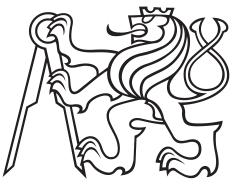
Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

Aplikace pro správu osobních financí

Kryštof Piorecký

Květen 2023

Vedoucí práce: Ing. Božena Mannová, Ph. D.

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Piorecký** Jméno: **Kryštof** Osobní číslo: **499067**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra kybernetiky**

Studijní program: **Otevřená informatika**

Specializace: **Základy umělé inteligence a počítačových věd**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Aplikace pro správu osobních financí

Název bakalářské práce anglicky:

Application for Managing Personal Finances

Pokyny pro vypracování:

Cílem je vytvořit aplikaci, která bude pomáhat jednotlivcům, nebo domácnostem udržovat přehled o jejich výdajích a příjmech. První částí je analýza existujících řešení, která poskytne přehled o funkcích, které konkurenční aplikace obsahují. Druhým cílem práce je navrhnout architekturu aplikace, zanalyzovat možné způsoby řešení a vybrat ty nejvhodnější
1. Seznamte se s problematikou správy osobních financí.
2. Proveďte analýzu vám dostupných konkrétních aplikací a existujících řešení, která poskytne přehled o funkcích, které tyto aplikace obsahují.
3. Na základě provedené analýzy navrhněte základní funkcionality aplikace.
4. Zvolte architekturu aplikace a vyberte nejvhodnější technologie pro tvorbu desktopové aplikace, mobilní aplikace, backendu a databáze.
5. Aplikaci implementujte a otestujte.
6. Zhodnoťte výsledky a navrhněte případné další funkcionality nebo jiná zlepšení.
7. Při řešení využívejte vhodných prostředků SE.

Seznam doporučené literatury:

[1] Roger S. Pressmann Bruce Maxim: Software Engineering: A Practitioner's Approach,
ISBN-10: 9780078022128
[2] https://financer.com/cz/blog/aplikace-spravce-financi/
[3] https://play.google.com/store/apps/details?id=ru.innim.my_finance&hl=cs&gl=US&pli=1

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Božena Mannová, Ph.D. kabinet výuky informatiky FEL

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

Datum zadání bakalářské práce: **14.02.2023** Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **22.09.2024**

Ing. Božena Mannová, Ph.D.	prof. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	podpis děkana(ky)

 © ČVUT v Praze, Design: ČVUT v Praze, VIC

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen 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 studenta

 © ČVUT v Praze, Design: ČVUT v Praze, VIC

Poděkování / Prohlášení

Rád bych poděkoval vedoucí Ing. Božena Mannová, Ph. D. za skvělou komunikaci a pomoc při zpracování této bakalářské práce.

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

V Praze, 24. května 2023

Abstrakt / Abstract

Cílem práce je zanalyzovat, navrhnout a implementovat aplikaci pro správu osobních financí. Aplikace bude dostupná ve formě progresivní webové aplikace. To umožní možnost nainstalovat aplikaci na mobilních systémech, které to podporují. Součástí práce je analýza existujících řešení, kde jsou porovnány existující aplikace se stejným účelem. Dále byla provedena analýza technologií a byly vybrány takové technologie, které zaručí plynulý běh a udržitelnost aplikace do budoucna. Následuje část implementace, kde jsou popsány funkcionality aplikace. Další částí práce je testování, kde je popsáno jak probíhá testování aplikace.

Klíčová slova: Osobní finance, Progresivní webová aplikace, Backend, MERN stack

The goal of this project is to analyze, design and implement an application for personal finance management. The application will be accessible in a form of a progressive web application. This will allow users to install the application on their devices. Part of the project is an analysis of existing solutions, which contains a comparison of existing applications with the same goal. Next, an analysis of technologies was done. Technologies that will guarantee a smooth running and sustainable application were selected. The implementation chapter describes individual functionalities of the application. The following chapter is about testing and it describes how the app was tested.

Keywords: Personal finance, Progressive web application, Backend, MERN stack

Title translation: Application for Managing Personal Finances

Obsah /

Kapitola 1
Úvod

Úkolem aplikace pro správu osobních financí je poskytnout uživateli přehledné shrnutí jeho finančních výdajů a příjmů. Přehled o výdajích organizovaných do kategorií může uživateli pomoci ušetřit finance do budoucna.

S narůstajícím množstvím služeb placených na měsíční bázi není těžké ztratit přehled o celkových měsíčních výdajích.

Spolubydlení je dalším místem, kde přijde správa financí vhod. Nákupy produktů pro společné užití je v takovém případě jeden z hlavních způsobů, jak jednoduše ušetřit finance. Ceny maličkostí se časem nasčítají a je dobré si o tom držet přehled.

Pro tuto práci jsem se rozhodl, protože sám používám větší množství služeb placených na měsíční bázi. Obecně si myslím, že je dobré vědět kolik a za co utrácím. Měsíční shrnutí v podobě grafů a výpisu výdajů rozděleného do kategorií by mohlo pomoci mně i všem ostatním, kteří rádi nakládají s financemi rozumně.

1.1 Cíl práce

Cílem je vytvořit aplikaci, která bude pomáhat jednotlivcům, nebo domácnostem udržovat přehled o jejich výdajích a příjmech. První částí je analýza existujících řešení, která poskytne přehled o funkcích, které konkurenční aplikace obsahují. Druhým cílem práce je navrhnout architekturu aplikace, zanalyzovat možné způsoby řešení a vybrat ty nejvhodnější.

Kapitola 2
Analýza

Tato kapitola slouží k analýze existujících řešení a analýze technologií. Provedena byla analýza existujících řešení, kde byly porovnány jejich funkcionality.

2.1 Analýza existujících řešení

V této části jsou představeny existující řešení aplikací pro správu osobních financí. Zahrnuty jsou jedny z nejznámějších a nejlépe hodnocených aplikací dostupné na českém trhu. Všechny aplikace byly testovány na mobilním telefonu s operačním systémem Android. Pokud aplikace nabízí instalovatelnou desktopovou verzi, byla testována na operačním systému Windows 10. Desktopové verze dostupné v prohlížeči byly testovány v internetovém prohlížeči Mozilla Firefox.

