta	74		
C.13 Seznam učitelů	75		
C.14 Seznam studentů	75		
C.15 Platby - pohled učitele	76		
C.16 Platby - pohled studenta	76		

Tabulky

5.1 Náročnost jednotlivých úkolů hodnocených učiteli	19
5.2 Průměrný čas učitelů strávený na úkolů ve vteřinách	19
5.3 Průměrná spokojenost učitelů s návrhem	19
5.4 Náročnost jednotlivých úkolů hodnocených studenty	22
5.5 Průměrný čas studentů strávený na úkolů ve vteřinách	22
5.6 Průměrná spokojenost studentů s návrhem	22
7.1 Vytvoření instance Pocketbase ..	33
7.2 Aktualizace stavu na základě změny tokenu	34
7.3 Odhlášení	34
7.4 Použití <code>pb.authStore.model</code> ..	35
7.5 Vytvoření časové dostupnosti ...	36
7.6 Aktualizace časové dostupnosti .	36
7.7 Získání časové dostupnosti	36
7.8 Smazání časové dostupnosti	37
7.9 React Routes	37
7.10 Tlačítko s Link	38
7.11 JSX	38
7.12 Modální okno	39
7.13 Zobrazení události	40
7.14 <code>useEffect</code> s argumentem prázdného pole	40
7.15 <code>useEffect</code> s argumentem	41
7.16 Logický operátor	41
7.17 <code>DayPilot</code> komponenta	43
8.1 Náročnost jednotlivých úkolů hodnocených učiteli	46
8.2 Průměrný čas učitelů strávený na úkolů ve vteřinách	47
8.3 Průměrná spokojenost učitelů s aplikací	47
8.4 Náročnost jednotlivých úkolů hodnocených studenty	49
8.5 Průměrný čas studentů strávený na úkolů ve vteřinách	50
8.6 Průměrná spokojenost studentů s aplikací	50

Kapitola 1

Úvod

Doučování je forma vzdělávání, které poskytuje studentovi individuální nebo skupinovou výuku od kvalifikovaných učitelů nebo studentů. V Pedagogickém slovníku je pojem doučování definován jako „forma pomoci žákům, kteří mají z různých důvodů problémy, nestačí školnímu tempu vyučování nebo dosahují slabších studijních výsledků“ [1]. Doučování tedy představuje způsob, který dokáže pomoci studentovi například pochopit látku ve škole, připravit se na zkoušky nebo se naučit nový jazyk. Doučování mohou být prováděna individuálně, ve skupinách nebo online.

V dnešní době nám pomáhá s nalezením doučování především internet. Samotný proces vybrání vhodného doučování nebo domluvení se na doučovací hodině může být zdlouhavý.

Z tohoto důvodu je záměrem této bakalářské práce vytvoření návrhu webové aplikace, která pomůže urychlit proces vybrání vhodného doučování a domluvení se na doučovací hodině mezi lektory a studenty. Aplikace umožní lektorům nastavit své časové možnosti, dle kterých se vytvoří časové sloty pro doučování v kalendáři. Studentům pak umožní vybrání vhodné doučovací hodiny dle parametrů jako čas, lokace, cena nebo hodnocení učitele.

V kapitole „Rešerše“ se práce zabývá přiblížením řešeného problému a návrhu jeho řešení. Dále tato část obsahuje rešerši existujících řešení na současném zahraničním i českém trhu a jejich srovnání.

Kapitola „Analýza“ se věnuje analýze problému. Jsou formulovány persony, funkční a nefunkční požadavky a případy užití aplikace.

Následující kapitola popisuje návrh uživatelského rozhraní. Tato část obsahuje popis tří zákonů použitelnosti, nástroj použitý k vytvoření návrhu uživatelského rozhraní a samotný drátěný model uživatelského rozhraní.

Další kapitola se věnuje testování návrhu uživatelského rozhraní. Je zde popsán způsob výběru počtu testovaných osob, scénáře použité k testování a zhodnocené výsledky testování.

Kapitola „Technologie“ rozebírá technologie serverové a klientské části aplikace, z nichž jsou vybrány frameworky, které budou využity při implementaci.

Kapitola „Implementace“ popisuje způsob implementace a využití funkce jednotlivých frameworků ve výsledné aplikaci.

Poslední část se věnuje uživatelskému testování a interpretaci výsledků spokojenosti uživatelů s finální podobou aplikace.

Kapitola 2

Rešerše

2.1 Seznámení s problémem

Po vlastní zkušenosti se zapsáním se na doučovací hodinu, jsem narazila na nedostatky, které bych chtěla v této práci řešit. Samotné nalezení časově vhodné doučovací hodiny nebylo snadné. Většina webových stránek na českém trhu umožňuje lektorům uvádět pouze nepřesné časové údaje jako například dopoledne, poledne, odpoledne atd. Pokud se tedy chcete zapsat na doučovací hodinu, aplikace vám umožní poslat e-mail se zájmem o doučování. Dále probíhá komunikace mezi učitelem a studentem pomocí emailů, která je dle mého názoru zdlouhavá. Z tohoto důvodu jsem se rozhodla řešit tento problém. Chtěla bych tedy zjednodušit a zrychlit komunikaci mezi lektorem a studentem.

2.2 Navržené řešení

Tento problém může pomoci řešit webová aplikace s kalendářem zaměřeným pro doučování, který umožní naplánovat doučovací hodiny. Využití kalendáře, který se umožňuje zapsat na určitou hodinu u daného lektora, kde uvidíme detaily jako místo a čas setkání, cenu doučovací hodiny, přehled o platbách za hodiny a další parametry, nám pak může ušetřit čas.

2.3 Analýza hotových řešení

Na trhu webových aplikací, které umožňují vytváření časových slotů a zapisování se na ně, najdeme mimo Českou republiku širokou škálu webových aplikací: Google Calendar, Appointlet, TutorBird, Calendly. Na českém trhu pak najdeme Doučuji a FelSight, které však neumožňují vytváření časových slotů. Dané produkty obecně zhodnotím a vyzdvihnu jejich silné a slabé stránky.

■ 2.3.1 Google Calendar

Jedna z velice známých aplikací[3], která má hojný počet uživatelů, je Google Calendar . Disponuje mnoha funkcemi například jako možnost přidání externího kalendáře pomocí URL. Proces zapsání se na určitou hodinu byl však velmi matoucí. Samotné zobrazení těchto schůzek v kalendáři je nepřehledné, musíte rozkliknout každou událost zvlášť, aby bylo možné vidět, kdo se zapsal na schůzku a v kolik hodin. Tento kalendář spíše vhodný pro plánování osobních záležitostí, avšak ne pro vytváření časových slotů pro schůzky.

Výhody:

- široká škála funkcí
- neplacená

Nevýhody:

- nepřehlednost

■ 2.3.2 Appointlet

Tato aplikace [4] obsahuje vše potřebné k vytvoření schůzky a také k zapsání se na ni, přitom je ale aplikace minimalistická a zachovává přehlednost a snadnost při používání. Umožňuje nastavit u schůzky parametry jako například počet možných studentů na dané schůzce. Dále nastavení časové dostupnosti, místa, ceny a způsobu platby. Naplánované schůzky jsou sice přehledně sepsané, avšak zde není možnost zobrazení schůzek v kalendáři.

Výhody:

- uživatelsky přívětivý interface
- mnoho parametrů u schůzky, které se dají nastavit
- neplacená

Nevýhody:

- absence samotného kalendáře

■ 2.3.3 TutorBird

Plánovací kalendář TutorBird [5], který byl specificky vytvořen pro doučování. Aplikace obsahuje mnoho funkcí kromě základních jako zapsání se na určitou událost, které mohou lektori využít. Je možné zobrazit ujetou vzdálenost za studentem, která se pak může připočítat do ceny doučovací hodiny. Mohou ručně přidat seznam studentů. Odtud se dají zobrazit profily studentů, které obsahují informace o počtu hodin odstudovaných u daného lektora, docházce a posláním zprávy. Je zde rovněž záložka s online zdroji, kde má učitel možnost nahrát soubory pro určitou skupinu nebo konkrétního studenta. Nalezneme zde také záložku s půjčením knih a možnost vystavit knihu k vypůjčení.

Výhody:

- obsahuje zajímavé funkcionality, které jsou zaměřené pro lektory

Nevýhody:

- placená aplikace, požadují \$14.95 USD za měsíc

■ 2.3.4 Calendly

Kalendář pod názvem Calendly [6], který umožňuje stanovit dostupnosti během týdne, můžete jich stanovit více pro různé události, což zrychlí vytváření událostí pro každý vyučovaný předmět. Při vytváření události totiž můžete zvolit určitou dostupnost, kterou jste stanovili.

Výhody:

- uživatelsky přívětivý interface

Nevýhody:

- v neplacené verzi je možnost vytvoření pouze jedné aktivní události, na kterou se dá zapisovat
- v neplacené verzi není možnost vytvářet události pro skupiny

■ 2.3.5 FelSight

Tato webová aplikace [7] je určena především pro příslušníky ČVUT FEL. Disponuje kalendářem, který umožňuje zobrazení vlastního rozvrhu. Taktéž umožňuje zobrazení rozvrhu vašich spolužáků a vyučujících ve vašem rozvrhu. Je tu možnost přidat událost nebo úlohu a doplnit určité parametry jako například čas, místo, opakování, náležitost k určitému předmětu, které máme daný semestr zapsány. Tyto události však slouží spíše pro vlastní účely, tedy se na ně nedá zapisovat.

Výhody:

- intuitivní uživatelský interface

Nevýhody:

- není možnost se zapisovat na události

■ 2.3.6 Doučuji.eu

Webová aplikace, která umožňuje vyhledávat doučování dle předmětu, lokality, ceny a preferovaného času. Domluvení se na hodině probíhá pomocí odeslání poptávky se zájmem o doučování, dále komunikace probíhá pomocí emailu nebo v chatu.

Výhody:

- filtrování doučovacích hodin dle parametrů
- neplacená

Nevýhody:

- absence kalendáře
- nepřesné časové údaje časové preference

Kapitola 3

Analýza

3.1 Vytvoření persony

Pro vytvoření jakékoli aplikace je dobré porozumět, kdo bude naším uživatelem. Persona představuje fiktivní reprezentaci ideálního zákazníka. Zahrnuje potřeby, cíle a chování určité cílové skupiny. Dané požadavky nám pomáhají si ujasnit funkcionality, které by aplikace měla obsahovat. [8]

■ Persona 1

Jan Novák, 32 let, učitel jazyků, v současnosti bez stálého zaměstnání. Má plán doučovat angličtinu a přál by si vytvořit klientelu studentů prostřednictvím internetové aplikace. S doučováním nemá praktické žádné zkušenosti.

Jeho cíle:

- Věnovat se doučování na plný úvazek a vydělat si na živobytí.

Jeho požadavky:

- Možnost sledovat nadcházející hodiny v kalendáři a evidovat platby.

Jeho omezení:

- Nechce strávit moc času administrací doučování.

■ Persona 2

Anna Kyselá, 55 let, učitelka chemie a fyziky, na gymnáziu na plný úvazek. Chtěla by si prostřednictvím doučování přilepšit. Má dlouhodobé zkušenosti s doučováním.

Její cíle:

- Věnovat se doučování ve svém omezeném volném čase.

Její požadavky:

- Přívětivé uživatelské rozhraní

Její omezení:

- Ovládání počítače ji dělá problémy.

■ Persona 3

Petr Černý, 16 let, student gymnázia. Má špatné známky z chemie, a proto mu rodiče uložili hledat doučování. Petra nic, co je nějak spojeno s doučováním, nebaví.

Jeho cíle:

- Mít klid od rodičů.
- To znamená, s co nejmenší námahou zlepšit známky z chemie,

Jeho požadavky:

- Ať je organizace doučování časově vhodná, rychlá a bezproblémová.

Jeho omezení:

- Nezájem ohledně organizace doučování.

■ Persona 4

Olga Kopecká, 26 let, čerstvá absolventka právnické fakulty. Má přítele, Itala, a ráda by se co nejlépe naučila italsky, aby mohla ucházet o zaměstnání v nějaké italské advokátní kanceláři.

Její cíle:

- Naučit se italsky.

Její požadavky:

- Dlouhodobé doučování, možnost platit online.

Její omezení:

- Žádné.

3.2 Funkční požadavky

Na základě rešerše existujících řešení, mé osobní zkušenosti a vytvořených person jsem identifikovala následující funkční požadavky:

- FRQ01 Aplikace bude umožňovat všem uživatelům si zobrazit kalendář.
- FRQ02 Aplikace bude umožňovat všem uživatelům si zobrazit rezervované hodiny v podobě seznamu.
- FRQ03 Aplikace bude umožňovat lektorům si nastavit časovou dostupnost.
- FRQ04 Aplikace bude umožňovat lektorům automaticky vytvořit dostupné časové sloty pomocí importu externího kalendáře z URL.

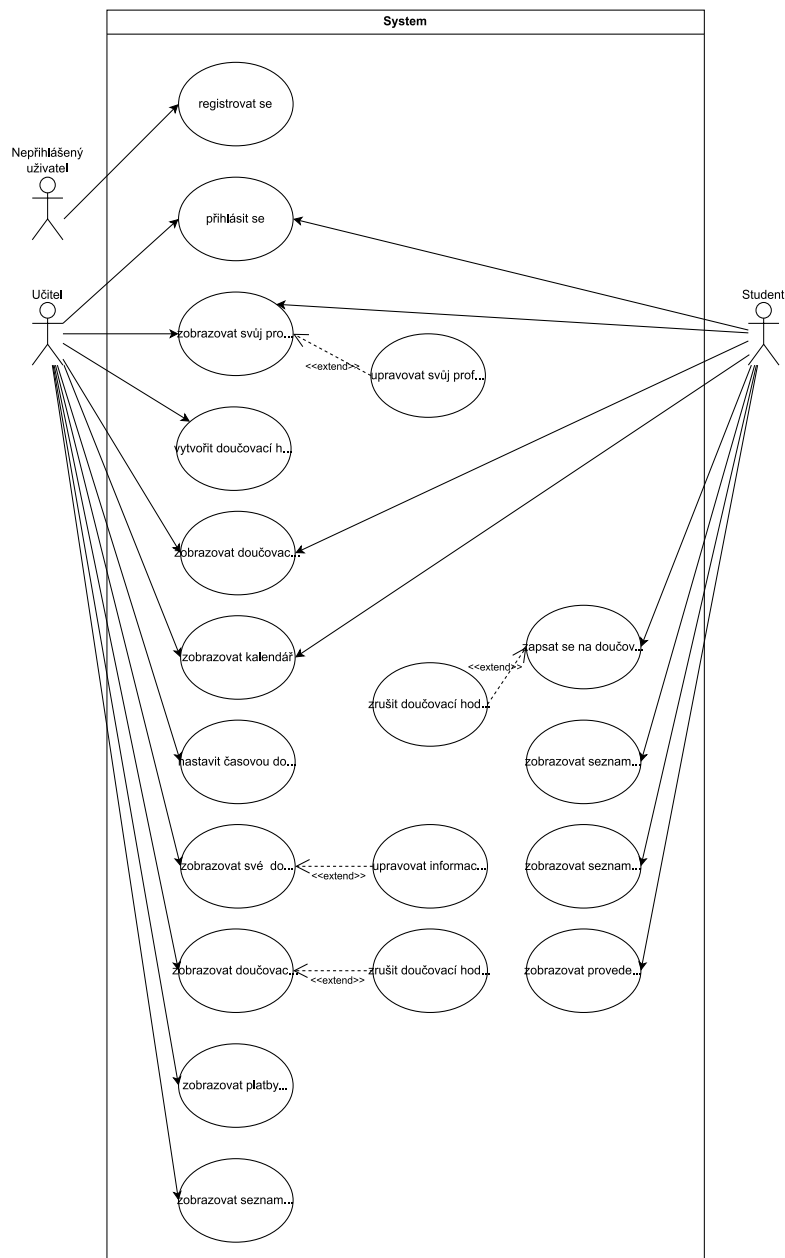
- FRQ05 Aplikace bude umožňovat lektorům vytvořit doučovací časové sloty.
- FRQ06 Aplikace bude umožňovat lektorům si zobrazit seznam svých studentů.
- FRQ07 Aplikace bude umožňovat lektorům zobrazit seznam hodin, na které se studenti přihlásili.
- FRQ08 Aplikace bude umožňovat lektorům zrušit doučovací hodinu.
- FRQ09 Aplikace bude umožňovat lektorům upravit doučovací hodinu.
- FRQ10 Aplikace bude umožňovat lektorům zobrazit seznam plateb.
- FRQ11 Aplikace bude umožňovat lektorům si poznamenávat absenci na výukových kurzech pro skupiny.
- FRQ12 Aplikace bude umožňovat studentům se zapsat do vyučovacího časové slotu.
- FRQ13 Aplikace bude umožňovat studentům provést platbu za doučovací hodinu.
- FRQ14 Aplikace bude umožňovat studentům si zobrazit seznam doučovacích hodin.
- FRQ15 Aplikace bude umožňovat studentům si zobrazit seznam svých doučovatelů.
- FRQ16 Aplikace bude umožňovat studentům si zobrazit seznam dostupných doučovacích hodin.
- FRQ17 Aplikace bude umožňovat studentům filtrovat dostupné doučovací hodiny na základě předmětu, lokace a ceny.
- FRQ18 Aplikace bude umožňovat studentům filtrovat zapsané doučovací hodiny nadcházejících a proběhlých.
- FRQ19 Aplikace bude umožňovat studentům zrušit zapsanou doučovací hodinu.

3.3 Nefunkční požadavky

- NFRQ01 Aplikace bude vytvořena tak, aby byla v budoucnu snadno rozšiřitelná o novou funkcionalitu.
- NFRQ02 Aplikace bude vytvořena tak, aby její používání bylo snadné a intuitivní.
- NFRQ03 Aplikace bude vytvořena tak, aby bylo možné ji používat ve všech webových prohlížečích.