V každé aplikaci bylo otestováno přidání opakujícího se příjmu, jednorázové a opakující se platby a úprava kategorií pro přehlednost.

U každé aplikace je též uvedený počet stažení a hodnocení, pro představu, jak moc je aplikace používaná. Údaje jsou aktuální k prosinci 2022.

2.1.1 Goodbudget

Goodbudget je aplikace pro správu osobních financí. Při prvním spuštění je uživatel dotázán pro vyplnění opakujících se příjmů a výdajů. Poté následuje registrace účtu. Aplikace je dostupná v podobě webové stránky a mobilní aplikace. Aplikace je zcela zdarma, avšak obsahuje i placený plán pro náročnější uživatele. Aplikace umožňuje rozdělovat výdaje do kategorií (Envelopes). Tyto kategorie může uživatel přidávat a upravovat, avšak bez placeného plánu má uživatel k dispozici prostor pro pouze 10 kategorií. Hlavní přehled uživateli poskytuje náhled na výdaje rozdělené do jednotlivých kategorií v aktuálním měsíci.

- Klíčové vlastnosti:
  - Synchronizace přes účet.
  - Registrace účtu pouze přes E-mail.
  - Rozdělení výdajů do kategorií.
  - K výdajům jde přidat velké množství podrobností, komu bylo placeno, datum a nepovinný popisek.
- Vlastnosti uživatelského rozhraní:
  - Aplikace zobrazuje přehled prostřednictvím grafů.
  - Možnost změnit barvy aplikace na tmavý nebo světlý režim.
  - Kalkulačka při zadávání hodnoty výdaje.
- Popularita aplikace:
  - Počet stažení 1M+
  - Google play: 4.2/5 z 18K recenzí
  - App store: 4.7/5 12K recenzí

2.1.2 Fudget

Fudget je aplikace pro správu osobních financí s velmi jednoduchým uživatelským prostředím. Aplikace obsahuje průvodce, který nového uživatele seznámí s aplikací a její obsluhou. Aplikace je zdarma, avšak obsahuje reklamy v podobě banneru ve spodní části obrazovky. Jednorázovou platbou je možné reklamy odstranit a přidat i další výhody, jako je např. volba barevného schématu a synchronizace přes Dropbox. Aplikace umožňuje náhled na příjmy a výdaje pomocí jednoho koláčového grafu. Aplikaci je možné stáhnout na Android, iOS a Windows.

- Klíčové vlastnosti:
  - Všechna data jsou ukládána lokálně.
  - Rozdělení výdajů i příjmů do kategorií.
  - Ke každé kategorii je třeba nastavit rozpočet.
  - K výdaji je možné zadat pouze poznámku a částku.
- Vlastnosti uživatelského rozhraní:
  - Neplacená verze obsahuje reklamy a pouze jedno barevné schéma v oranžové barvě.
  - Aplikace zobrazuje jeden koláčový graf, který ukazuje procentuální rozdělení výdajů.
- Popularita aplikace:
  - Počet stažení 100K+
  - Google play: 4.7/5 4K recenzí
  - App store: 4.7/5 4K recenzí

2.1.3 Spendee

Spendee je aplikace sloužící ke správě osobních financí. Aplikace po prvním spuštění uvítá uživatele registrací, kde nabízí řadu možností pro přihlášení pomocí SSO. Aplikace za příplatek také podporuje integraci s bankami a crypto peněženkami. Výdaje i příjmy jsou organizovány do kategorií, které si může uživatel upravovat a přidávat. Aplikace je dostupná pro mobilní telefony s operačními systémy Android a iOS. Pro desktop je aplikace dostupná prostřednictvím webových stránek.

- Klíčové vlastnosti:
  - Synchronizace pomocí účtu.
  - Registrace je možná pomocí Google účtu, Facebook účtu, Apple účtu nebo pomocí E-mailové adresy.
  - Graf s výdaji a seznam posledních výdajů hned na hlavní obrazovce.
  - Přidávání nových a úprava existujících kategorií. U každé kategorie jde nastavit jméno, barva a ikonka pro přehlednost. Kategorie jsou rozdělené na kategorie pro výdaje a kategorie pro příjmy.
  - Přítomné jsou také štítky, které fungují podobně jako kategorie, pro větší přehlednost.
- Vlastnosti uživatelského rozhraní:
  - Velmi přehledné a moderní rozhraní.
  - Možnost výběru mezi světlým a tmavým režimem.

- Popularita aplikace:
  - Počet stažení 1M+
  - Google play: Hodnocení 4.5/5 z 39 tisíc recenzí
  - App store: Hodnocení 4.6/5 z 4500 recenzí

2.1.4 Buxfer

Buxfer je aplikace pro správu osobních financí a sledování více bankovních účtů v jedné aplikaci. Aplikaci je možné nainstalovat systémy Android, iOS. Na desktopu je možné využít její webovou verzi.

- Klíčové vlastnosti:
  - Synchronizace dat přes účet.
  - Registrace přes E-mail, Google a Facebook účty.
  - K výdajům je možné přidat velké množství podrobností. Mezi ně patří: množství, popis, datum a tagy.
- Vlastnosti uživatelského rozhraní:
  - Aplikace nabízí pouze světlý režim.
  - Aplikace je obohacena velkým množstvím funkcionalit navíc. Mezi takové funkcionality patří: připomínky, investice a předpověď.
- Popularita aplikace:
  - Počet stažení 10K+
  - Google play: Hodnocení 4.2/5 z 676 recenzí
  - App store: Hodnocení 4.3/5 z 119 recenzí

2.1.5 Porovnání

V této části jsou jednotlivé existující aplikace porovnané v různých kategoriích.

Funkcionalita	Goodbudget	Fudget	Spendee	Buxfer
Kategorie	10 kategorií odlišené pouze jménem	Kategorie odlišené jménem	Kategorie odlišené jménem, barvou a ikonkou	Kategorie odlišené pouze jménem
Grafy	Koláčový graf pro výdaje. Sloupcový graf pro měsíční výdaje a příjmy	Každá kategorie obsahuje vlastní koláčový graf obsahující jednotlivé výdaje.	Sloupcový graf výdajů, nastavitelné časové období grafů, koláčový graf rozdělený na kategorie výdajů.	Pouze placená verze.
Změna měny	Neobsahuje měny.	Možnost výběru ze 14 předpřipravených symbolů, nebo možnost zadat vlastní symbol pomocí textového pole.	Změna měny v nastavení na velké množství předpřipravených měn. Příjmy i výdaje mohou být v různých měnách.	Neobsahuje měny.
Reklamy	Neobsahuje.	Banner reklama ve spodku aplikace a banner reklama pro zakoupení placené verze.	Občasná reklama pro zakoupení placeného plánu.	Části aplikace zamknuté pouze pro placenou verzi s proklikem na koupi.
Synchronizace	Přes online účet.	Placená verze obsahuje synchronizaci přes Dropbox.	Přes online účet.	Přes online účet.
Heslo aplikace	Neobsahuje.	Číselné heslo, nutné zadat při každém vstupu do aplikace.	Čtyřmístný číselný kód a biometriku. Zadání hesla při každém vstupu do aplikace.	Čtyřmístný číselný kód a biometrika. Zadání hesla při každém vstupu do aplikace.
Platformy	Aplikace pro iOS a Android, webová aplikace.	Aplikace pro Android, iOS a Windows.	Aplikace pro iOS a Android, webová aplikace.	Aplikace pro iOS a Android, webová aplikace.
Barevné schéma	Tmavý/světlý režim.	V neplacené verzi pouze oranžový režim.	Tmavý/světlý režim, nebo automatické nastavení podle systému.	Pouze světlý režim.
Velikost	iOS - 27MB, Android - 7MB	iOS - 22MB, Android - 7MB	iOS - 70MB, Android - 44MB	iOS - 20MB, Android - 45MB

Tabulka 2.1. Tabulka porovnání funkcionalit existujících řešení.

2.2 Závěr

Byly porovnány nejoblíbenější a nejpoužívanější aplikace pro správu osobních financí. V porovnání jsou analyzovány funkcionality jednotlivých testovaných aplikací a posuzována jejich užitečnost. Užitečné funkcionality jsou takové, které značně ulehčí uživateli používání klíčových funkcionalit aplikace a celkové navigovaní aplikací. Tyto funkcionality mají za účel uživateli ušetřit čas a zpříjemnit používání celé aplikace.

- Užitečné funkcionality:
  - Kalkulačka při zadávání financí.
    - V některých případech uživatel zakoupil stejný produkt vícekrát, nebo byl nákup rozdělen na více částí. V takových případech uživatel nebude muset jednotlivé položky sčítat v hlavě.
  - Přidání výdaje nebo příjmu v jakékoliv části aplikace.
    - Hlavním důvodem aplikace pro správu osobních financí na mobilním telefonu je jednoduchost přístupu a rychlost zadání právě obdržené nebo vydané částky. Je tedy důležité, aby se uživatel nemusel navigovat do speciální záložky, ale měl toto tlačítko vždy po ruce.
  - Změna barvy prostředí.
    - Upravení barvy prostředí může uživateli pomoct s přehledností.
  - Změna měny.
    - Aplikace by měla poskytovat uživateli změnit měnu. Pro jednoduchost je dobré uživateli nabídnout na výběr již předpřipravené měny a pro krajní případy je dobré poskytnout uživateli také možnost zadat ručně vlastní symbol.

Kapitola 3
Návrh řešení

3.1 Analýza požadavků

Softwarové požadavky popisují potřeby a omezení výsledné aplikace. Dělí se na funkční a nefunkční požadavky. Funkční požadavky představují funkcionality, které by měl software vykonávat. Nefunkční požadavky popisují omezení. [1]

3.1.1 Funkční požadavky

	Požadavek	Popis
FP-1	Registrace	Možnost vytvořit uživatelský účet.
FP-2	Přihlášení	Možnost přihlášení k existujícímu uživatelskému účtu.
FP-3	Zapomenuté heslo	Obnova hesla přes E-mail.
FP-4	Odhlášení	Možnost odhlásit se z přihlášeného uživatelského účtu.
FP-5	Odstranění účtu	Možnost odstranění uživatelského účtu.
FP-6	Úprava preferencí	Možnost úpravy uživatelských údajů a nastavení prostředí.
FP-7	Jazyk	Možnost změnit jazyk uživatelského prostředí.
FP-8	Měna	Možnost změnit měnu.
FP-9	Grafy	Aplikace umožní zobrazit grafy pro přehled příjmů a výdajů.
FP-10	Vyhledání uživatele	Možnost vyhledat registrované uživatele.
FP-11	Založení nástěnky	Možnost vytvořit novou nástěnku.
FP-12	Odstranění nástěnky	Možnost smazat existující nástěnku.
FP-13	Úprava nástěnky	Možnost změny informací o nástěnce.
FP-14	Přizvání uživatele k nástěnce	Možnost přijmout pozvánku k nástěnce. Připojí uživatele k dané nástěnce.
FP-15	Přijmutí pozvánky k nástěnce	Možnost odmítnout pozvánku k nástěnce. Odstraní danou pozvánku.
FP-16	Odmítnutí pozvánky k nástěnce	Možnost pozvat další uživatele k existující nástěnce.
FP-17	Opuštění nástěnky	Možnost opustit nástěnku, které je uživatel součástí.
FP-18	Seznam nástěnek	Možnost zobrazit seznam všech nástěnek, ke kterým je uživatel připojen.
FP-19	Výběr nástěnky	Uživatel vybere nástěnku, ve které si přeje provádět další akce. Systém uživatele přepne do této nástěnky.