3.4 Diagram případů užití

Po identifikování funkčních požadavků systému, byl vytvořen diagram případů užití, který zachycuje aktivity, které mohou provádět aktéři v rámci systému dle jejich role.



Obrázek 3.1: Diagram případů užití

Kapitola 4

Návrh uživatelského rozhraní

V následující kapitole rozebereme pojem mentálního modelu a jeho využití při návrhu uživatelského rozhraní. Dále definujeme zákony použitelnosti dle Steva Kruga. Na základě mentálního modelu a těchto zákonů bude vytvořen drátěný model webové aplikace a celková mapa interakce uživatele.

4.1 Mentální model

Mentální model představuje uživatelské předpoklady o tom, jak funguje určitý produkt, například webová aplikace. Tyto modely se u uživatelů utváří na základě zkušeností s používáním podobných produktů. Dalšími faktory, ovlivňující daný model, mohou být vzdělání, věk nebo kultura. Na základě toho si každý uživatel utváří svůj vlastní model. [9]

Vytváření produktu na základě uživatelských znalostí a představách o podobných produktech může přispět k bezmyšlenkovitým interakcím, rychlému osvojení a lepší použitelnosti produktu. [9] Z tohoto důvodu bude návrh uživatelského rozhraní vytvořen dle identifikovaných požadavků v kapitole 3 a existujících řešení v sekci 2.3.

4.1.1 Sjednocení mentálních modelů

Problémy s použitelností mohou vzniknout při vytvoření odlišného mentálního modelu u uživatele a návrháře. Při neshodě takových modelů můžeme produkt přizpůsobit předpokladům uživatele nebo uživatele poučit, a tím jednotlivé mentální modely sjednotit. [10]

Pokud více uživatelů naráží na podobný problém, je vhodné přizpůsobit produkt modelu uživatele.

Pokud se však jedná o ojedinělé neshody, může návrhář uživatele nasměrovat dle své představy. Tohoto se dá docílit například interaktivním průchodem aplikace nebo přidáním vysvětlivek.

4.2 Tři zákony použitelnosti

Steve Krug definuje použitelnost jako něco, co dokáže člověk pochopit a použít k dosažení nějakého výsledku. Formuloval tři zákony použitelnosti, které by se měly dodržet, pokud chceme dosáhnout dobré použitelnosti.

Zákon číslo 1: Nenuťte mě přemýšlet

Tento zákon považuje za klíčový. Určuje, zda náš návrh je vhodný či naopak. Pokud se uživatel podívá na webovou stránku, musí mu být zřejmé, o co se jedná a jak ji používat bez dlouhého přemýšlení. Každá otázka, která vznikne u uživatele a nutí ho přemýšlet, odvádí pozornost od provedení nějaké akce. [11]

Zákon číslo 2: Každé kliknutí musí být instinktivní

Na počtu kliknutí nezáleží, pokud je každé kliknutí bezmyšlenkovité a zřejmé. [11]

Zákon číslo 3: Zredukujte počet slov

Tento zákon radí odstranit přebytečná slova, která uživatele jen zbytečně rozptylují. Tímto umožníme vyniknutí klíčového obsahu a minimalizujeme potřebu posouvání webové stránky. [11]

4.3 Drátěný model

Proces návrhu aplikace začíná vytvořením drátěných modelů neboli wireframes, které umožňují vizualizovat rozložení elementů na jednotlivých stránkách aplikace.

Při vytváření drátěného modelu je nejlepší vytvářet drátěné modely s nízkou až střední věrností. Chceme poskytnout dostatečně detailní návrh, aby si jej dokázal každý představit. Nechceme však vytvářet něco zbytečně komplikovaného ihned ze začátku, aby bylo možné návrh v budoucnu jednoduše přizpůsobovat představám uživatele. [12]

4.3.1 Balsamiq

Balsamiq Wireframes je nástroj pro tvorbu low-fidelity drátěných modelů uživatelského rozhraní (modelů nízké věrnosti). Pro vytvoření drátěného modelu byl vybrán tento nástroj z toho důvodu, že umožňuje vytvořit podrobný, ale jednoduchý návrh. Nutí uživatele se soustředit na strukturu a obsah a vyhnout se přemýšlení o barvách a detailech, které by měly přijít později v procesu. [13]

4.4 Mapa interakce

Mapa interakce slouží k vizualizaci toho, jak jednotlivé wireframes spolu souvisí, tedy jak se bude moci uživatel pohybovat mezi jednotlivými stránkami navržené aplikace.

Na základě funkčních požadavků v kapitole 3 byl vytvořen low-fidelity drátěný model webové aplikace pro učitele.



Obrázek 4.1: Mapa interakce

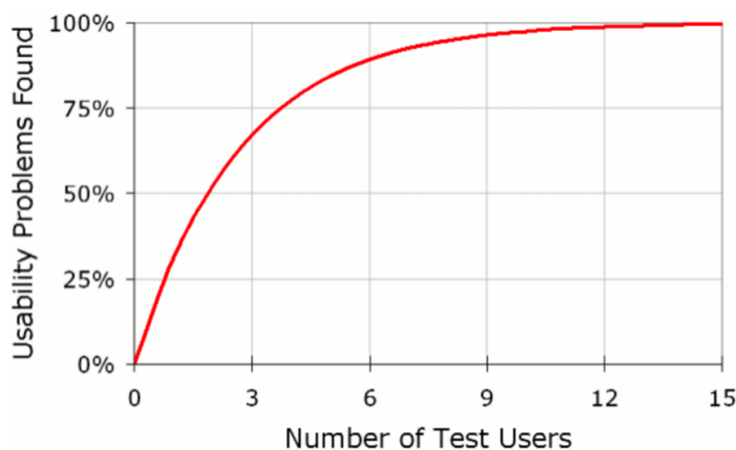
Kapitola 5

Testování návrhu uživatelského rozhraní

Cílem uživatelského testování použitelnosti je zhodnotit navržený produkt potenciálními uživateli, určit spokojenost s produktem a odhalit jeho případné nedostatky.

Návrhy obrazovek byly testovány na uživateli pomocí empirické evaluace. Dle Jakoba Nielsena pocházejí nejlepší výsledky testováním ne více než pěti uživateli. Shromážděním dat od prvního uživatele se dozvíme téměř třetinu potenciálních problémů. Testováním každého dalšího uživatele pak získáváme čím dále méně nových informací. [14] Z tohoto důvodu byly otestovány návrhy GUI pěti učiteli a pěti studenty.

Dle vzorce pro výpočet procenta odhalených chyb viz rovnice 5.1, dostaneme otestováním pěti uživatelů zaokrouhleně 84,36% úspěšnost odhalených chyb viz rovnice 5.2, což můžeme vypočítat i z obrázku 5.1. N určuje celkové množství problémů v návrhu a n značí počet testovaných uživatelů. L je množství nalezených problémů jedním uživatelem, které Jakob Nielsen udává jako hodnotu 31 %. [14]



Obrázek 5.1: Procento nalezených chyb testovanými uživateli

$$N = (1 - (1 - L)^n) \quad (5.1)$$

$$N = (1 - (1 - 0,31)^5) = 84,36\% \quad (5.2)$$

5.1 Způsob testování

Uživatelé měli k dispozici „klikací“ pdf soubor s návrhy obrazovek a formulář. Formulář obsahoval scénář s úkoly, které se měli uživatelé pokusit splnit. Po splnění či nesplnění každého úkolu následně ve formuláři zodpovídali otázky týkající se náročnosti úkolu a čas strávený na úkolu. Po dokončení všech úkolů následně byli dotázáni ohledně celkové spokojenosti s návrhem.

5.2 Uživatelské testování použitelnosti učitelů

Následující kapitola se zabývá uživatelským testováním použitelnosti učitelů.

5.2.1 Testovací scénář

Podrobný popis scénáře pro učitele:

Jste 32letý učitel a rádi byste nabízel doučování matematiky prostřednictvím internetu. Kolega Vám doporučil aplikaci Doučto.

Úkoly:

1. Vytvořte účet učitele, nastavte Vaše osobní údaje, profilovou fotografii a Vaši časovou dostupnost.
2. Nabídněte doučování matematiky pro jednotlivce v NTK Dejvice v Praze za 300 Kč.
3. Nastavte, že budete moci doučovat dva jednohodinové sloty v pondělí od 10:00-12:00 v období od 12. ledna 2023 po 15. února 2023.
4. Zkontrolujte pomocí kalendáře, kolik studentů se vám na hodiny přihlásilo.
5. Zobrazte si seznam Vašich aktuálních studentů a jaké doučovací hodiny si u Vás rezervoval Tomáš Novotný.
6. Potvrďte platbu doučovací hodiny.
7. Zobrazte si své hodnocení od studentů a odpovězte na komentáře.

5.2.2 Hodnocení náročnosti splnění úkolů

Tabulka 8.1 popisuje náročnost jednotlivých úkolů hodnocených studenty na stupnici od 1 do 5, kde 5 znamená „velmi snadné“.

Největší potíže dělali učitelům úkoly číslo 3, 2 a 4. Mírné potíže měli učitelé s úkoly číslo 7, 5 a 1. S úkolem číslo 6 neměl žádný učitel problém.

	Učitel 1	Učitel 2	Učitel 3	Učitel 4	Učitel 5	Náročnost úkolu
Úkol 1	5	5	4	5	5	4,8
Úkol 2	5	5	4	5	5	4,4
Úkol 3	5	5	4	5	5	4
Úkol 4	5	5	4	5	5	4,4
Úkol 5	5	5	4	5	5	4,8
Úkol 6	5	5	4	5	5	5
Úkol 7	5	5	4	5	5	4,6

Tabulka 5.1: Náročnost jednotlivých úkolů hodnocených učiteli

5.2.3 Čas strávený na úkolu

Tabulka 8.2 popisuje čas strávený studenty na každém úkolu. Nejvíce času strávili učitelé plněním prvního úkolu, což může být způsobeno tím, že při registraci opět prochází několik stránek drátěným modelem, kde se vyplňuje velké množství dat. Dále dost času strávili na druhém a třetím úkolu.

	Učitel 1	Učitel 2	Učitel 3	Učitel 4	Učitel 5	Průměrný čas strávený na úkolu
Úkol 1	<30	<90	<30	>120	<60	78
Úkol 2	<60	<30	<90	<120	<15	63
Úkol 3	<30	<90	>120	<60	<60	72
Úkol 4	<15	<30	<60	<30	<15	30
Úkol 5	<15	<15	<15	<15	<60	24
Úkol 6	<15	<30	<15	<15	<15	18
Úkol 7	<30	<15	<15	<15	<15	21

Tabulka 5.2: Průměrný čas učitelů strávený na úkolu ve vteřinách

5.2.4 Celková spokojenost

Silná většina respondentů byla velmi spokojena s návrhem této webové aplikace. Celková spokojenost učitelů s drátěným modelem byla 85% viz Tabulka 8.3.

	Učitel 1	Učitel 2	Učitel 3	Učitel 4	Učitel 5	Průměrná spokojenost
Spokojenost	100%	100%	75%	50%	100%	85%

Tabulka 5.3: Průměrná spokojenost učitelů s návrhem

■ 5.2.5 Požadavky na změnu

■ Úkol 1

V úkolu 1 byla zaznamenána připomínka, která upozorňuje na nevytvoření účtu, pokud se přeskočí při registraci výběr fotografie. Toto je způsobeno špatným mapováním odkazu u tlačítka „Přeskočit“.

■ Úkol 2

V úkolu 2 si dva respondenti mysleli, že bude moct splnit tento úkol pomocí „Nabídky doučování“. To bylo způsobeno nevhodným pojmenováním položky v menu.

■ Úkol 3

V úkolu 3 si většina z dotázaných učitelů myslela, že na kartě časová dostupnost bude možné provést změny ihned. Tlačítka „upravit“ si všimli až vzápětí. Objevila se zde také zmínka o tom, že se v kalendáři kombinoval český a anglický jazyk, což působilo dojmem převzetí části aplikace s neúplným překladem.

■ Úkol 4

V úkolu 4 měl jeden z učitelů potíže s nalezením aktuálního počtu studentů. Ten tam však není konkrétně uveden, jelikož se na kartě „Studenti“ zobrazí jednotlivá „okénka“ pro každého studenta s informacemi o něm. Počet těchto „okének“ tedy představuje počet studentů.

■ Úkol 5

U úkolu 5 se objevily pouze připomínky, a to ohledně toho, že jsem použila neobvyklý slovní výraz „dlužnost“. Byla tím myšlena částka, která zatím nebyla studentem uhrazena. Dále se zde vyskytla rada ohledně přidání možnosti přímé komunikace se studenty, formou chatu.

■ Úkol 6

S úkolem 6 žádné problémy nebyly, jen zde bylo zmíněno, že by se v budoucnu mohlo vyskytnout více možností formy platby.

■ Úkol 7

V úkolu 7 byla vytknuta nefunkčnost tlačítka „odeslat“. To bylo způsobeno tím, že chyběl odkaz na obrazovku s odeslaným komentářem.

■ 5.2.6 Celkové zhodnocení

U dotázaných nejčastěji vyvolala spokojenost přehlednost, jednoduchost a celkový design aplikace. Dále jim vyhovovala časová nenáročnost snadnost a užitečnost funkcionalit prostředí.

Co se týče výhrad, bylo namítnuto, že by v kalendáři mohla přidána informace o obsazenosti a kapacitě určitého slotu.

5.3 Uživatelské testování použitelnosti studenty

Následující kapitola se zabývá uživatelským testováním použitelnosti studenty.

5.3.1 Testovací scénář

Podrobný popis scénáře pro studenta:

Jste 16letý student, žijete v Praze a máte ve škole potíže s matematikou. Rádi byste našli doučování matematiky prostřednictvím internetu. Spolužák Vám doporučil aplikaci Doučto.

Úkoly:

1. Vytvořte účet studenta, nastavte Vaše osobní údaje, profilovou fotografii a Váš preferovaný způsob platby.
2. Najděte doučování matematiky v Praze do 300 Kč a zobrazte si více informací o vybraném doučování a hodnocení učitele.
3. Vyberte termín doučovací hodiny od 10:00-11:00, způsob platby kartou a rezervujte danou hodinu.
4. Zobrazte si rezervovanou hodinu a informace o ní ve Vašem kalendáři.
5. Zobrazte si seznam plateb a zaplaťte za rezervovanou hodinu u Jana Nováka.
6. Zobrazte si seznam Vašich aktuálních učitelů a Vaše rezervované termíny u Jana Nováka.
7. Zjistili jste, že se Vám termín nehodí. Vyberte nový termín u rezervované doučovací hodiny od 11:00-12:00.

5.3.2 Hodnocení náročnosti splnění úkolů

Tabulka 8.4 popisuje náročnost jednotlivých úkolů hodnocených studenty na stupnici od 1 do 5, kde 5 znamená velmi snadné.

Největší potíže dělali studentům úkoly číslo 3 a 7. Mírné potíže měli studenti s úkoly číslo 2 a 6. S úkoly číslo 1, 4, 5 neměl žádný student problém.

5.3.3 Čas strávený na úkolu

Tabulka 8.5 popisuje čas strávený studenty na každém úkolu. Nejvíce času strávili studenti plněním prvního úkolu, což může být způsobeno tím, že při registraci se prochází 3 stránky drátěným modelem, kde se vyplňuje velké množství dat. Ostatní úkoly až na výjimky byly spíše „klikací“, proto jejich splnění nevyžadovalo tolik času.

	Student 1	Student 2	Student 3	Student 4	Student 5	Náročnost úkolu
Úkol 1	5	5	5	5	5	5
Úkol 2	5	5	5	4	4	4,6
Úkol 3	5	4	5	4	4	4,4
Úkol 4	5	5	5	5	5	5
Úkol 5	5	5	5	5	5	5
Úkol 6	5	4	5	5	5	4,8
Úkol 7	5	4	5	3	5	4,4

Tabulka 5.4: Náročnost jednotlivých úkolů hodnocených studenty

	Student 1	Student 2	Student 3	Student 4	Student 5	Přůměrný čas strávený na úkolu
Úkol 1	<90	<90	<90	<30	<90	78
Úkol 2	<30	<15	<30	<60	<60	39
Úkol 3	<30	<60	>30	<60	<60	48
Úkol 4	<30	<15	<30	<15	<15	21
Úkol 5	<15	<15	<15	<30	<15	18
Úkol 6	<15	<15	<15	<15	<30	18
Úkol 7	<30	<30	<30	<30	<60	36

Tabulka 5.5: Průměrný čas studentů strávený na úkolu ve vteřinách

5.3.4 Celková spokojenost

Silná většina respondentů byla velmi spokojena s návrhem této webové aplikace. Celková spokojenost s drátěného modelu byla 95% viz Tabulka 8.6.

	Student 1	Student 2	Student 3	Student 4	Student 5	Průměrná spokojenost
Spokojenost	100%	100%	100%	75%	100%	95%

Tabulka 5.6: Průměrná spokojenost studentů s návrhem

5.3.5 Požadavky na změnu

■ Úkol 1

V úkolu 1 vznikly připomínky ohledně tlačítka „Přeskočit“ u výběru fotografie, jelikož se přeskočilo i zadávání údajů o kreditní kartě. Bylo to způsobeno chybným mapováním odkazu u tohoto tlačítka.

Další připomínkou byla absence možnosti registrace využitím třetí strany jako Google, Apple nebo Facebook. Tuto připomínku bych zahrнула do funkčních požadavků, které nebudou součástí implementace, ale v budoucnu se o tomto může uvažovat.

■ Úkol 2

V úkolu 2 byly připomínky ohledně toho, že je na stránce profilu učitele zajímavější více termíny doučování než detaily hodnocení, přitom pro úplný přehled termínů musím posouvat. Toto bylo způsobeno tím, že v Balsamiqu se mi nepodařilo zmenšit kalendář tak, aby se vešel na stránku, aniž by se stránkou muselo posouvat. Každopádně si myslím, že je toto případně možné vyřešit posunutím sekce s hodnocením učitele pod kalendář s termíny.