3.1.2 Funkční požadavky 2

	Požadavek	Popis
FP-20	Přidání kategorie	Možnost vytvořit novou kategorii výdajů a příjmů.
FP-21	Seznam kategorií	Možnost zobrazit seznam všech kategorií.
FP-22	Úprava kategorií	Možnost upravit podrobnosti existující kategorie.
FP-23	Odstranění kategorií	Možnost odstranit existující kategorie.
FP-24	Přidání jednorázového příjmu	Možnost přidat jednorázový příjem.
FP-25	Přidání opakujícího příjmu	Možnost přidat opakující příjem.
FP-26	Přidání jednorázového výdaje	Možnost přidat jednorázový výdaj.
FP-27	Přidání opakujícího výdaje	Možnost přidat opakující výdaj.
FP-28	Ukončení opakujícího příjmu	Možnost ukončit opakující příjem.
FP-29	Ukončení opakujícího výdaje	Možnost ukončit opakující výdaj.
FP-30	Odstranění příjmu	Možnost odstranit příjmy obou typů.
FP-31	Odstranění výdaje	Možnost odstranit výdaje obou typů.
FP-32	Úprava příjmu	Možnost upravit příjmy obou typů.
FP-33	Úprava výdaje	Možnost upravit výdaje obou typů.
FP-34	Seznam příjmů	Možnost zobrazit seznam všech příjmů.
FP-35	Rychlé zadání výdaje a příjmu	Aplikace umožní rychlé přidání výdajů, nebo příjmů.
FP-36	Přidání předplatných	Možnost přidat nové předplatné.
FP-37	Seznam předplatných	Možnost zobrazit seznam všech předplatných.
FP-38	Úprava předplatných	Možnost upravit podrobnosti existujícího předplatného.
FP-39	Odstranění předplatných	Možnost odstranit existující předplatné.

Tabulka 3.1. Tabulka funkčních požadavků.

3.1.3 Nefunkční požadavky

	Požadavek	Popis
NFP-1	Bezpečnost	Šifrovaná hesla a ochrana proti XSS útokům.
NFP-2	Přehlednost	Aplikace bude přehledná a jednoduchá k použití.
NFP-3	Univerzalita	Aplikace bude dostupná na různých operačních systémech a v různých prohlížečích.

Tabulka 3.2. Tabulka nefunkčních požadavků.

3.2 Návrh

Hlavním cílem aplikace je, aby byla přístupná na mobilních telefonech a uživatelé jí mohli pohodlně používat pro zadání výdajů ihned po zaplacení. Druhotným cílem je, aby byla aplikace dostupná i na desktopu pro přehlednější zobrazení různých dat v

tabulkách a grafech. Z důvodu sdílení dat mezi mobilním telefonem a desktopem bude potřeba synchronizovat data.

Aplikace má za účel pomáhat jednotlivcům a domácnostem udržet přehled o jejich výdajích a příjmech.

U porovnání technologií je kladen větší důraz na znovupoužitelnost a udržitelnost kódu před výkonem. Aplikace neobsahuje složité výpočty, ale slouží pro zobrazení a úpravu dat na vzdáleném serveru.

Je třeba vhodně vybrat technologie pro tvorbu desktopové aplikace, mobilní aplikace, backendu a databáze. Některé kombinace technologií jsou často používané dohromady, to znamená, že existuje velké množství návodů a bude jednodušší řešit problémy při implementaci. Použití některých technologií také umožní využití stejného kódu pro více částí projektu, to má za následek jednodušší správu aplikace do budoucna.

3.2.1 Webová aplikace

Webová aplikace zajistí možnost používání aplikace nezávisle na operačním systému. Webovou aplikaci je možné vytvořit pouze pomocí HTML, CSS a JS bez potřeby frameworku. Výhodou použití frameworku pro tvorbu webové aplikace je předdefinovaná struktura projektu, kterou používá velké množství vývojářů. Pokud tedy na projektu pracuje, nebo bude v budoucnu pracovat větší množství vývojářů, je lehčí udržet strukturu kódu. Další výhodou je bezpečnost, frameworky obsahují ochranu proti XSS útokům.

- React
  - React je knihovna pro vývoj uživatelských rozhraní. Zaměřuje se na efektivní správu stavů a změny v uživatelském rozhraní aplikace. To je dosaženo pomocí virtuálního Document Object Modelu. [2]
  - React.js je podle Stack Overflow Developer Survey 2022 nejpoužívanější frontend technologií. [3]
  - React používa JSX pro psaní struktury komponent. JSX je syntaktické rozšíření pro JavaScript, které umožňuje psát kód podobný HTML v JavaScriptu.
- Angular
  - Angular je druhá nejpopulárnější JavaScript technologie pro vytváření uživatelských rozhraní. Angular poskytuje širokou škálu nástrojů a funkcí pro vývoj webových aplikací, včetně podpory pro správu stavu aplikace, routing, formuláře a mnoho dalšího. [4]
- Vue.js
  - Vue.js je progresivní frontendový JavaScriptový framework pro tvorbu uživatelských rozhraní. Vue.js se skládá z komponent, což jsou opakovatelné kusy kódu, které lze snadno kombinovat k vytvoření složitých uživatelských rozhraní. Obsahuje funkce, jako je routing a správa stavu. [5]

3.2.2 Mobilní aplikace

Řada aplikací funguje pouze jako webové aplikace bez dedikované mobilní aplikace. Ovšem dedikovaná aplikace poskytuje uživateli mnohem snadnější přístup. Místo zadávání URL adresy do prohlížeče, stačí ke spouštění z domovské obrazovky jeden klik na ikonu.

- Xamarin
  - Aplikační framework Xamarin umožňuje psaní jednoho kódu pro více platforem. Xamarin je ideální, pokud je primárním cílem tvorba desktopové aplikace a sekundárním cílem i instalovatelná aplikace na mobilní telefon. Framework Xamarin využívá programovací jazyk C#. [6]
- Flutter
  - Flutter je open source framework pro vývoj mobilních aplikací pro Android, iOS, desktop i web. Umožňuje vytvářet aplikace s přehledným a moderním designem pomocí jazyka Dart. [7]
- PWA
  - PWA (Progressive Web App) se v posledních letech stává stále populárnější díky schopnosti poskytovat uživatelům zkušenost podobnou nativní aplikaci. PWA se instaluje na zařízení uživatele a fungují offline, stejně jako nativní aplikace. PWA také umožňuje vývojářům vytvářet aplikace se sdíleným kódem pro více platforem. To vše přispívá k jejich rostoucí popularitě mezi vývojáři a uživateli. [8]

3.2.3 Backend

Úlohou serverové části aplikace je poskytnout uživateli synchronizaci jeho dat. Serverová část komunikuje s databází a na dotaz zpracovává a předává data frontendové části.

- PHP
  - PHP je server-side skriptovací jazyk. PHP kód je na serveru spuštěn po uživatelském dotazu na server. PHP nativně nepodporuje WebSockety ani UDP. Případné budoucí push notifikace by potom musely být řešeny zvlášť jinou technologií. Mezi nejčastější PHP frameworky patří: Laravel, Symfony, CodeIgniter, Zend Framework a CakePHP. [9]
- Java
  - Java je populárním jazykem a má širokou škálu nástrojů pro vývojáře, včetně integrovaných vývojových prostředí (IDE), knihoven a frameworků. Mezi nejčastější Java frameworky patří: Spring, Hibernate, Struts, Play a JSF. [10]
- Node.js
  - Node.js je asynchronní událostmi řízený modul JavaScriptu určený k tvorbě škálovatelných síťových aplikací. JavaScriptu bylo nejdříve pouze zinteraktivnit klientskou část webových aplikací, ale s každoroční narůstající popularitou tohoto jazyka se v roce 2009 poprvé objevila serverová alternativa v podobě Node.js. V posledních letech je Node.js nejpoužívanější technologií pro vývoj webových aplikací. Nejčastěji používaným frameworkem používaným společně s Node.js je Express. [11]

3.2.4 Databáze

Databáze slouží k ukládání a organizaci dat. Backend aplikace získává data z databáze pomocí dotazů.

- MySQL
  - MySQL je relační databáze. Data jsou ukládána v tabulkách s předem definovanou strukturou. MySQL se často používá pro aplikace, které potřebují rychlý přístup k datům a podporu transakcí. [12]
- MongoDB
  - MongoDB je NoSQL databáze. Data jsou ukládána ve formě dokumentů, které mohou mít libovolnou strukturu. MongoDB je často používána, pokud se pracuje s velkým objemem dat. [13]

3.3 Závěr

Byly porovnány jednotlivé technologie. Kritérii bylo, jak dobře fungují s ostatními technologiemi, jak často jsou používány a osobní zkušenosti. Pro jednotlivé části aplikace budou využity následující technologie.

- Webová aplikace
  - Frontend aplikace bude využívat React, protože je to velmi rozšířený framework.
- Mobilní aplikace
  - Pro mobilní aplikaci bude použito PWA, která umožní použití stejného kódu pro mobilní a webovou aplikaci.
- Backend
  - Serverová část aplikace bude tvořena v Node.js společně s frameworkem Express.
- Databáze
  - Databáze bude použita MongoDB kvůli jejímu častému a snadnému použití s Node.js.

Celý tento stack se nazývá MERN a je jedním z nejčastěji používaným stackem pro vývoj webových aplikaci.

Kapitola 4
Implementace

V kapitole implementace se nachází popis tvorby aplikace. Popsaný je zde proces grafického návrhu, tvorby frontendové a backendové částí aplikace.

4.1 Grafický návrh

Aplikace je určena pro domácnosti a jednotlivce, to znamená, že je určena i na děti a seniory a je tedy velmi důležité navrhnout design aplikace tak, aby byl přehledný a snadno ovladatelný. Aplikace by tedy měla působit mile a hravě, důležité je tedy vybrat takovou barevnou paletu, aby byla velmi barevná. Obzvláště v sekci grafů je velká příležitost použít pěkné barvy.

4.1.1 Responzivita

Aplikace je vyvíjena s myšlenkou mobile-first. To znamená, že největší důraz je kladen na použitelnost na telefonech. Je důležité aby byl uživatel schopný vykonat všechny úkony v aplikaci pouze z telefonu, i když zrovna nebude mít přístup k počítači. Na desktopu je poté možné např. zobrazit více dat najednou, díky většímu prostoru na obrazovce.

U webových aplikacích se používají break-pointy pro optimalizaci designu pro více různých rozlišení. Aplikace je určena primárně k používaní ve vertikálním režimu na telefonech a v horizontálním režimu na počítači. Aplikace je nastavená tak, aby se škálovala do virtuálního rozlišení. Díky tomu se např. rozlišení telefonu iPhone 14 Pro Max s fyzickým rozlišením 1290x2796, jeví jako 430x932. To umožní rozlišit zařízení podle šířky obrazovky.

V aplikaci jsou pro responzivní design použity tyto break-pointy.

Zařízení	Rozmezí
Telefon	width 768px
Tablet	768px $\leq width 1024px$
Desktop	1024px $\leq width$

Tabulka 4.1. Tabulka break-pointů.

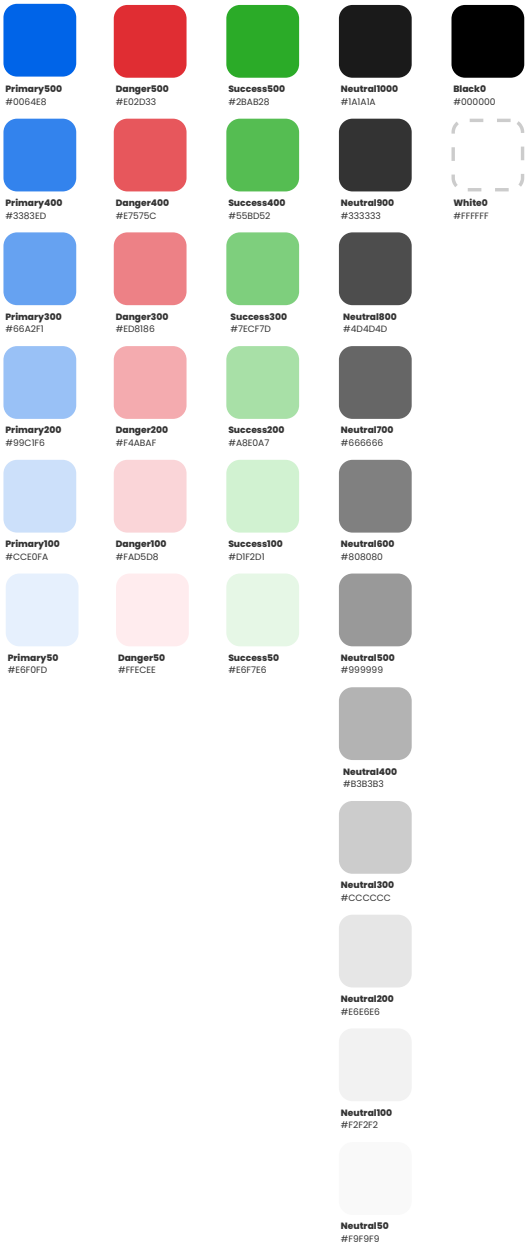
4.1.2 Výběr barevné palety

V aplikaci je potřeba jedna primární barva, barva pro nebezpečné akce a chybové zprávy, barva zprávy pro úspěšnou akci a skupinu barev, ze kterých si bude uživatel moci volit, např. při výběru barvy kategorie. Paleta bude také obsahovat škálu pro neutrální barvu, kde byly zvoleny odstíny šedé.