■ Úkol 3

V úkolu 3 se vyskytla připomínka ohledně tlačítka „Vybrat termín“ jelikož jej student přehlédl a snažil se kliknout na nadpis sekce „Termíny“. Vzhledem k tomu, že se jedná o low-fidelity wireframe, to je pochopitelné. Při implementaci návrhu toto tlačítko bude možné vizuálně zvýraznit například použitím barevného tlačítka.

■ Úkol 4

V úkolu 4 chyběla studentovi možnost, přejít na stránku s informacemi o učiteli, či zapsanou hodinu zrušit přímo v kalendáři při kliknutí na rezervovanou hodinu. S touto připomínkou souhlasím, bylo by vhodné tyto funkční požadavky zahrnout do návrhu.

■ Úkol 6

V úkolu 6 student nevěděl, co znamená zelená tečka na kartě učitele. To bylo způsobeno tím, že jsem danou funkcionalitu dostatečně nepopsala. Bylo by vhodné při přejetí nad tečkou zobrazit informace o dostupnosti učitele.

■ Úkol 7

V úkolu 7 studenta zmátlo to, že při kliknutí na rezervované termíny, byla možnost termín pouze zrušit, ale když se tam dostal přes učitele a následně rezervované termíny, tak tam možnost výběru jiného termínu byla. Bylo to způsobeno tím, že jsem využila chybný odkaz pro stránku s rezervovanými termíny, která toto tlačítko ještě neobsahovala.

■ 5.3.6 Celkové zhodnocení

Nejvíce byli studenti spokojeni s uživatelsky přívětivým a intuitivním prostředím, jednoduchostí a přehledností návrhu. Úkoly popisují jako snadno splnitelné i bez předchozí znalosti návrhu.

Nejméně byli studenti spokojeni s tím, že vše probíhalo bez možnosti kontaktu (chatu) pedagoga. Proces rezervace se jim tudíž zdál neosobní, ale zato rychlý. Také nevěděli, kde a jak mohou změnit platební metodu. Dále padl návrh přidat do aplikace notifikace.

Kapitola 6

Technologie

V této kapitole budou rozebrány technologie, které budou použity v rámci implementace webové aplikace.

6.1 Backend

Jelikož se tato práce věnuje převážně vytvoření frontendu, budou zde rozebrány nástroje, které umožňují implementaci backendu bez nutnosti vytvoření client-server architektury. Na základě poskytovaných funkcionalit a nalezených omezení těchto nástrojů bude zvolen nástroj, který bude použit v implementaci.

6.1.1 Json Server Api

Json Server Api je nástroj, který umožňuje vytvořit mock REST API, tedy vytvořit endpointy, na které se může dotazovat pomocí standardních metod HTTP. [16]

Samotnou databázi představuje json soubor, který obsahuje data, která se mění na základě prováděných CRUD operací.

Tento nástroj umožňuje:

- filtrování a řazení dat pomocí query parametrů v požadavku
- pomocí parametru `__page` a `__limit` v požadavku je možné implementovat stránkování
- možnost implementace autentikace nebo autorizace
- vytvoření vlastních routes

■ Vytvoření souboru JSON

```
"teachers": [  
  {  
    "id": 1,  
    "firstname": "Petr",  
    "lastname": "Rada",  
    "email": "petr.rada@example.com",  
    "studentId": 1  
  }  
],  
"students": [  
  {  
    "id": 1,  
    "firstname": "Jan",  
    "lastname": "Novak",  
    "email": "jan.novak@example.com",  
    "teacherId": 1  
  },  
  {  
    "id": 2,  
    "firstname": "Pepa",  
    "lastname": "Novy",  
    "email": "pepa.novy@example.com",  
    "teacherId": 1  
  }  
]
```

■ Spuštění serveru

Nastavíme sledování změn pomocí parametru `-w` pro json soubor s daty. Parametrem `-p` stanovíme port, na kterém chceme data sledovat. Tento soubor se aktualizuje na základě prováděných operací.

```
json-server -w data/db.json -p 3500
```

■ Zobrazení dat v prohlížeči

Nyní můžeme získat tyto data na URL adrese `http://localhost:3500`. Json Server Api vytvoří jednotlivé endpointy na základě struktury ze souboru `data.json`, na se které můžeme dotazovat pomocí REST API metod. Pokud chceme například získat data o studentech, tak se dotážeme na URL adresu `http://localhost:3500/students`, která nám vrátí

```
{
  "id": 1,
  "firstname": "Jan",
  "lastname": "Novak",
  "email": "jan.novak@example.com",
  "teacherId": 1
},
{
  "id": 2,
  "firstname": "Pepa",
  "lastname": "Novy",
  "email": "pepa.novy@example.com",
  "teacherId": 1
}
```

■ Nalezená omezení

Tento nástroj obsahuje omezení, která neumožňují vytvořit plně funkční mock aplikace. Nástroj totiž nepodporuje vytváření vazeb typu many to many mezi jednotlivými entitami.

■ 6.1.2 Pocketbase

PocketBase je open source backend vytvořený v jazyce go. Představuje vestavěnou databázi SQLite poskytující odběry v reálném čase, vestavěnou správu auth, uživatelským rozhraním dashboardu a rozhraní REST. [21]

■ SQLite

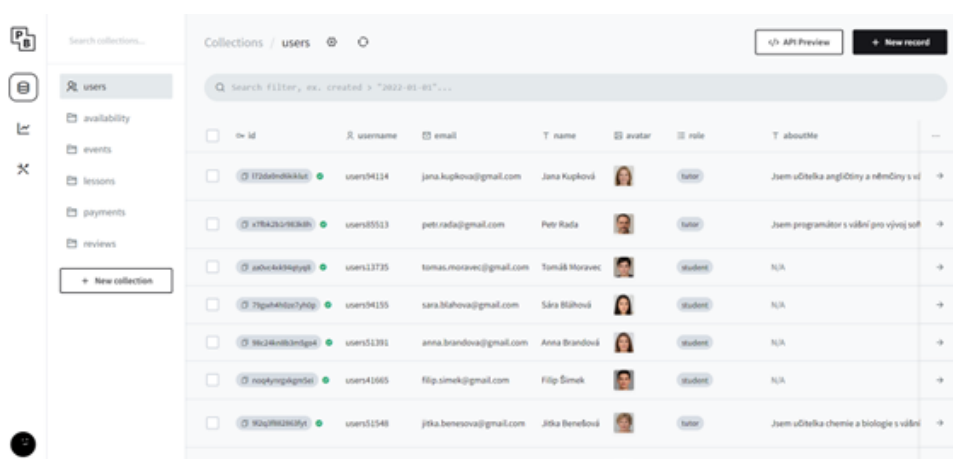
SQLite je relační databázový systém, který implementuje a spravuje relační databáze ve formě jednoho samostatného souboru. To znamená, že celá databáze je uložena v jediném souboru na disku. Svou architekturou tak poskytuje rychlý přístup k datům. [20]

SQLite je založen na standardní struktuře databázi s tabulkami, sloupci a relacemi. Pro práci s daty využívá standardní SQL.

■ Vytvoření dat

V Pocketbase se data vytváří pomocí tzv. kolekcí, které představují SQLite tabulky. Tabulky obsahují jednotlivé sloupce, u kterých stanovujeme datový typ. Řádky těchto tabulek pak nazýváme záznamy.

Pocketbase poskytuje WEB API neboli dashboard viz Obrázek 6.1, pomocí kterého můžeme spravovat databázi. Využitím tohoto dashboardu byly vytvořeny jednotlivé kolekce a záznamy.



Obrázek 6.1: Pocketbase dashboard

6.1.3 BaaS

Backend as a service (BaaS) je typ cloudové služby, který umožňuje propojovat aplikace s cloudovým úložištěm [15]. Tento typ služeb poskytuje vlastní SDK nebo API, pomocí kterých můžeme přistupovat k potřebným funkcionalitám jako správa dat nebo autentizace. Využitím této služby tedy nepotřebujeme vytvářet vlastní backendovou infrastrukturu, což umožňuje, plně se soustředit na frontend.

Tento typ služeb však vyžaduje poměrně složitou konfiguraci. Do těchto typů služeb spadá například Firebase ?? nebo Supabase ??.

6.2 Frontend

V následující kapitole budou rozebrány dva známé JavaScriptové frameworky pro tvorbu uživatelského rozhraní.

6.2.1 React

React byl vyvinut v roce 2011 zaměstnancem společností Facebook. Tento framework umožňuje vytvářet komponenty, které se dají znovupoužít. Používá JSX syntaxi, což je rozšíření syntaxe JavaScriptu, které umožňuje vkládat HTML do JavaScriptu. [18]

6.2.2 Vue

Vue bylo vyvinuto zaměstnancem společnosti Google v roce 2014, za účelem vytvořit lepší verzi frameworku Angular. Tento framework odděluje HTML, CSS a JavaScript, ale umožňuje využití i JSX. [19]

■ 6.2.3 Srovnání

Na GitHubu můžeme pozorovat, že Vue je o trochu více oblíbený mezi uživateli než React, jelikož získal 202 000 hvězd oproti Reactu, který jich získal 201 000. Více oblíbeným frameworkem dle NPM je však React. Přestože Vue má větší velikost produkčního buildu, tak je oproti Reactu zhruba dvakrát rychlejší v načtení stránky. [19]

■ 6.3 CSS frameworky

Vytvoření estetického designu aplikace nám v dnešní době mohou značně usnadnit a urychlit CSS frameworky. Tyto frameworky poskytují sadu předpřipravených HTML a CSS nebo i JS souborů. Jejich využitím můžeme docílit interaktivního a konzistentního designu.

■ 6.3.1 Vlastnosti CSS frameworků

Většina CSS frameworků disponuje následujícími vlastnostmi [17]:

- Grid
- Typografické prvky
- Cross-browser kompatibilita
- Navigační komponenty (navigační panel, boční panel, menu)
- Pomocné třídy pro umístování elementů
- Mediální prvky (odznaky, tooltipy, komentáře)
- Používají preprocesory jako Sass nebo LESS

■ 6.3.2 Bootstrap

Bootstrap je založen na předdefinovaných třídách a komponentách s přednastaveným vzhledem a stylem, který byl vytvořen v roce 2011 společností Twitter. Poskytuje šablony a třídy CSS pro komponenty, které kladou důraz na responzivitu a mobilní design. [24]

■ 6.3.3 Tailwind

Tailwind CSS vytvořili v roce 2017 Adam Wathan a Steve Schoger. Poskytuje nízkoúrovňovou sadu tříd, které se používají přímo ve značkách HTML. Spojením těchto tříd lze vytvořit jedinečné uživatelské rozhraní. [24]

■ 6.3.4 Flowbite

Flowbite kombinuje přístupy Bootstrapu a Tailwindu. Poskytuje sadu předdefinovaných komponent s interaktivními prvky s možností úpravy pomocí vlastního CSS. [25]

■ 6.4 Vybrané technologie

Pro implementaci backendu bude využit Pocketbase z důvodu snadné konfigurace.

Pro implementaci frontendu bude použit framework React z důvodu získaných zkušeností s tímto frameworkem. Pokud má již někdo základní ponětí o HTML a JavaScriptu, nebude mít dotyčný problém se React naučit během krátké doby. Navíc možnost práce s JSX je efektivní.

Pro implementaci frontendu bude využit Tailwind v kombinaci s Flowbite, jelikož to umožňuje vytvoření komponent, které mohou být snadno přizpůsobovány dle potřeb.

Kapitola 7

Implementace

7.1 Backend

Následující kapitola rozebírá implementaci serverové části aplikace.

7.1.1 Konfigurace

Pro využití Pocketbase je potřeba vytvořit instanci třídy `PocketBase` viz 7.1, kterou exportujeme, tedy umožníme její import do jiných souborů. Konstruktoru předáme URL adresu `http://127.0.0.1:8090`, která se použije pro komunikaci s PocketBase serverem.

```
1 import PocketBase from 'pocketbase';
2 const pb = new PocketBase('http://127.0.0.1:8090');
3 export default pb;
```

Tabulka 7.1: Vytvoření instance Pocketbase

7.1.2 AuthStore

SDK Pocketbase udržuje ověřený token a model auth prostřednictvím instance `pb.authStore`. Výchozí `pb.authStore` rozšiřuje `BaseAuthStore`, který umožňuje použít tyto parametry a funkce:

- `pb.authStore.isValid` - kontroluje, zda má úložiště platný token
- `pb.authStore.model` - obsahuje záznam přihlášeného uživatele
- `clear()` - odhlášení uživatele
- `save(token, model)` - aktualizace auth dat
- `onChange(callback, fireImmediately = false)` - zaznamenání callbacku při změně v úložišti

Zmíněné parametry a funkce byly využity pro přihlašování/odhlášení v aplikaci. V ukázce 7.6 získáváme stav `isLoggedIn` na základě hodnoty obsažené v `pb.authStore.isValid`. Pokud dojde ke změně tohoto tokenu, aktualizuje se stav `isLoggedIn`.

```
function App() {
  const [isLoggedIn, setIsLoggedIn] =
    ↪ useState(pb.authStore.isValid);

  pb.authStore.onChange(() => {
    setIsLoggedIn(pb.authStore.isValid);
  });
}
export default App;
```

Tabulka 7.2: Aktualizace stavu na základě změny tokenu

V tabulce 7.3 je znázorněn příklad odhlášení v aplikaci. Po kliknutí na tlačítko „Odhlásit se“ v bočním panelu, dojde v úložišti k smazání tokenu a odhlášení uživatele.

```
export default function SideBar() {
  const handleLogout = () => {
    pb.authStore.clear();
  }

  return(
    <Sidebar>
      <Sidebar.Items>
        <Sidebar.Item href="/" onClick={handleLogout}>
          ↪ >
            Odhlásit se
        </Sidebar.Item>
      </Sidebar.Items>
    </Sidebar>
  )
}
```

Tabulka 7.3: Odhlášení

Jak již bylo zmíněno, `pb.authStore.model` uchovává záznam přihlášeného uživatele. Pokud tedy máme validní token, můžeme získávat vlastnosti tohoto záznamu. Pomocí `pb.authStore.isValid` zjistíme, zda úložiště obsahuje platný token. Pokud to platí, můžeme následně nastavit roli přihlášeného uživatele pomocí `pb.authStore.model.role` viz 7.4. Na základě hodnoty role je pak vykreslováno tlačítko „Přihlásit se“.

```

export default function SideBar() {
  if (pb.authStore.isValid) {
    role = pb.authStore.model.role;
  }
  if (role === null) {
    return(
      <div>
        <Link to="/auth">
          <button>Přihlásit se</button>
        </Link>
      </div>
    )
  }
}

```

Tabulka 7.4: Použití `pb.authStore.model`

■ 7.1.3 CRUD operace

Následující část popisuje možné parametry dotazů a demonstruje ukázky použití CRUD operací při implementaci aplikace.

■ Parametry dotazů

Pocketbase umožňuje v dotazu specifikovat parametry:

- filter - filtrování dat vrácených záznamů
- expand - rozšíření záznamů
- sort - určení pořadí záznamů

■ Vytvoření záznamu

V tabulce 7.5 vytvoříme událost, které předáme parametry jako začátek, konec a tvůrce události, kterého nastavíme na základě přihlášeného uživatele.

■ Aktualizace záznamu

V tabulce 7.6 událost aktualizujeme. Ve funkci `update()` nastavíme jako první argument `id` události, kterou chceme aktualizovat. Do druhého parametru nastavíme `data`, která aktualizujeme.

```

const createEvent = async() => {
  await pb.collection('availability').create({
    start: args.start,
    end: args.end,
    creator: pb.authStore.model.id
  });
}

```

Tabulka 7.5: Vytvoření časové dostupnosti

```

const handleEventMoved = async (args) => {
  const e = args.e;
  await pb.collection('availability').update(e.data.id, {
    start: e.data.start,
    end: e.data.end
  })
}

```

Tabulka 7.6: Aktualizace časové dostupnosti

■ Získání záznamu

V tabulce 7.7 získáme všechny události pomocí funkce `getFullList()`, uvnitř této funkce pak filtrujeme události pomocí parametru `filter()` na základě jejího tvůrce. To nám tedy vrátí všechny události vytvořené námi.

```

const getAvailability = async() => {
  return await
  ↪ pb.collection('availability').getFullList({filter:
  ↪ `creator="${pb.authStore.model.id}"`});
}

```

Tabulka 7.7: Získání časové dostupnosti

■ Smazání záznamu

Pro smazání události použijeme funkci `delete()`, do argumentu této funkce dosadíme id události, kterou chceme smazat.

■ 7.2 Frontend

V následující kapitole bude rozbrán způsob implementace klientské části aplikace. Kapitola čerpá z dokumentace Reactu. [18]

```

const handleEventDelete = async (args) => {
  const eventId = args.e.data.id;
  try {
    await pb.collection("availability").delete(eventId);
  } catch (error) {
    return { error: error.message };
  }
}

```

Tabulka 7.8: Smazání časové dostupnosti

7.2.1 React Router

Pro navigaci v rámci aplikace byl využit React Router, který poskytuje směrování na straně klienta.

Směrování nastavíme pomocí Router v kořenové komponentě a vytvořením tras pomocí Route viz tabulka 7.9, kde nastavíme parametry jako je cesta a příslušná komponenta pro vykreslení.

```

const handleEventDelete = async (args) => {
  return (
    <Router>
      <Routes>
        <Route path="/" exact
          ↪ element={<Homepage/>}></Route>
        <Route path="/auth" exact element={<Auth/>} />
        <Route path="/calendar" exact
          ↪ element={<Calendar/>} />
        <Route path="/calendar/:id" exact
          ↪ element={<Calendar/>} />
      </Routes>
    </Router>
  );
}

```

Tabulka 7.9: React Routes

Pro navigaci mezi trasami pak používáme Link. V tabulce 7.10 nás Link přesměruje na přihlašovací stránku.