- Výběr primární barvy

  Primární barva byla zvolena modrá s ohledem na přehledné rozlišení barvy chybových zpráv a zpráv pro úspěšné akce, které je důležité zvolit jako červenou a zelenou. Taková paleta se nazývá triadická, barvy jsou rovnoměrně rozprostřené po barevném spektru. Vizualizace: https://www.canva.com/colors/color-wheel/
- Výběr odstínů barev

  V aplikaci se barvy vyskytují na bílém pozadí jako text, ale také na pozadí prvků (např. tlačítko), kde se vyskytuje bílý text na pozadí dané barvy. Je důležité vybrat takové odstíny, aby byl dodržen dostatečný kontrast. Všechny barvy v plné sytosti splňují standart WCAG úrovně AA pro text a úrovně AAA pro nadpisy. Pro výběr barev a jejich otestování byl použit designerský nástroj Color review: https://color.review/



Obrázek 4.1. Barevná paleta aplikace.

- Uživatelské barvy

  Tyto barvy budou uživateli k dispozici pro barevné odlišení kategorií a opakujících se plateb. Bylo vybráno 10 barev rovnoměrně rozprostřených po barevném spektru.

Obrázek 4.2. Uživatelské barvy.

4.1.3 Ikona

Ikonou aplikace je kreslená postava držící bankovku a symbolizující finanční zaměření aplikace. Ikona využívá primární barvu aplikace. Návrh na ikonu byl prvně vytvořen na papíru a následně zpracován do vektorové ikony.

Obrázek 4.3. Ikona aplikace.

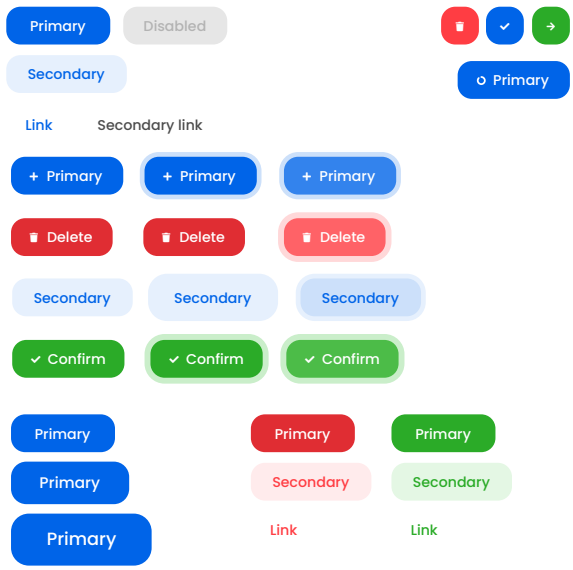
4.1.4 Návrh prostředí

Design language aplikace je primárně plochý. Stíny budou použity pouze v krajních situacích a stín bude velmi jemný. Mezi takové krajní situace patří např. vyskakovací okna, která překrývají jiné prvky uživatelského prostředí. Stín je zde třeba, jelikož bílý popup na bílém pozadí splývá a pokud by měl stejný okraj jako prvky, které překrývá, jednotlivé prvky by spolu také splývaly.

Aplikace bude primárně využívána na mobilních zařízeních. Ta jsou ovládaná dotykem a je tedy důležité všechny prvky uživatelského prostředí, které slouží ke kliknutí udělat dostatečně velké.

- Typografie
  - Pro celé prostředí aplikace byl zvolen font Poppins, který je dostupný na google fonts.
  - Tloušťky textu
    - 300 - Malé popisky.
    - 400 - Běžný text.
    - 500 - Aktivní prvky a zvýrazněné texty.
    - 700 - Zvýraznění v grafech.

- Tlačítka
  - Aktivní tlačítko a najetí kurzorem se projeví bledým obrysem tlačítka.
  - Tlačítko může obsahovat ikonu, ta se zobrazí nalevo od textu.
  - Tlačítko může obsahovat pouze ikonu bez textu, v tom případě bude mít poměr stran 1:1.
  - Tlačítko bylo navrhnuto do 3 velikostí a liší se primárně ve výšce:
    - Small - 32px
    - Medium - 36px
    - Large - 48px
  - Barva tlačítka má 3 možné barvy: primary, danger a success.
  - Pro oddělení důležitosti tlačítka jsou navržené 3 režimy.
    - Primary - Hlavní akce, tlačítko má syté pozadí a bílý text.
    - Secondary - Vedlejší akce, tlačítko má bledou barvu pozadí a sytou barvu textu.
    - Link - Akce s nejmenší prioritou, tlačítko má průhledné pozadí a sytou barvu textu.

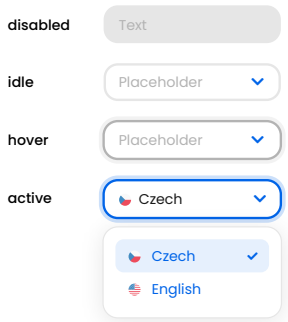


Obrázek 4.4. Varianty tlačítek.

- Textové pole
  - Aktivní textové pole a najetí kurzorem se projeví dvojím obrysem prvku.
  - Velikost textového pole je na výšku stejná jako tlačítka ve velikosti Large, stejně jako všechny ostatní formulářové prvky.

Obrázek 4.5. Varianty textových polí.

- Výběrové pole
  - Aktivní výběrové pole a najetí kurzorem se projeví dvojím obrysem prvku.
  - Zde u seznamu možností bylo potřeba mírně vybočit z obecného směru designu a použít jemný stín pro oddělení možností od prvků, které budou případně překrývat.
  - Kliknutím na prvek se otevře seznam možností.
  - Výběr se potvrdí kliknutím na jednu z možností.

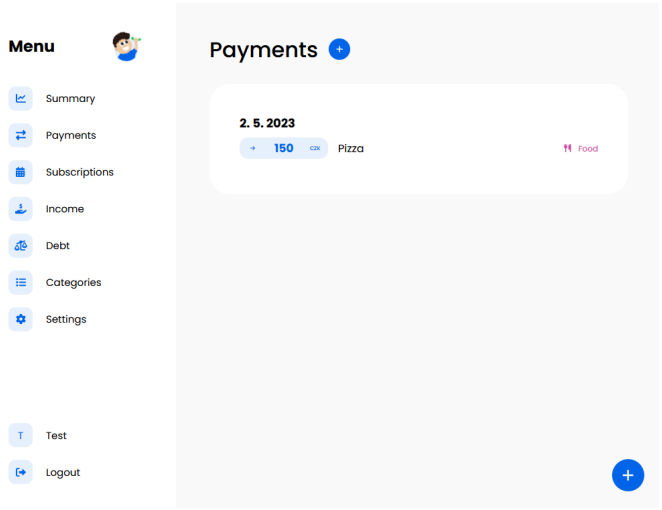


Obrázek 4.6. Varianty výběrových polí.

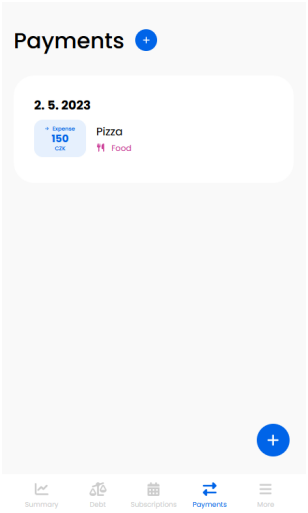
- Ikony
  - Použity jsou ikony z FontAwesome, které jsou zdarma.
  - Pro ukazatel načítání byla navržená vlastní ikona.

- Rozložení
  - Desktopová aplikace má v levé části statické postranní menu, v pravé části se nachází obsah aktuální stránky.

- Mobilní aplikace má postranní menu schované a primární navigace probíhá přes spodní menu.
- Spodní menu obsahuje hlavní 4 stránky a tlačítko menu, které zobrazí celé postranní menu se všemi stránkami.
- Každá stránka obsahuje velký nadpis. Stránky, které slouží k zobrazení seznamu nějakých položek, obsahují vedle nadpisu tlačítko pro přidání nové položky.



Obrázek 4.7. Desktopové rozložení.



Obrázek 4.8. Mobilní rozložení.

4.2 Frontend

Frontend aplikace je rozhraní, pomocí kterého uživatel komunikuje se serverem. Aplikace je tvořena ve frameworku React a splňuje požadavky pro použití jako PWA. To znamená, že je ji možné nainstalovat na zařízení a přiblížit se tak nativnímu pocitu z používání.

4.2.1 Instalovatelnost

Pro implementaci PWA je potřeba, aby aplikace obsahovala platný manifest soubor specifikující jméno aplikace, různé velikosti ikon a barvy. Dále je třeba, aby aplikace měla registrovaného service workera. O to se zde stará knihovna workbox.

Na zařízeních s operačním systémem Android se při otevření webové stránky s aplikací automaticky doporučí instalace aplikace.

Apple je striktnější, co se týče bezpečnosti a tedy i PWA. Aplikaci je možné přidat na domovskou obrazovku pouze z prohlížeče Safari. Na rozdíl od Androidu se instalace sama nedoporučí, ale je potřeba zvolit možnost Přidat na domovskou obrazovku v záložce sdílení.

Na obou operačních systémech se přidá ikona na domovskou obrazovku. Po otevření aplikace je aplikace v režimu celé obrazovky a působí tak nativně.

Obrázek 4.9. Logo PWA.

4.2.2 Stylování

Součástí webpacku je SASS, rozšíření CSS. To usnadňuje práci při psaní kaskádových stylů. V SASSu je například možné selektory s pravidly větvit, to zásadně zpřehlední kód a ubere práci repetetivního psaní dlouhých selektorů. Při buildu aplikace se všechny SASS styly převedou na CSS.

Obrázek 4.10. Logo SASS.

4.2.3 Typovaní

Javascript samotný v základu není typovaný jazyk. Pro dlouhodobou udržitelnost projektu je dobré použít nějakou z nadstaveb pro typování javascriptu. Zde byl zvolen Typescript. Otypované jsou props všech komponent i celé API. Typy slouží hlavně ke kontrole pro vývojáře, při buildu aplikace se tedy odeberou a Typescript kód se převede na čistý Javascript.

Obrázek 4.11. Logo Typescript.

4.2.4 Životní cyklus dotazu na backend

Každý endpoint má svůj vlastní hook, který využívá hlavní useApi hook. Podle potřeby některé hooky hned při vytvoření odešlou požadavek na backend a předají funkci pro obnovení dat. Jiné hooky, předávají funkci pro odeslání.

Práce hlavního hooku useApi:

- Načítání
  - Nastaví property loading na true.
  - Loading se vypne poté, co se dotaz úspěšně vrátí, nebo selže.
- První pokus o dotaz
  - Na backend se odešle dotaz se všemi parametry a přidaným access tokenem.
  - Pokud je access token stále platný a tedy se hned první dotaz vrátí, data odpovědi se předají ke kontrole chyb.
- Refresh tokenu

- Pokud první pokus selhal, odešle se na backend dotaz pro nový access token.
- Pokud vypršela platnost i refresh tokenu, dojde k odhlášení uživatele.
- Jinak se nový access token uloží a dojde k druhému odeslání původního dotazu.

- Chyba nedostupného serveru
  - Pokud se odpověď na dotaz vrátí s chybovým kódem a bez specifikovaných chyb, předpokládá se, že backend není dostupný a zobrazí se globální chybová zpráva o nedostupnosti serveru.