7.2.2 Vykreslování elementů

Elementy představují základní stavební prvky aplikace v Reactu. Na rozdíl od elementů DOM prohlížeče, React elementy představují obyčejné objekty. React DOM se stará o aktualizaci DOM tak, aby odpovídal elementům React.

```
const handleEventDelete = async (args) => {
  return(
    <Link to="/auth">
      <button>Přihlásit se</button>
    </Link>
  )
}
```

Tabulka 7.10: Tlačítko s Link

7.2.3 JSX

JSX je syntaktické rozšíření jazyka JavaScript, syntaxí podobný HTML, který umožňuje přidat do složených závorek jakýkoli validní JavaScriptový výraz.

JSX můžeme používat uvnitř if podmínek, for cyklů, přiřazovat ho proměnným a přijímat ho jako argumenty, jelikož se po kompilaci stanou běžnými voláními funkcí JavaScriptu a vyhodnotí se jako objekty JavaScriptu, které můžeme také vracet.

V ukázce níže 7.11 je demonstrováno použití JSX uvnitř for cyklu. Tato ukázka vrací div, který dynamicky vykreslí komponenty „Review“ v závislosti na stavu pole reviews.

```
return(
  <div>
    {reviews.map((review)=> {
      return(
        <Review review={review} key={review.id}
          ↪ onDelete={() => handleDelete(review.id)}/>
      )
    })}
  </div>
)
```

Tabulka 7.11: JSX

7.2.4 Komponenta

Komponenty umožňují rozdělit uživatelské rozhraní na nezávislé, opakovaně použitelné části. Validní komponenta vrací React element a obsahuje props neboli properties, které představují objekt s daty, který komponenta přijímá jako argument.

■ Class komponenta

Pro vytvoření class komponenty je nutno vlastní komponentu rozšířit o třídu `React.Component` a použít render funkci, která bude vracet React element. V class komponentách můžeme používat metody životního cyklu Reactu. K uchování stavu jsou použity konstruktory.

■ Funkční komponenta

Funkční komponenta představuje plain JavaScript funkci, která přijímá props a vrací JSX. Na rozdíl od class komponent je bezstavová a nepoužívá konstruktory, lifecycle metody ani funkci render.

Od Reactu 16.8 je však možné využívat React Hooks API, který umožňuje ve funkčních komponentách uchovávat stav pomocí `useState` podobně jako v class komponentách.

Při implementaci proto byly využity funkční komponenty, které jsou také více preferovány v novějších verzích Reactu a mají kratší syntaxi.

Následující ukázka 7.12 demonstruje použití funkční komponenty s props, konkrétně prop `children`. Tento prop umožňuje vnořování dalších komponent. Vnořené komponenty jsou pak předány `MyModal` 7.12 komponentě, které jsou pak zobrazeny uvnitř modálního okna. Komponenta `ViewEvent` 7.13 tedy zobrazí modální okno s nadpisem „Rezervovaná hodina“ a datumem.

```
export default function MyModal(props) {
  const [showModal, setShowModal] =
    ↪ useState(props.showModal);
  const toggleShow = () => {
    props.toggleShow();
  }

  return (
    <div onClick={toggleShow}>
      <Button>
        <XMarkIcon className="h-5 w-5
          ↪ text-gray-500" />
      </Button>
      {props.children}
    </div>
  )
}
```

Tabulka 7.12: Modální okno

```

export default function ViewEvent(props) {
  const [showModal, setShowModal] =
    → useState(props.showModal);

  return(
    <MyModal showModal={showModal}
    → toggleShow={props.toggleShow}>
      <h2>Rezervovaná hodina</h2>
      <Label htmlFor="date" value="Datum"/>
      <div id="date">{date}</div>
    </MyModal>
  )
}

```

Tabulka 7.13: Zobrazení události

7.2.5 Životní cyklus a stav

Stav uchovává data, která se mění během života komponenty. Ve funkčních komponentách můžeme uchovávat stav pomocí `useState`. Pokud dojde ke změně stavu komponenty, dochází k překreslení komponenty.

Ve výchozím nastavení se `useEffect` spouští po každém vykreslení. Toto chování můžeme měnit na základě našich potřeb.

Přidáním prázdného pole jako druhý argument ve funkci `useEffect` zajistíme vykonání `useEffect` pouze jednou při prvním vykreslení viz 7.14.

```

export default function ViewEvent(props) {
  useEffect(() => {
    getAvailability().then(data => {
      displayData(data);
      setEvents(displayData(data));
    })
  }, [])
}

```

Tabulka 7.14: `useEffect` s argumentem prázdného pole

Uvedením proměnné v druhém argumentu funkce `useEffect` zajistíme překreslování komponenty, pouze pokud dojde ke změně uvedené hodnoty. V ukázce níže 7.15 tedy bude docházet k překreslení komponenty, pokud dojde ke změně hodnoty `id`.

Tohoto bylo při implementaci využito v rámci změny `id` v URL, kdy bylo nutné události aktualizovat dle nového `id` viz 7.15.


```
useEffect(() => {
  getEvents().then(data => {
    displayData(data);
    setEvents(displayData(data));
  })
}, [id]);
```

Tabulka 7.15: useEffect s argumentem

7.2.6 Podmíněné vykreslování

React umožňuje podmíněné vykreslování na základě stavu komponenty, kterého můžeme docílit použitím operátory if, if-else nebo logických operátorů.

Při implementaci aplikace toho bylo využito především pro vykreslování částí komponent v závislosti na roli přihlášeného uživatele viz níže uvedená tabulka 7.16.

V JavaScriptu platí:

- `true && výraz` se vyhodnotí na výraz
- `false && výraz` se vyhodnotí na false

Proto pokud výraz `role === "student"` je pravdivý, bude tlačítko „Ohodnotit učitele“ vykresleno. Pokud podmínka splněna není, tlačítko se nevykreslí.

```
return (
  <div>
    {role === "student" && <Button
      ↪ onClick={toggleShow}>Ohodnotit učitele</Button>}
  </div>
)
```

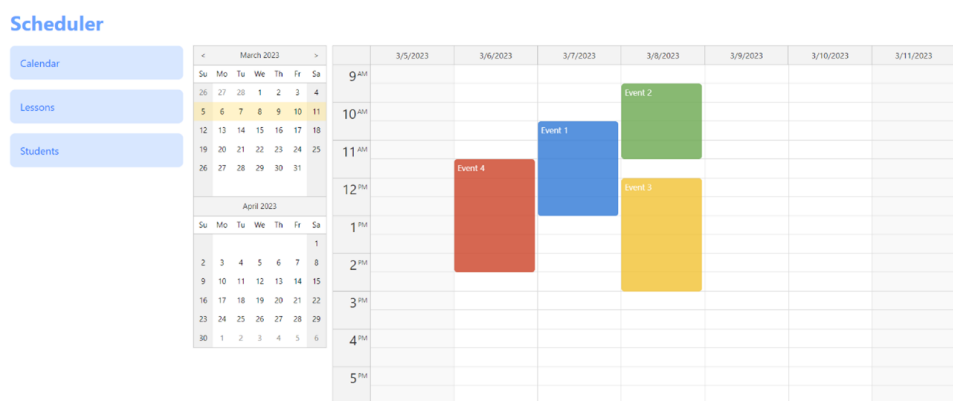
Tabulka 7.16: Logický operátor

7.3 Implementace kalendáře

7.3.1 DayPilot

Daypilot poskytuje komponenty, které usnadňují vytváření klientské části webových aplikací. Jednou z takových komponent je `CalendarComponent`. Tato komponenta byla dle poskytnuté dokumentace snadno integrována do projektu v Reactu.

Následující implementace vytvoří události specifikované v proměnné `events` 7.17. Po spuštění aplikace bude výsledkem obrazovka s kalendářem viz. Obrázek 7.1.



Obrázek 7.1: Integrace DayPilot komponenty

```

get calendar() {
  return this.calendarRef.current.control;
}
componentDidMount() {
  const events = [
    {
      id: 1,
      text: "Event 1",
      start: "2023-03-07T10:30:00",
      end: "2023-03-07T13:00:00"
    },
    {
      id: 2,
      text: "Event 2",
      start: "2023-03-08T09:30:00",
      end: "2023-03-08T11:30:00",
      backColor: "#6aa84f"
    }
  ];
  const startDate = "2023-03-07";
  this.calendar.update({startDate, events});
}
render() {
  return (
    <DayPilotNavigator
      selectMode={"week"}
      showMonths={2}
      skipMonths={2}
      startDate={"2023-03-07"}
      selectionDay={"2023-03-07"}
      onTimeRangeSelected={ args => {
        this.calendar.update({
          startDate: args.day
        });
      }}
    />
    </div>
    <div style={styles.main}>
      <DayPilotCalendar
        {...this.state}
        ref={this.calendarRef}
      />
    </div>
  );
}
export default Calendar;

```

Tabulka 7.17: DayPilot komponenta

Kapitola 8

Testování aplikace

Výsledná aplikace byla otestována potenciálními uživateli, tedy pěti učiteli a pěti studenty. Uživatelé měli k dispozici lokální verzi aplikace a formulář. Formulář obsahoval scénář s úkoly, které se měli uživatelé pokusit splnit. Po splnění či nesplnění každého úkolu následoval průzkum podobně jako při testování low fidelity návrhu obrazovek. Následně byla zpětná vazba vyhodnocena.

8.1 Testování aplikace učiteli

Následující kapitola se zabývá testováním aplikace učiteli.

8.1.1 Testovací scénář

Podrobný popis scénáře pro učitele:

Jste 32letý učitel a již nabízíte doučování prostřednictvím aplikace Doučto. Nově jste se naučil/a Španělštině a chcete ji doučovat.

Úkoly:

1. Přihlašte se.
2. Aktualizujte informace o sobě, tj. že nově umíte španělsky.
3. Vytvořte nový předmět, tj. španělštinu. Vyplňte údaje o novém předmětu.
4. Nastavte v kalendáři dostupnost, tj. kdy budete mít čas doučovat.
5. Jelikož již nějaké předměty doučujete, zjistěte, jaké hodiny Vás v blízké době čekají.
6. Zobrazte si seznam Vašich studentů a podívejte se na příchozí platby od Filipa Šimka.
7. Zjistěte, jaké předměty si u Vás Filip Šimek rezervoval.
8. Zrušte doučovací hodinu matematiky s Filipem Šimkem 27.5.2023 v 13:00.

9. Zobrazte si všechny platby a zjistěte, kdo Vám ještě nezaplatil za Matematiku dne 12.05.2023 v 13:30.
10. Zobrazte si svá hodnocení od studentů a odpovězte na nové komentáře.

■ Náročnost úkolu

Následující tabulka popisuje náročnost jednotlivých úkolů hodnocených učiteli na stupnici od 1 do 5, kde 5 znamená „velmi snadné“. Největší potíže dělali učitelům úkoly číslo 4, 5 a 7. Mírnější potíže byly zaznamenány u učitelů v úkolech číslo 3 a 6. S úkoly číslo 1, 2, 8, 9 a 10 neměl žádný učitel problém.

	Učitel 1	Učitel 2	Učitel 3	Učitel 4	Učitel 5	Náročnost úkolů
Úkol 1	5	5	5	5	5	5
Úkol 2	5	5	5	5	5	5
Úkol 3	3	4	5	4	5	4,6
Úkol 4	3	4	5	4	5	4,2
Úkol 5	5	3	4	5	4	4,2
Úkol 6	4	5	5	4	5	4,6
Úkol 7	3	5	5	4	4	4,2
Úkol 8	5	5	5	5	5	5
Úkol 9	5	5	5	5	5	5
Úkol 10	5	5	5	5	5	5

Tabulka 8.1: Náročnost jednotlivých úkolů hodnocených učiteli

■ Čas strávený na úkolu

V následující tabulce hodnoty vypovídají o průměrném čase stráveném učiteli na příslušných úkolech. Nejvíce času strávili učitelé plněním pátého úkolu, což může být způsobeno tím, že respondenti si nebyli jisti, co mají hledat. Řešením totiž mohly být dvě možnosti. Poměrně kratší dobu se zabývali vyplňováním úkolů číslo 2, 3, 4 a 7. Nejméně času pak učitelé strávili u prvního, šestého a desátého úkolu.

	Učitel 1	Učitel 2	Učitel 3	Učitel 4	Učitel 5	Průměrný čas strávený na úkolu
Úkol 1	<15	<30	<30	<30	<15	24
Úkol 2	<15	<60	<30	<30	<15	30
Úkol 3	<60	<60	<30	<60	<30	48
Úkol 4	<60	<30	<30	<60	<15	39
Úkol 5	<15	<90	<60	<60	<60	57
Úkol 6	<30	<15	<15	<30	<15	21
Úkol 7	<60	<15	<15	<30	<30	30
Úkol 8	<15	<15	<15	<15	<15	15
Úkol 9	<15	<15	<15	<15	<15	15
Úkol 10	<30	<15	<15	<15	<30	21

Tabulka 8.2: Průměrný čas učitelů strávený na úkolu ve vteřinách

■ Celková spokojenost

Silná většina byla velmi spokojena s návrhem této webové aplikace. V průměru byla celková spokojenost učitelů s aplikací 95 %.

	Učitel 1	Učitel 2	Učitel 3	Učitel 4	Učitel 5	Průměrná spokojenost
Spokojenost	100%	100%	100%	75%	100%	95%

Tabulka 8.3: Průměrná spokojenost učitelů s aplikací

■ Požadavky na změnu

Níže jsou uvedeny konkrétní úkoly, u kterých se objevily výtky.

■ Úkol 3

U třetího úkolu bylo potřeba doplnit dobu trvání doučování, kde však nebyly uvedené jednotky, ve kterých se to má doplnit. Dále zde jednoho z respondentů zmátlo tlačítko „Vytvořit hodinu“ Další námitkou bylo, že chybělo tlačítko „Zpět“.

- Úkol 4
U tohoto úkolu jeden vyučující narazil na problém ohledně vytvoření slotů delšího než 30 minut. Další respondent si všiml, že v kategorii „Nastavení dostupnosti“ byl zvýrazněn nadpis aktuální stránky, ale v předchozím úkolu tomu tak nebylo.
- Úkol 5
U pátého úkolu se několik respondentů rozhodlo pro nesprávnou kategorii v navigačním panelu („Předměty“ nebo „Kalendář“).
- Úkol 6
V tomto úkolu bylo trochu nejasné, kdo ne/zaplatil, kvůli chybějícím pojmenování jednotlivých sloupců.
- Úkol 7
Jeden z respondentů primárně vybral nesprávnou cestu přes kategorii „Předměty“.
- Úkol 8
Jeden z učitelů navrhl přidání potvrzovací dialog typu „Opravdu chcete zrušit událost?“.
- Úkol 9
V tomto úkolu se vyskytla připomínka ohledně popisu sloupce: „Do nadpisu by nebylo špatné přidat „a“ („Nezaplatil/a:“) k informacím o provedených či neprovedených platbách jednotlivými studenty.“
- Úkol 10
U posledního úkolu by jeden z respondentů uvítal popis tlačítka „Odeslat“.

■ Celkové zhodnocení

Nejvíce byli učitelé spokojeni s jednoduchostí a přehledností aplikace. Navigace přišla respondentům snadná.

Nejméně byli respondenti spokojeni s barvami, uvítali by změnu barev událostí v kalendáři.

■ 8.2 Testování aplikace studenty

Následující kapitola se zabývá testováním aplikace studenty.

■ 8.2.1 Testovací scénář

Podrobný popis scénáře pro studenta:

Jste 16letý student a nyní máte ve škole potíže nejen s matematikou, ale i fyzikou. Znáte aplikaci Doučto a chcete najít doučování.

Úkoly:

1. Přihlašte se.
2. Najděte učitele, který doučuje fyziku.
3. Vyberte pro Vás vhodný termín doučovací hodiny a rezervujte je.
4. Zobrazte si seznam doučovacích hodin, které Vás v blízké době čekají.
5. Zjistili jste, že se Vám termín nehodí. Zrušte doučování, které jste si právě rezervovali.
6. Zobrazte si seznam Vašich aktuálních učitelů a Vaše platby panu Richardu Novákovi.
7. Zobrazte si doučovací hodiny s Richardem Novákem.
8. Zkontrolujte jaké doučovací hodiny jste již zaplatili.
9. Napište hodnocení panu Richardu Novákovi.

■ Náročnost úkolu

V následující tabulce hodnoty vyjadřují náročnost jednotlivých úkolů hodnocených studentům na stupnici od 1 do 5, kde 5 znamená „velmi snadné“. Studenti považovali za nejobtížnější třetí úkol. O něco snazší pro ně byly úkoly číslo 2, 7 a 9. S úkoly číslo 1, 4, 5, 6 a 8 si studenti dokázali poradit bez jakýchkoli obtíží.

	Student 1	Student 2	Student 3	Student 4	Student 5	Náročnost úkolu
Úkol 1	5	5	5	5	5	5
Úkol 2	5	4	5	5	5	4,8
Úkol 3	3	4	4	4	3	3,6
Úkol 4	5	5	5	5	5	5
Úkol 5	5	5	5	5	5	5
Úkol 6	5	5	5	5	5	5
Úkol 7	5	4	5	5	4	4,6
Úkol 8	5	5	5	5	5	5
Úkol 9	3	5	5	5	5	4,6

Tabulka 8.4: Náročnost jednotlivých úkolů hodnocených studenty

■ Čas strávený na úkolu

Hodnoty v následující tabulce představují čas v průměru strávený studenty na jednotlivých úkolech. Nejdéle se studenti zabývali třetí otázkou, kde měli potíže s rezervací hodiny v kalendáři. Dále se zdrželi u úkolů číslo 1, 7 a 9. Nejkratší dobu se studenti zaobírali úkoly číslo 2, 4, 5, 6 a 8.

	Student 1	Student 2	Student 3	Student 4	Student 5	Průměrný čas strávený na úkolu
Úkol 1	<30	<30	<30	<30	<30	30
Úkol 2	<15	<30	<30	<15	<15	21
Úkol 3	<90	<60	<60	<30	<60	60
Úkol 4	<15	<15	<15	<15	<15	15
Úkol 5	<15	<15	<15	<15	<15	15
Úkol 6	<15	<15	<15	<15	<60	15
Úkol 7	<15	<60	<15	<15	<15	33
Úkol 8	<15	<15	<15	<15	<15	15
Úkol 9	<60	<60	<30	<30	<30	42

Tabulka 8.5: Průměrný čas studentů strávený na úkolu ve vteřinách

■ Celková spokojenost

Studenti byli s aplikací spokojeni z 90 %.

	Student 1	Student 2	Student 3	Student 4	Student 5	Průměrná spokojenost
Spokojenost	75%	100%	100%	100%	75%	90%

Tabulka 8.6: Průměrná spokojenost studentů s aplikací

■ Požadavky na změnu

Níže jsou uvedeny konkrétní úkoly, u kterých se objevily nějaké výtky.

- Úkol 2
Ve druhém úkolu většina respondentů nejdříve hledala v kategorii „Učitelé“. Chyba mohla být ve formulaci úkolu, ale i přesto by bylo vhodné přejmenování této kategorie na „Mí učitelé“.
- Úkol 3
Ve třetím úkolu nebylo některým respondentům zřejmé, jak danou rezervaci mohou vytvořit. Tento problém by se mohl odstranit použitím popisků.

- Úkol 5
Jeden z respondentů u tohoto úkolu navrhl přidání dialogového okna s možností potvrzení akce. S tímto návrhem souhlasím.
- Úkol 7
V sedmém úkolu respondent navrhoval chronologické seřazení nadcházejících událostí.
- Úkol 9
Jeden z respondentů u posledního úkolu zaváhal a omylem smazal hodnocení, které následně nemohl znovu vytvořit. Zde by bylo vhodné taktéž použít dialogové okno pro potvrzení akce.

■ Celkové zhodnocení

Nejvíce byli studenti spokojeni s designem a uspořádáním aplikace.

Nejméně byli respondenti spokojeni s tím, že měli problémy spojené s rezervováním hodiny v kalendáři.

Kapitola 9

Závěr

Cílem této práce bylo navržení uživatelského rozhraní pro webovou aplikaci poskytující organizaci doučování, otestování wireframes potenciálními uživateli, následná implementace, testování aplikace a vyhodnocení výsledků z uživatelského testování.

V rámci práce byla provedena analýza existujících řešení na současném zahraničním i českém trhu. Tato analýza byla podkladem pro identifikaci funkčních požadavků a také pro vytvoření návrhu webové aplikace. V nástroji Balsamiq byl vytvořen drátěný model (wireframe) webové aplikace.

Následně bylo vybráno pět studentů a pět učitelů pro otestování návrhu webové aplikace a vytvořeny testovací scénáře pro tyto cílové skupiny. Návrh byl otestován a ohodnocen pomocí dotazníku. V tomto dotazníku bylo zkoumáno, jak snadno a rychle se v návrhu naviguje, celková spokojenost uživatelů a jaké nedostatky obsahuje návrh.

Z výsledků dotazníku vyplývá, že studenti byli s návrhem aplikace spokojeni z 90 %, učitelé pak z 85 %. U dotázaných nejčastěji vyvolala spokojenost přehlednost, jednoduchost a celkový design aplikace. Nejméně byli studenti spokojeni s tím, že vše probíhalo bez možnosti kontaktu (chatu) pedagoga. Proces rezervace se jim tudíž zdál neosobní, ale zato rychlý.

Pro implementaci backendu této webové aplikace byl využit Pocketbase. Frontend byl implementován pomocí React, Tailwind a Flowbite s využitím DayPilot kalendáře.

V poslední fázi byla finální podoba aplikace taktéž podrobena uživatelskému testování. Byly vytvořeny testovací scénáře na základě implementovaného uživatelského rozhraní. Z výsledků bylo vyzorováno, že respondenti byli nejvíce spokojeni s designem a rozložením aplikace. Studenti se potýkali s komplikacemi při rezervování hodiny v kalendáři.

Ačkoliv se povedlo odhalit nedostatky i ve finální podobě aplikace, uživatelské rozhraní však splnilo očekávání respondentů. Procentuálně byli studenti uživatelským rozhraním s návrhem spokojeni z 90 %, učitelé z 95%.

Na závěr bych chtěla podotknout, že časově nejnáročnější byl proces vytvoření návrhu uživatelského rozhraní aplikace. Jelikož se musely vytvořit dva pohledy, které se lišily dle přihlášeného uživatele – studenta a učitele. Naopak proces stylování komponent byl zpracován poměrně rychle díky využití Flowbite.

V případném budoucím vývoji této práce by bylo vhodné se zaměřit na odstranění nalezených nedostatků v aplikaci a také na implementaci navržených funkcionalit respondenty v rámci testování, například možnost přihlášení do aplikace pomocí Google účtu. Další vývoj aplikace by mohl také spočívat v implementaci funkčních požadavků FRQ04 a FRQ11.



Literatura

- [1] PRŮCHA, J., WALTEROVÁ E. a MAREŠ J. *Pedagogický slovník. 6., aktualiz. a rozš. vyd.* Praha: Portál, 2009. ISBN 9788073676476.
- [2] Doučování. Hledáš nebo nabízíš doučování? | Doučuji.eu. *Doučování. Hledáš nebo nabízíš doučování? | Doučuji.eu* [online]. Copyright © 2023 [cit. 12.11.2022]. Dostupné z: <https://www.doucuji.eu/>
- [3] Kalendář Google: Online kalendář a aplikace pro plánování schůzek. [online]. Dostupné z: <https://workspace.google.com/products/calendar/>
- [4] Appointlet: Online Appointment Scheduling Software. *Appointlet: Online Appointment Scheduling Software* [online]. Copyright © 2022 Appointlet [cit. 12.11.2022]. Dostupné z: <https://www.appointlet.com/>
- [5] TutorBird: #1 Tutor Management Software. *TutorBird: #1 Tutor Management Software* [online]. Copyright © Copyright 2023 Port 443 Inc. [cit. 12.11.2022]. Dostupné z: <https://www.tutorbird.com/>
- [6] Free Online Appointment Scheduling Software | Calendly. *Free Online Appointment Scheduling Software | Calendly* [online]. Dostupné z: <https://calendly.com/>
- [7] FELSight 2.20.4 | Academic portal. *Document moved* [online]. Dostupné z: <https://felsight.fel.cvut.cz/welcome.xhtml>
- [8] VEAL, R. How to Define a User Persona. In: *CareerFoundry.com* [online]. CareerFoundry, 6.12.2022 [15.11.2022]. Dostupné z: <https://careerfoundry.com/en/blog/ux-design/how-to-define-a-user-persona/>
- [9] BENSON, S. Leveraging Mental Models in UX Design. In: *Toptal.com* [online]. 29. 8. 2019 [cit. 22.11.2022]. Dostupné z: <https://www.toptal.com/designers/ux/mental-models-ux-design>
- [10] NIELSEN, J. Mental Models. In: *Nngroup.com* [online]. Nielsen Norman Group, 17.10.2010 [cit. 25.11.2023]. Dostupné z: <https://www.nngroup.com/articles/mental-models/>

- [11] KRUG, S. *Don't make me think, Revisited: A Common Sense Approach to Web Usability*. 3rd Edition. San Francisco: New Riders, 2013. ISBN 9780133597271.
- [12] OSMAN, M. How to Create a Wireframe Map. In: *Blog.hubspot.com* [online]. HubSpot, 22.5.2020 [cit. 1.12.2022]. Dostupné z: <https://blog.hubspot.com/website/wireframe-map>
- [13] Balsamiq Wireframes - Industry Standard Low-Fidelity Wireframing Software | Balsamiq. *Balsamiq. Rapid, Effective and Fun Wireframing Software / Balsamiq* [online]. Copyright © their respective [cit. 1.12.2022]. Dostupné z: <https://balsamiq.com/wireframes/>
- [14] NIELSEN, J. Why You Only Need to Test with 5 Users. In: *Nngroup.com* [online]. Nielsen Norman Group, 18.3.2000 [cit. 7.1.2022]. Dostupné z: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>
- [15] LANE, K. Overview Of The Backend as a Service (BaaS) Space. In: *Apievangelist.com* [online]. API Evangelist, 3.5.2013 [cit. 10.5.2023]. Dostupné z: <https://apievangelist.com/2013/05/03/overview-of-the-backend-as-a-service-baas-space/>
- [16] GitHub - typicode/json-server: Get a full fake REST API with zero coding in less than 30 seconds (seriously). *GitHub: Let's build from here · GitHub* [online]. Copyright © 2023 GitHub, Inc. [cit. 26.05.2023]. Dostupné z: <https://github.com/typicode/json-server>
- [17] SHENOY, Aravind a Anirudh PRABHU. *CSS Framework Alternatives: Explore Five Lightweight Alternatives to Bootstrap and Foundation with Project Examples*. Berkeley, CA: Apress L. P, 2018. ISBN 9781484233986;
- [18] Getting Started – React. *React – A JavaScript library for building user interfaces* [online]. Copyright © 2023 Meta Platforms, Inc. [cit. 15.05.2023]. Dostupné z: <https://legacy.reactjs.org/docs/getting-started.html>
- [19] SAKS, E. *JavaScript frameworks: Angular vs React vs Vue* [online]. 2019 [21.01.2023]. Dostupné z: <https://www.theseus.fi/bitstream/handle/10024/261970/Thesis-Elar-Saks.pdf>. Bachelor's Thesis. Haaga-Helia University of Applied Sciences.
- [20] About SQLite. [online]. Dostupné z: <https://www.sqlite.org/about.html>
- [21] PocketBase - Open Source backend in 1 file. PocketBase - Open Source backend in 1 file [online]. Copyright © 2023 Pocket [cit. 10.05.2023]. Dostupné z: <https://pocketbase.io/>
- [22] Firebase. *Firebase* [online]. Dostupné z: <https://firebase.google.com/?gad=1>

- [23] The Open Source Firebase Alternative | Supabase. *The Open Source Firebase Alternative / Supabase* [online]. Copyright © Supabase Inc [cit. 11.05.2023]. Dostupné z: <https://supabase.com/>
- [24] Tailwind CSS vs. Bootstrap: Which is better? - Prismic. *Prismic: Headless Website Builder - Launch and Iterate Faster* [online]. Dostupné z: <https://prismic.io/blog/tailwind-vs-bootstrap>
- [25] Flowbite React Components. Flowbite React Components [online]. Dostupné z: <https://flowbite-react.com/>
- [26] GitHub - pocketbase/js-sdk: PocketBase JavaScript SDK. *GitHub: Let's build from here · GitHub* [online]. Copyright © 2023 GitHub, Inc. [cit. 16.05.2023]. Dostupné z: <https://github.com/pocketbase/js-sdk>
- [27] Tailwind CSS - Rapidly build modern websites without ever leaving your HTML. *Tailwind CSS - Rapidly build modern websites without ever leaving your HTML*. [online]. Dostupné z: <https://tailwindcss.com/>



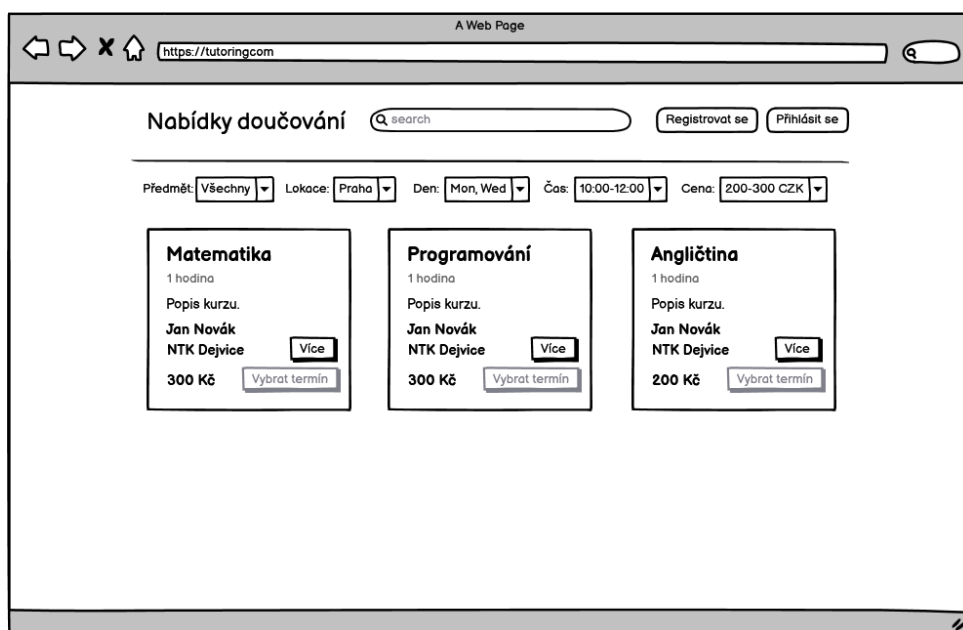
Příloha A

Seznam zkratek

Zkratka	Význam
FRQ	Functional Requirement
NFRQ	Non-Functional Requirement
JSX	JavaScript XML
JSON	JavaScript Object Notation
HTML	Hypertext Markup Language
NPM	Node Package Manager
API	Application Programming Interface
CSS	Cascading Style Sheets

Příloha B

Ukázka low fidelity prototypu



Obrázek B.1: Pohled pro nepřihlášeného uživatele

A Web Page
https://tutoring.com

Vytvoření účtu

* Jméno: * Příjmení:

* Jste učitel nebo student?
 učitel
 student

* Email: ⓘ

* Heslo:

Již máte vytvořený účet? [Přihášte se.](#)

Obrázek B.2: Registrační stránka

A Web Page
https://tutoring.com

Přihlášení

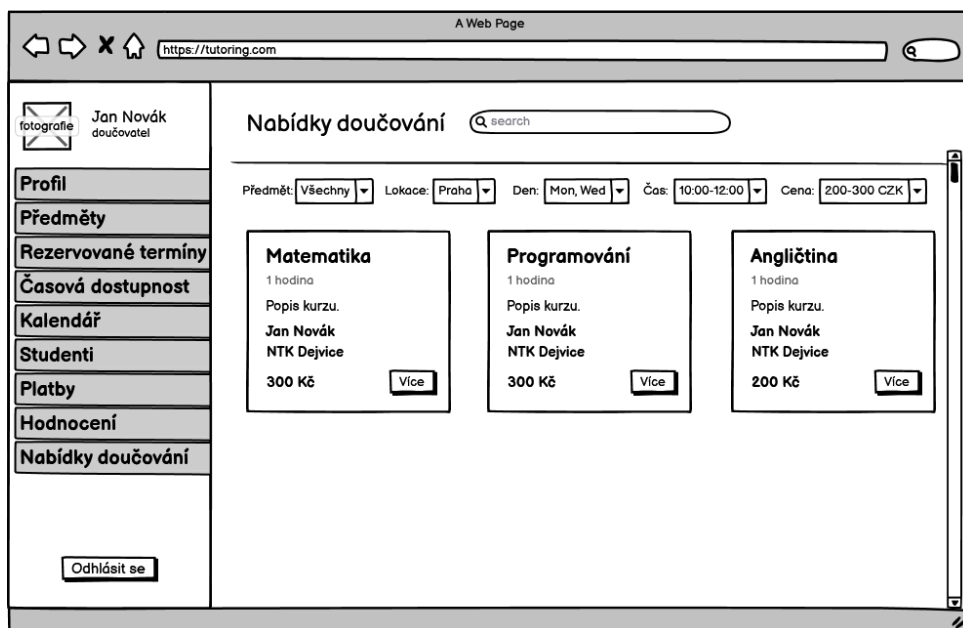
Email:

Heslo:

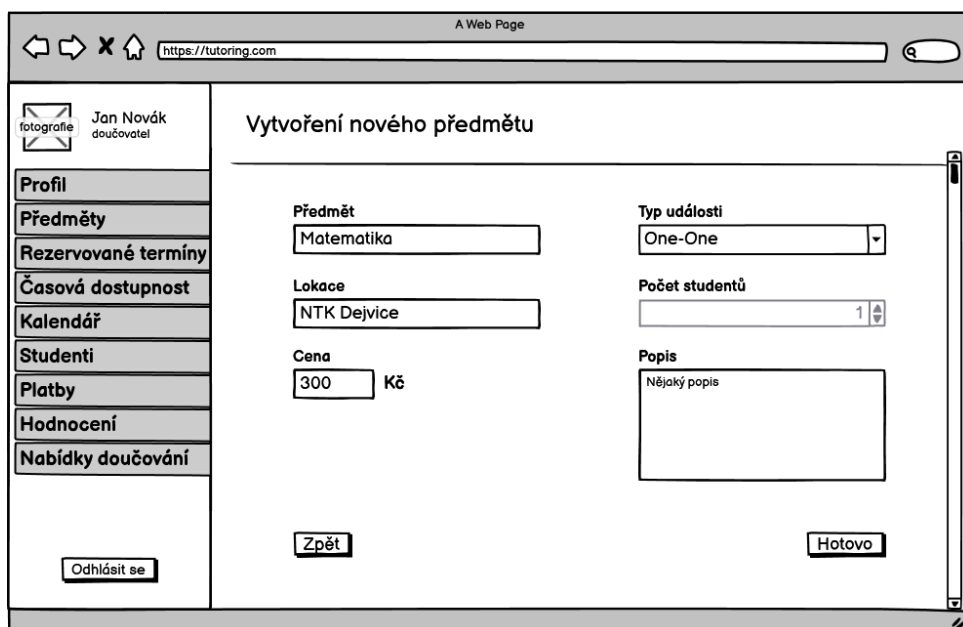
[Zapomenuté heslo?](#)

Nemáte vytvořený účet? [Registrujte se.](#)

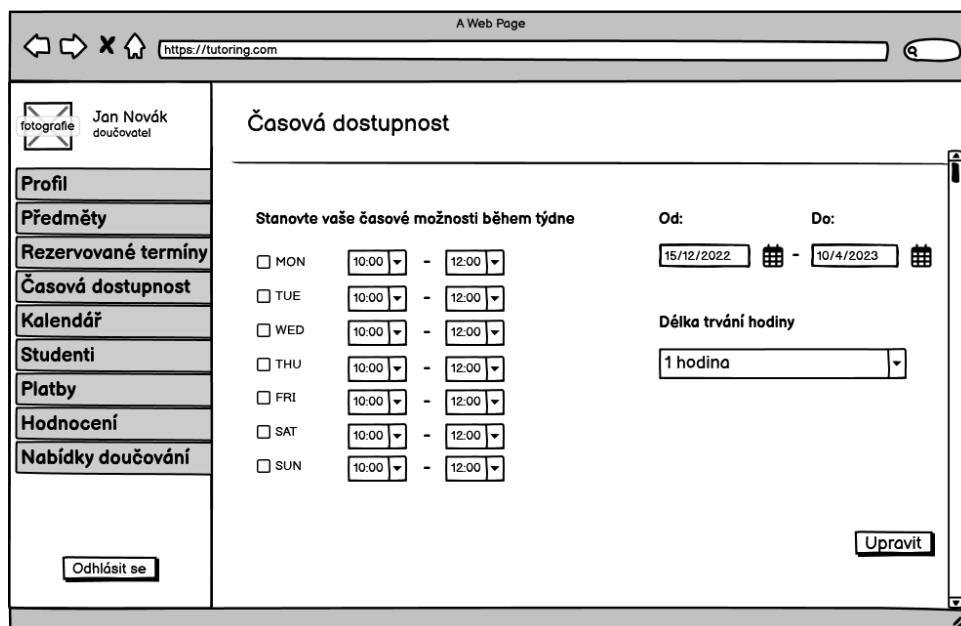
Obrázek B.3: Přihlašovací stránka



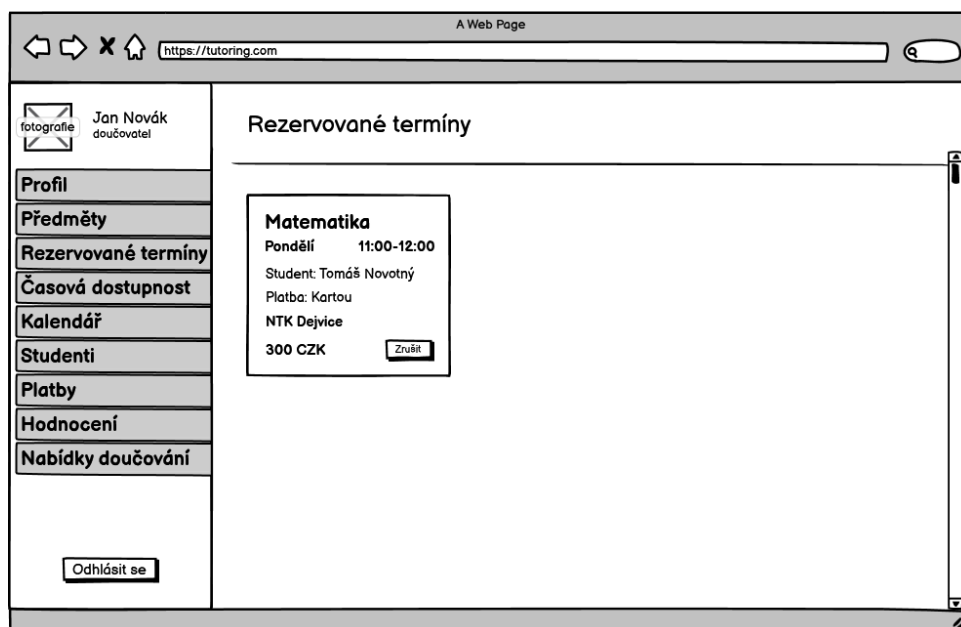
Obrázek B.4: Nabídky doučování



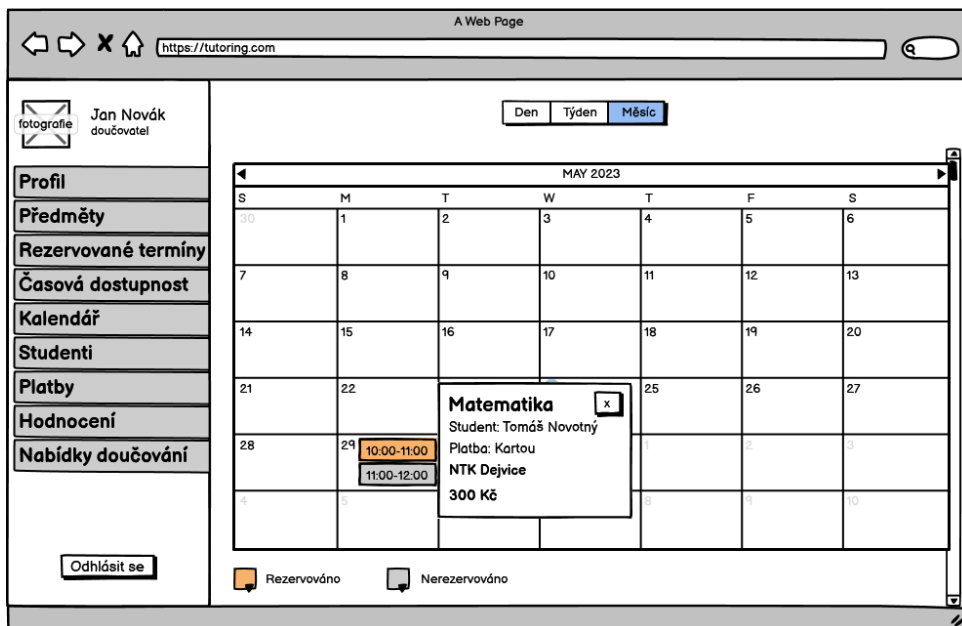
Obrázek B.5: Vytvoření předmětu



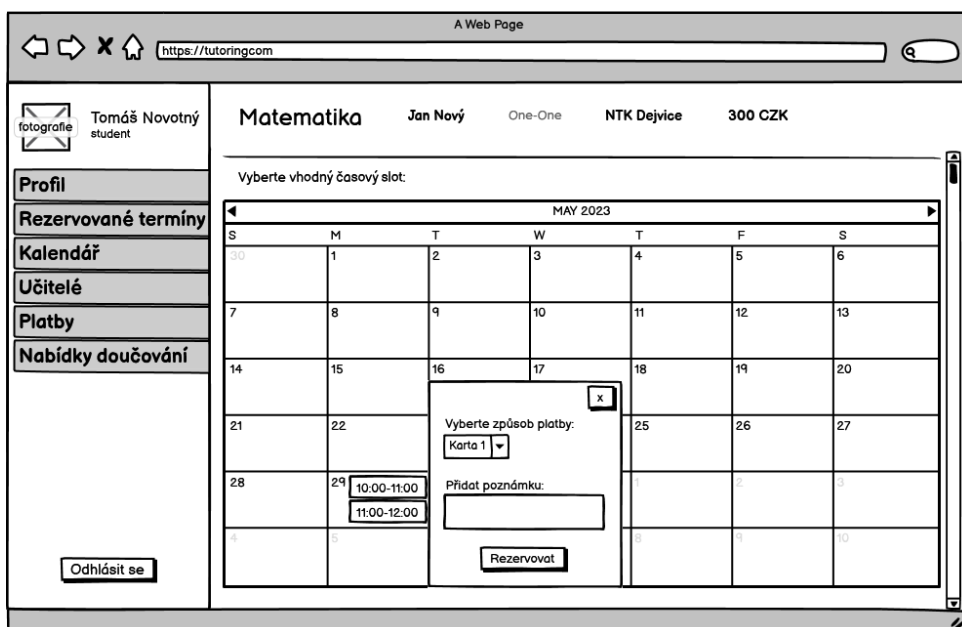
Obrázek B.6: Nastavení dostupnosti



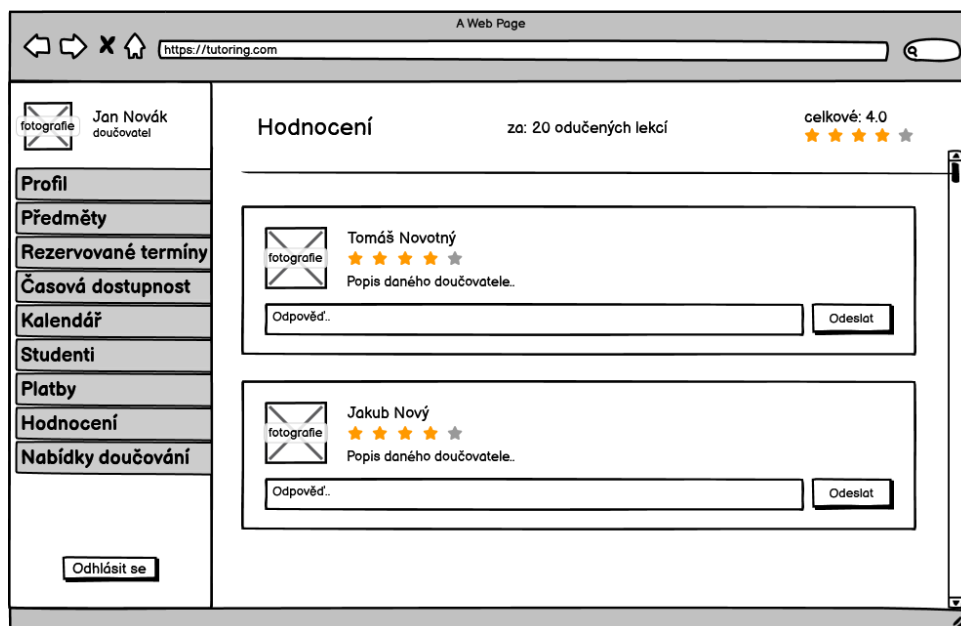
Obrázek B.7: Rezervované termíny



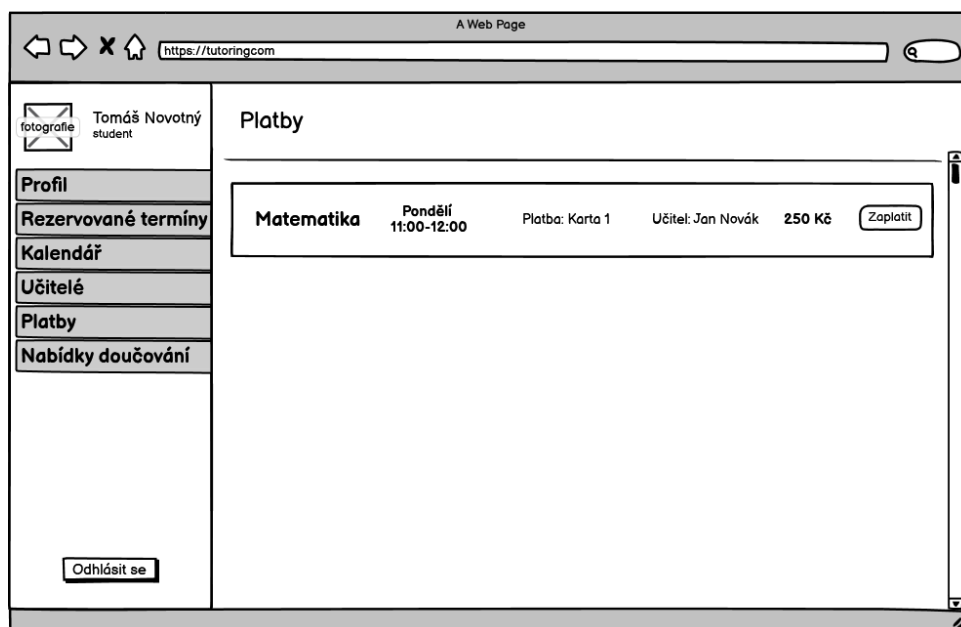
Obrázek B.8: Rezervované termíny v kalendáři



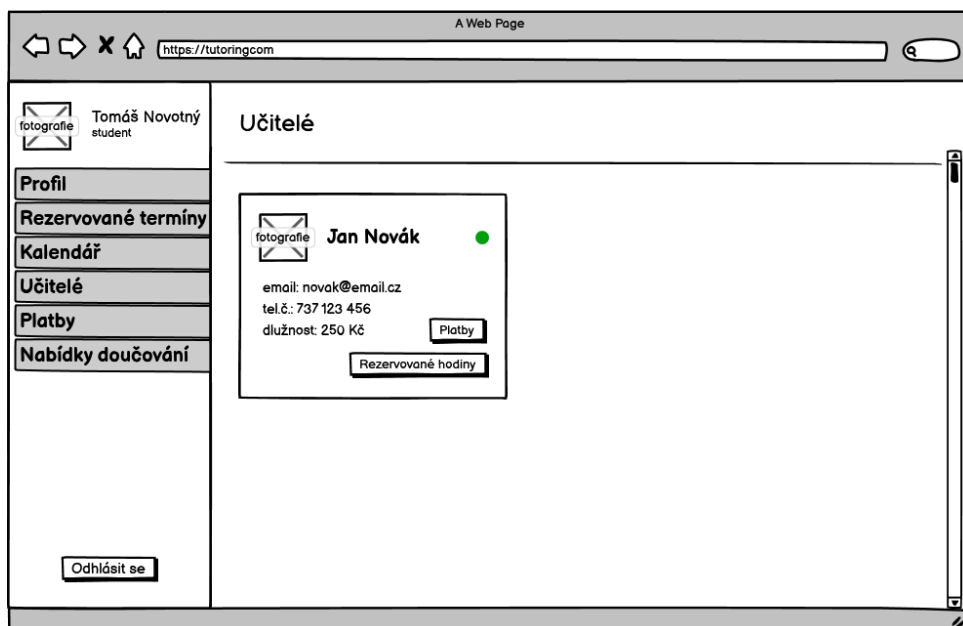
Obrázek B.9: Vybrání slotu pro rezervaci



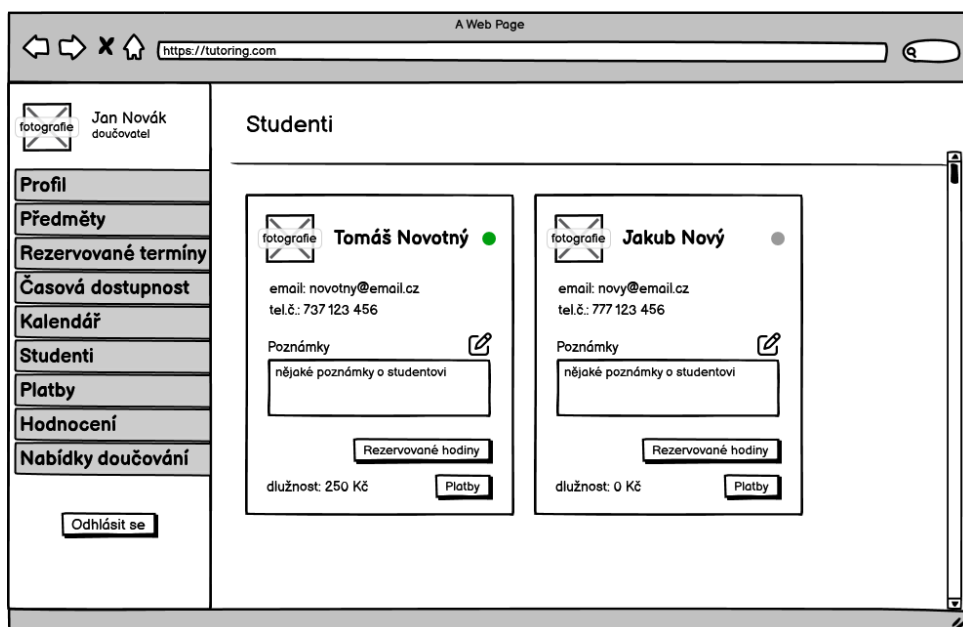
Obrázek B.10: Hodnocení učitele



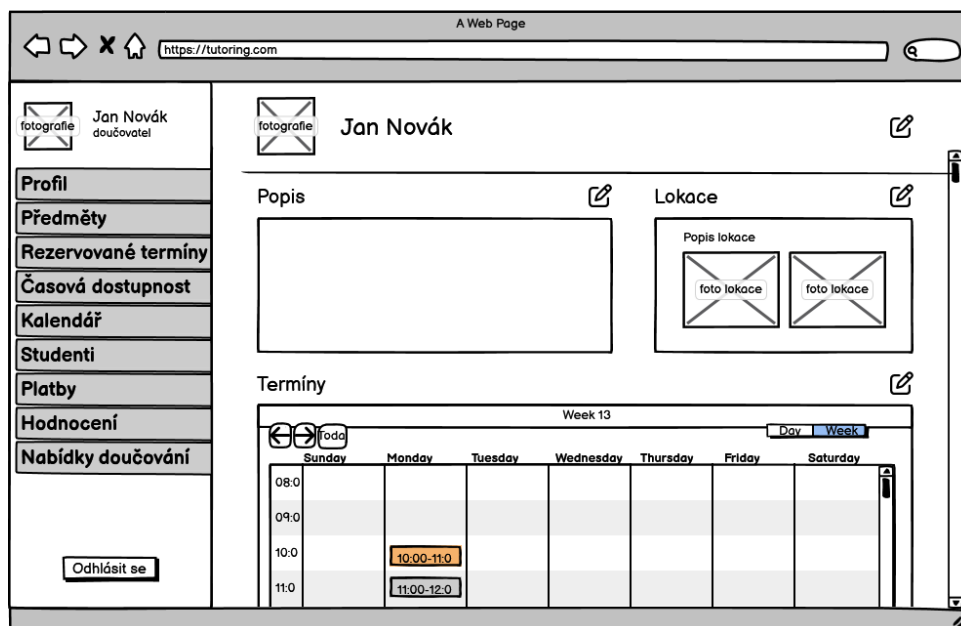
Obrázek B.11: Platby



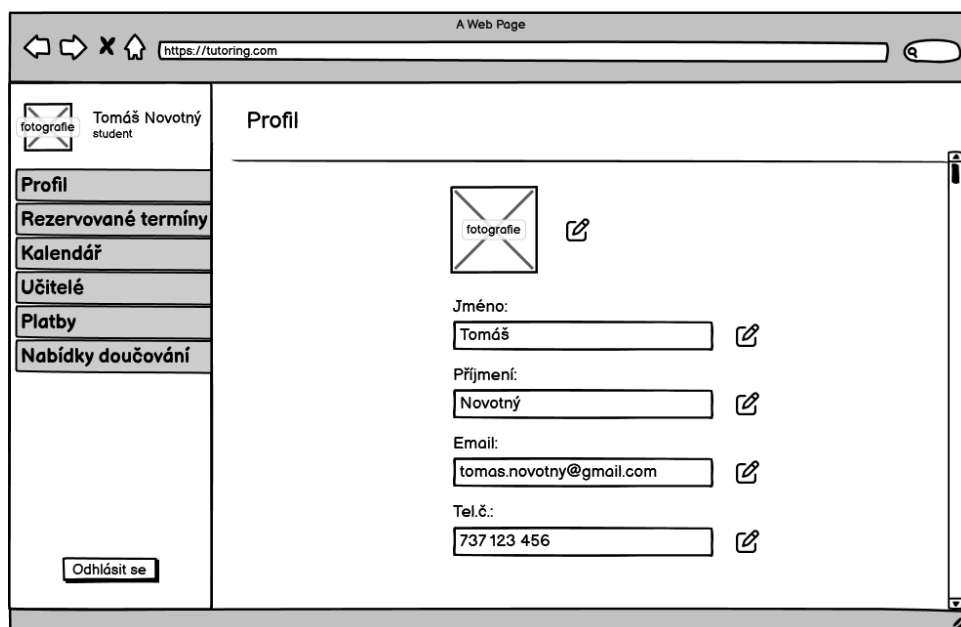
Obrázek B.12: Učitelé



Obrázek B.13: Studenti



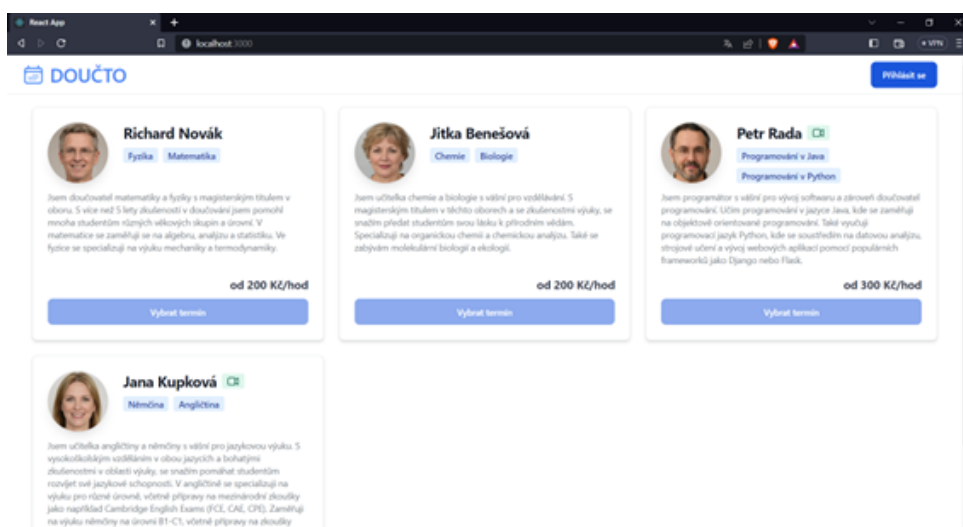
Obrázek B.14: Profil učitele



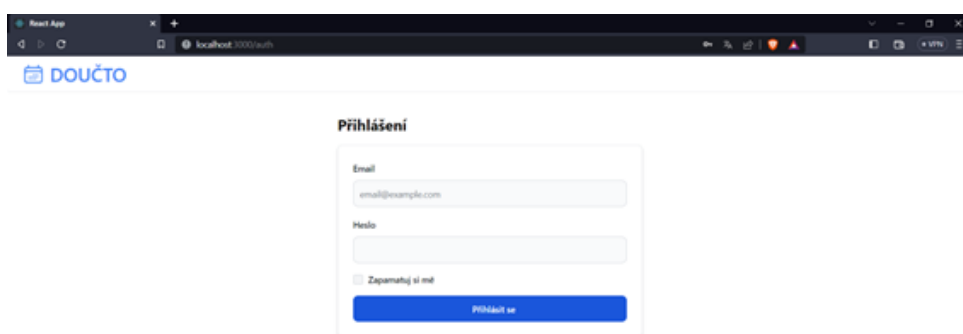
Obrázek B.15: Profil studenta

Příloha C

Ukázka výsledné aplikace

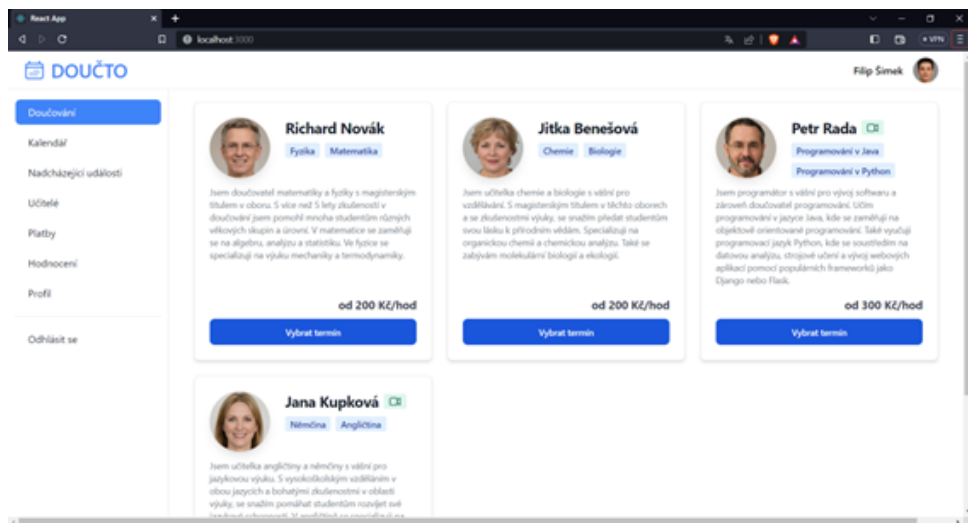


Obrázek C.1: Pohled pro nepřišlášeného uživatele

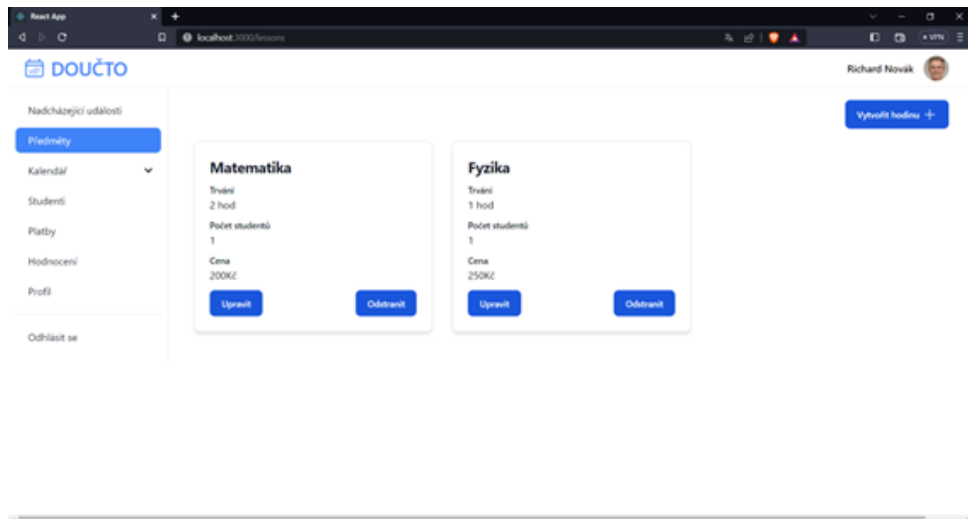


Obrázek C.2: Přihlašovací stránka

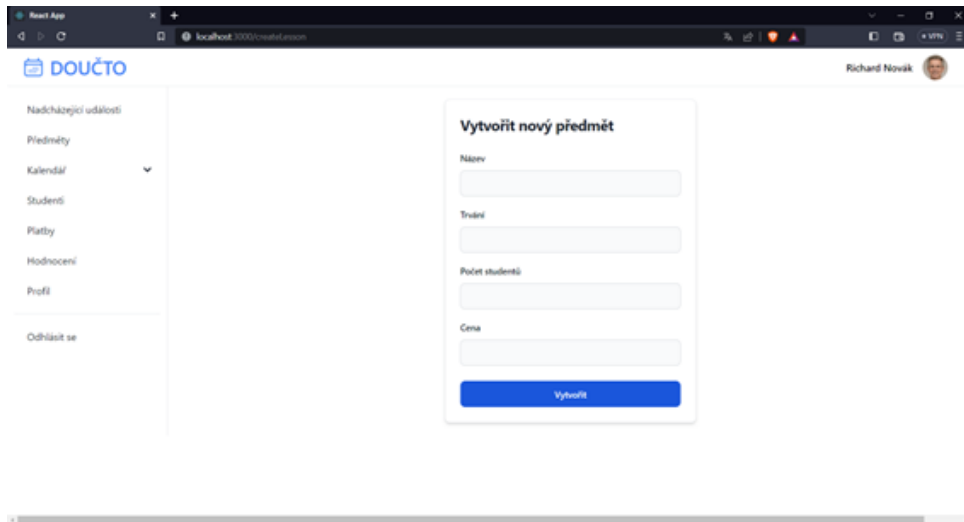
C. Ukázka výsledné aplikace



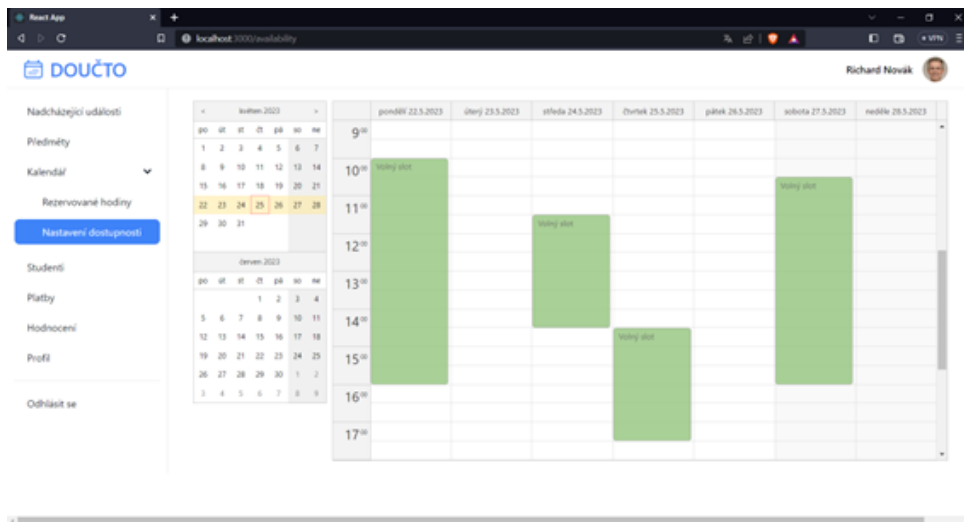
Obrázek C.3: Nabídky doučování



Obrázek C.4: Předměty

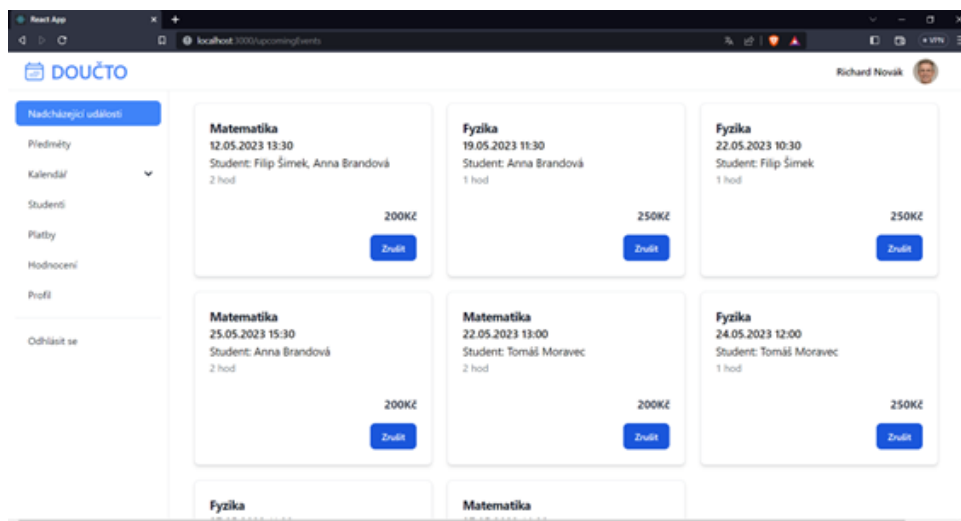


Obrázek C.5: Vytvoření předmětu

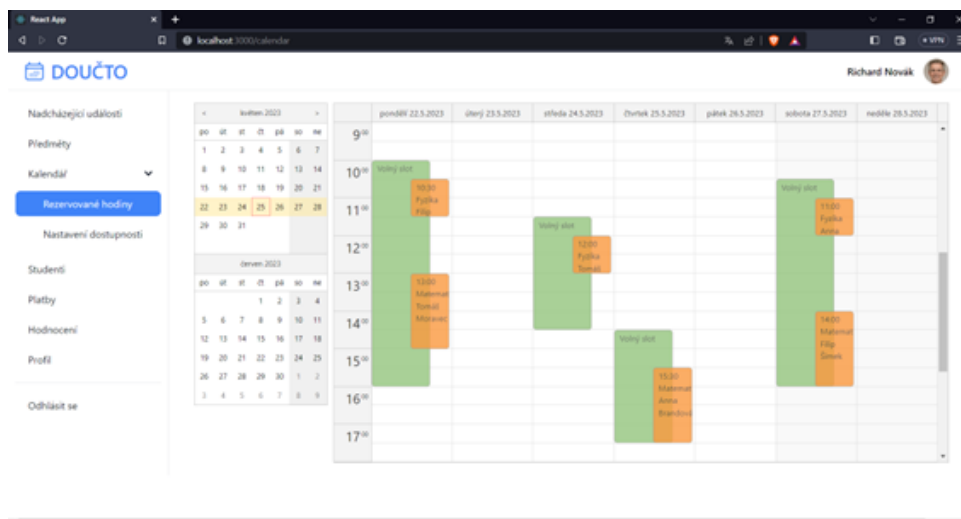


Obrázek C.6: Nastavení dostupnosti

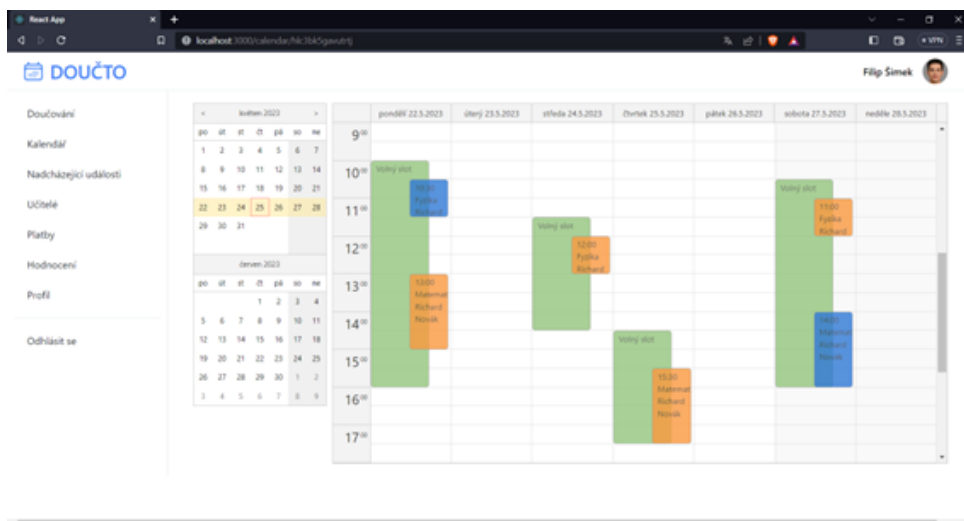
C. Ukázka výsledné aplikace



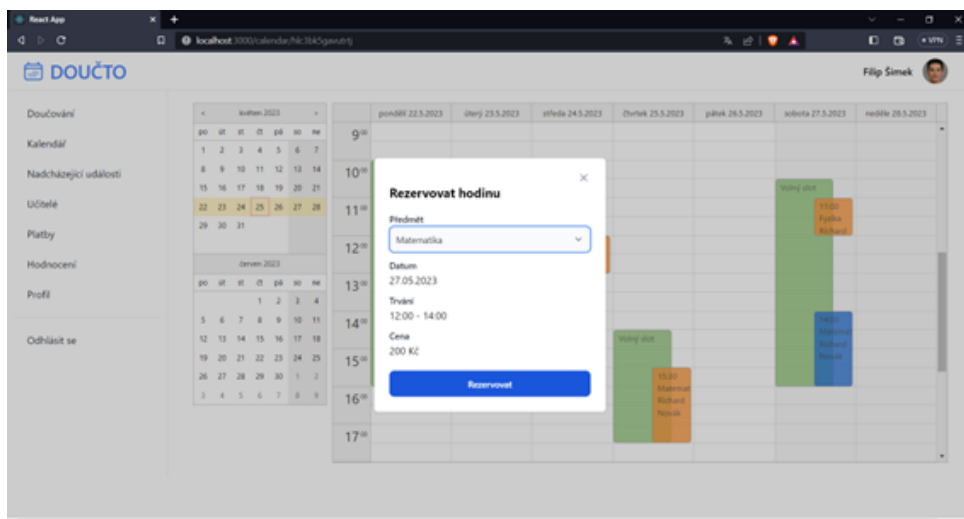
Obrázek C.7: Nadcházející události



Obrázek C.8: Rezervované události v kalendáři - pohled učitele

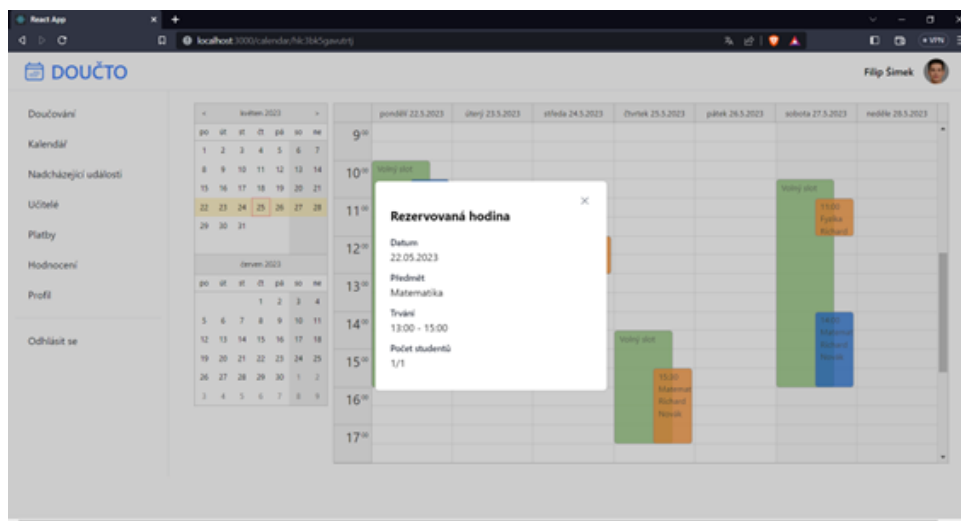


Obrázek C.9: Rezervované události v kalendáři učitele - pohled studenta

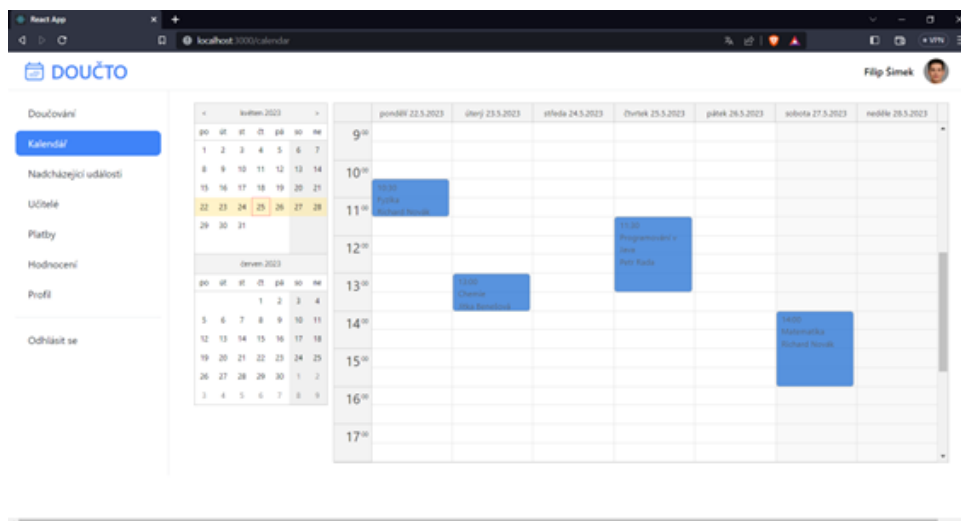


Obrázek C.10: Vybrání slotu pro rezervaci

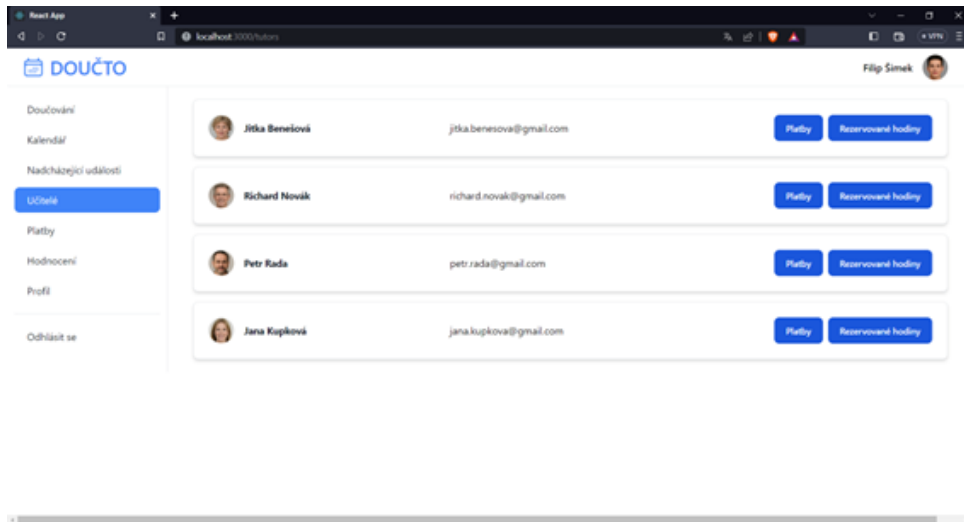
C. Ukázka výsledné aplikace



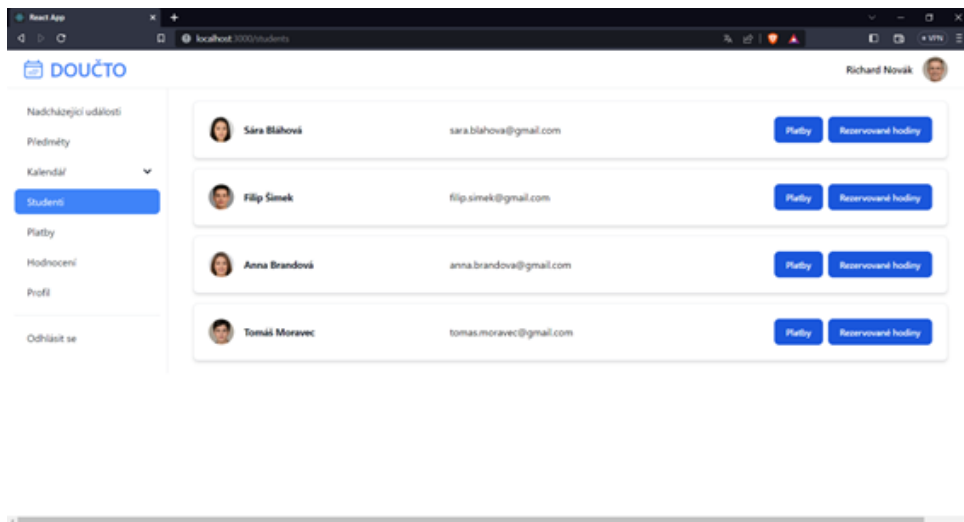
Obrázek C.11: Detail události



Obrázek C.12: Rezervované události v kalendáři - pohled studenta

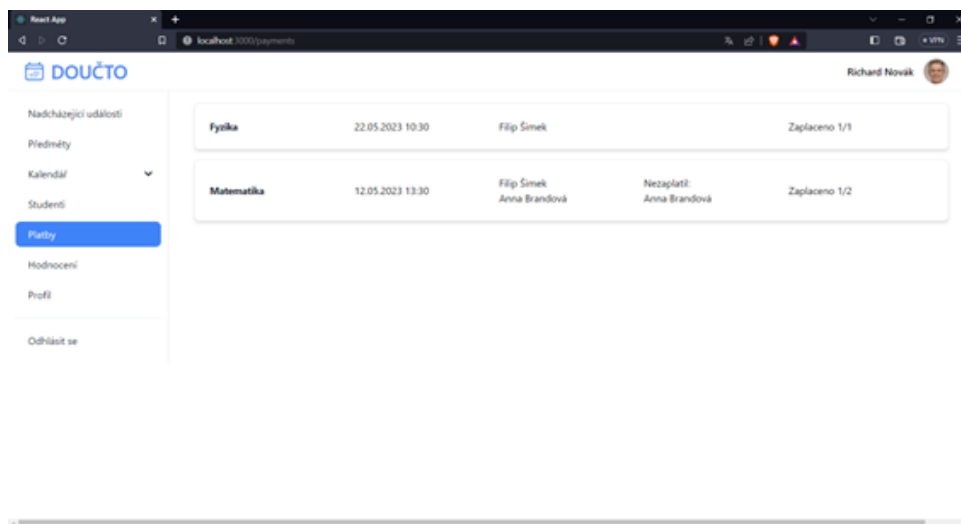


Obrázek C.13: Seznam učitelů

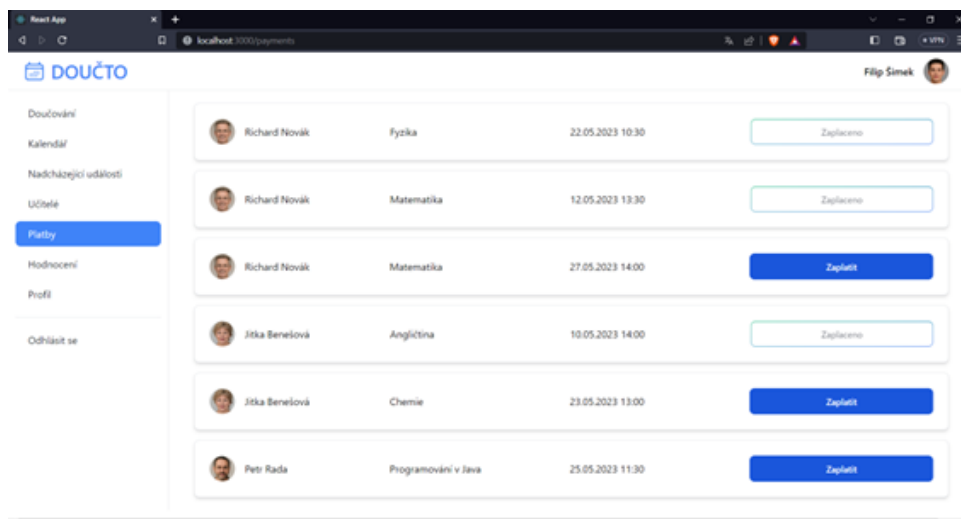


Obrázek C.14: Seznam studentů

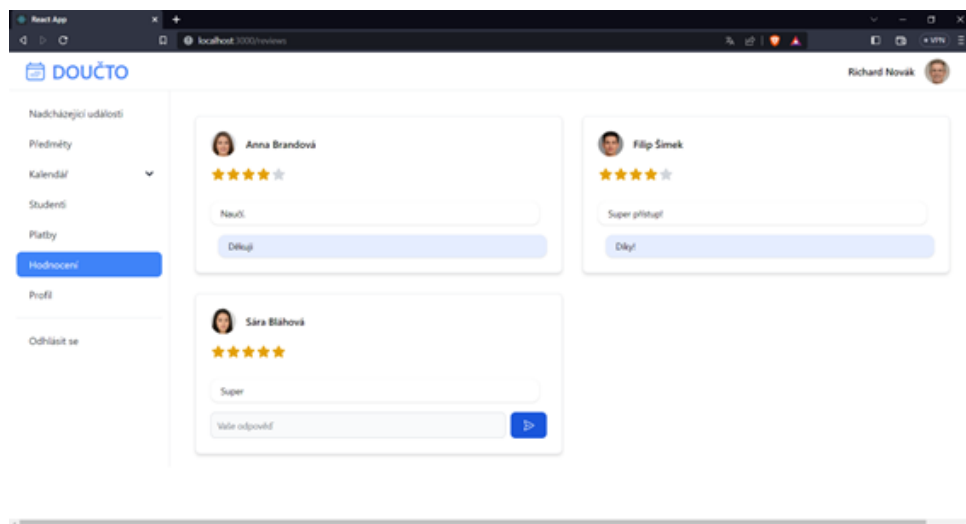
C. Ukázka výsledné aplikace



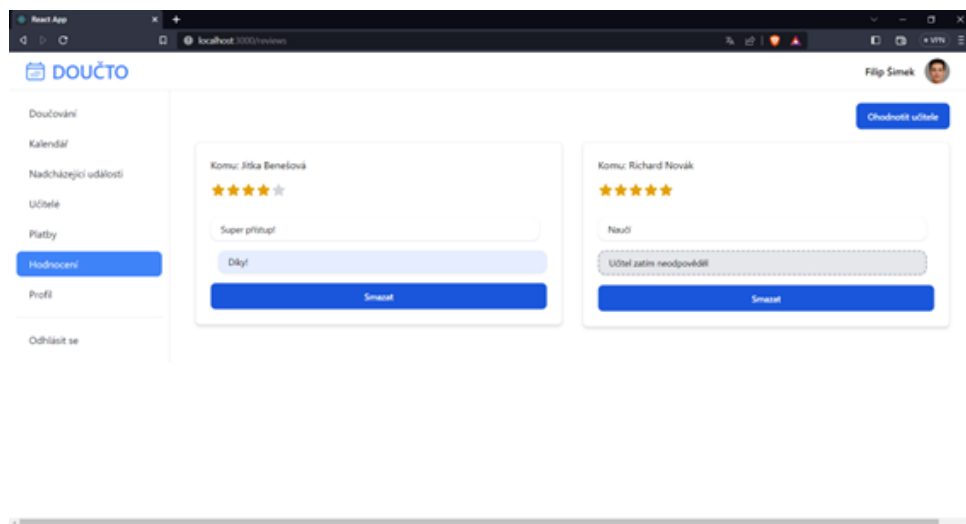
Obrázek C.15: Platby - pohled učitele



Obrázek C.16: Platby - pohled studenta

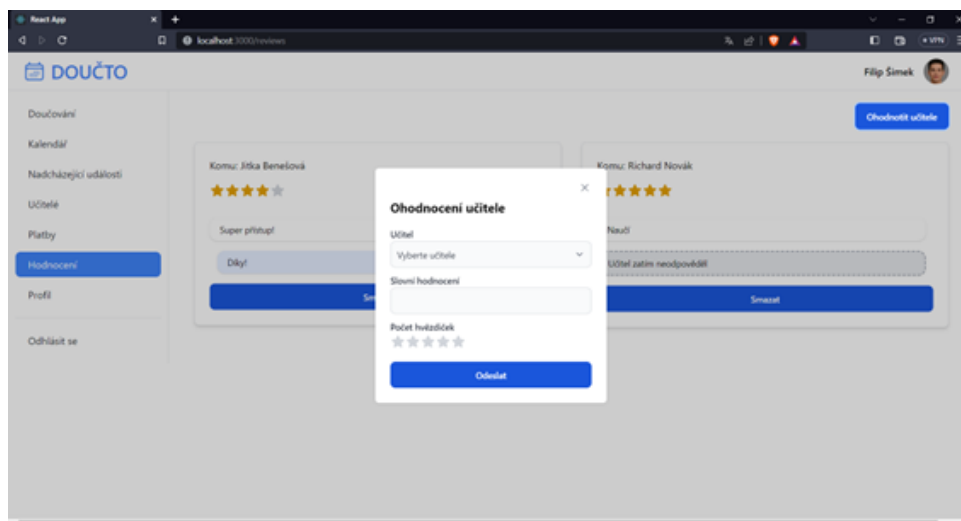


Obrázek C.17: Hodnocení - pohled učitele

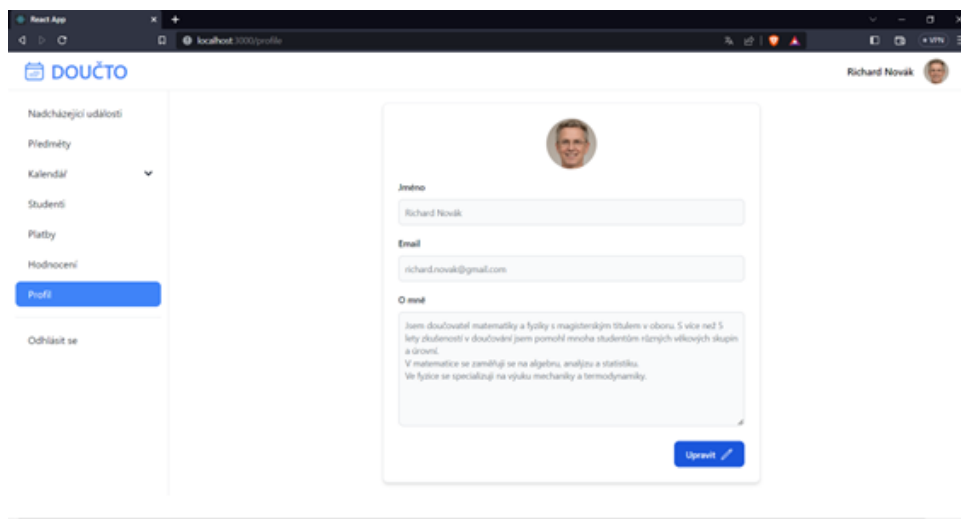


Obrázek C.18: Hodnocení - pohled studenta

C. Ukázka výsledné aplikace



Obrázek C.19: Ohodnocení učitele



Obrázek C.20: Profil učitele

Příloha D

Instalace a spuštění aplikace

Pro instalaci a spuštění aplikace postupujte dle instrukcí v souboru `README.md`



Obrázek D.1: Odkaz na GitLab repozitář se zdrojovým kódem aplikace