- Chyba nedostupné nástěnky
  - K takovému stavu může dojít, pokud je uživatel přepnutý na nástěnce, které již není součástí, nebo již neexistuje.
  - S takovou chybou se vrátí i jméno a id nástěnky, do které se uživatel automaticky přepne.

4.2.5 Životní cyklus pozvánky k nástěnce

- Vytvoření pozvánky
  - Uživatel s rolí vlastníka nástěnky nechá v nastavení nástěnky generovat novou pozvánku.
  - Odešle se dotaz na backend pro vytvoření nové pozvánky.
  - Na backendu se vytvoří JWT s omezenou platností (1 den) obsahující id nástěnky.
  - Zpět na frontend se vrátí odpověď obsahující token pro ověření pozvánky

- Zobrazení pozvánky
  - Vytvoří se odkaz na frontend s routou accept-invite a tokenem pozvánky.
  - Uživateli se zobrazí vyskakovací okno obsahující tento odkaz.

- Přijmutí pozvánky
  - Jiný uživatel otevře odkaz pozvánky.
  - Na backend se odešle token pro přijmutí pozvánky obsahující id nástěnky.
  - Backend ověří platnost tokenu a přidá uživatele k příslušné nástěnce.
  - Zpět na frontend se vrací odpověď obsahující jméno a id nově připojené nástěnky.
  - Frontend se přepne do nové nástěnky.

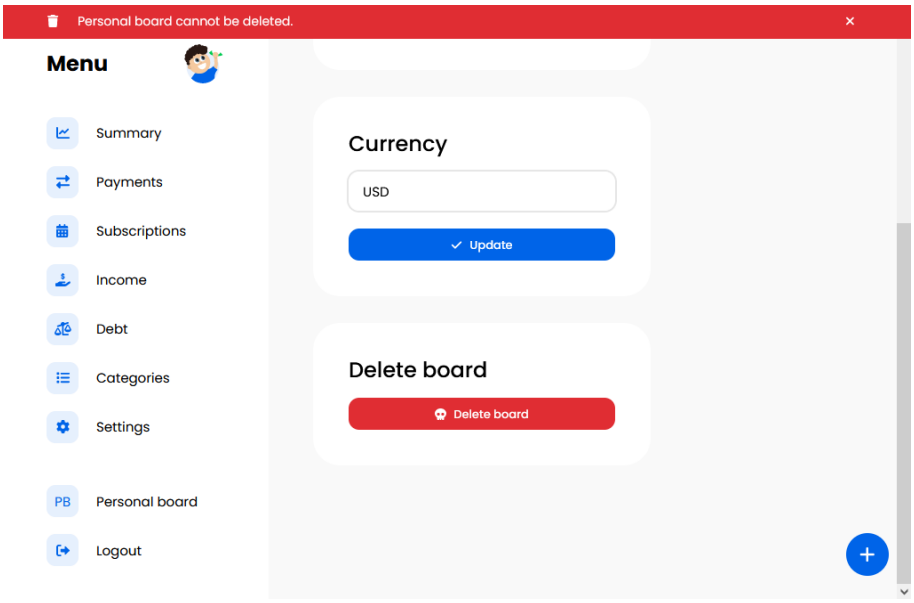
4.2.6 Zobrazení zkratky měny

Na každé nástěnce je možné nastavit právě jednu globální zkratku měny a bylo by tedy zbytečné uvádět zkratku u každé peněžní položky. Místo toho se u dotazů na endpointy s platbami měna posílá pouze jako jeden parametr odpovědi ze serveru. Tento parametr je poté třeba rozdistribuovat do dceřinných komponent na frontendu. To je možné docílit předáváním měny v props, ale react nabízí mnohem elegantnější řešení v podobě hooku useContext.

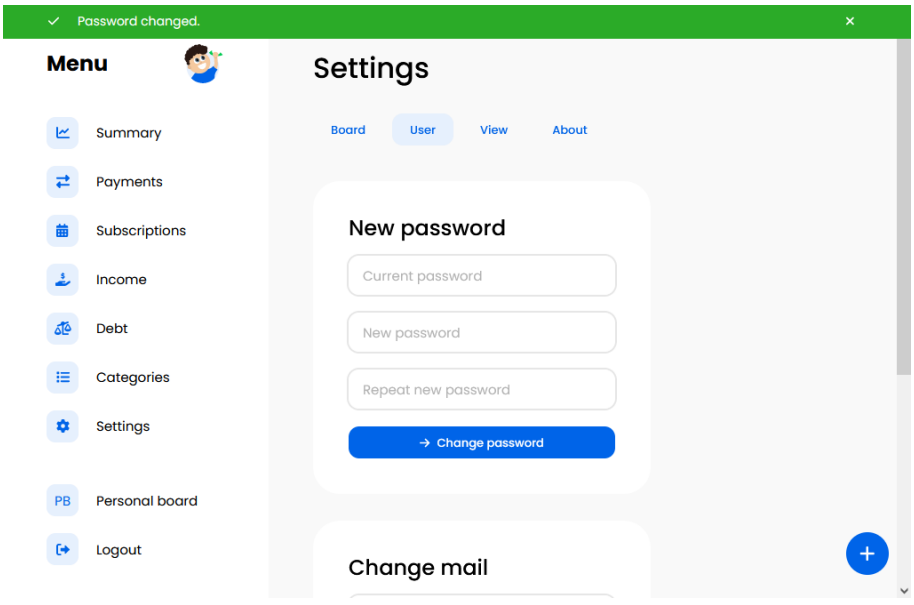
Komponenta, která získala data z backendu obalí dceřinné komponenty do Context-Provideru a dceřinné komponenty mohou následně zkratku měny získat pomocí hooku useContext, bez potřeby předávat tuto hodnotu přes prvky, které jí nepotřebují.

4.2.7 Globální zprávy

V aplikaci se nachází globální zprávy pro zobrazení chybové zprávy a zprávy pro úspěšnou akci. Zobrazení hlášky je možné pomocí hooku, takže je možné je zavolat odkudkoliv.



Obrázek 4.12. Chybová zpráva.



Obrázek 4.13. Zpráva pro úspěšnou akci.

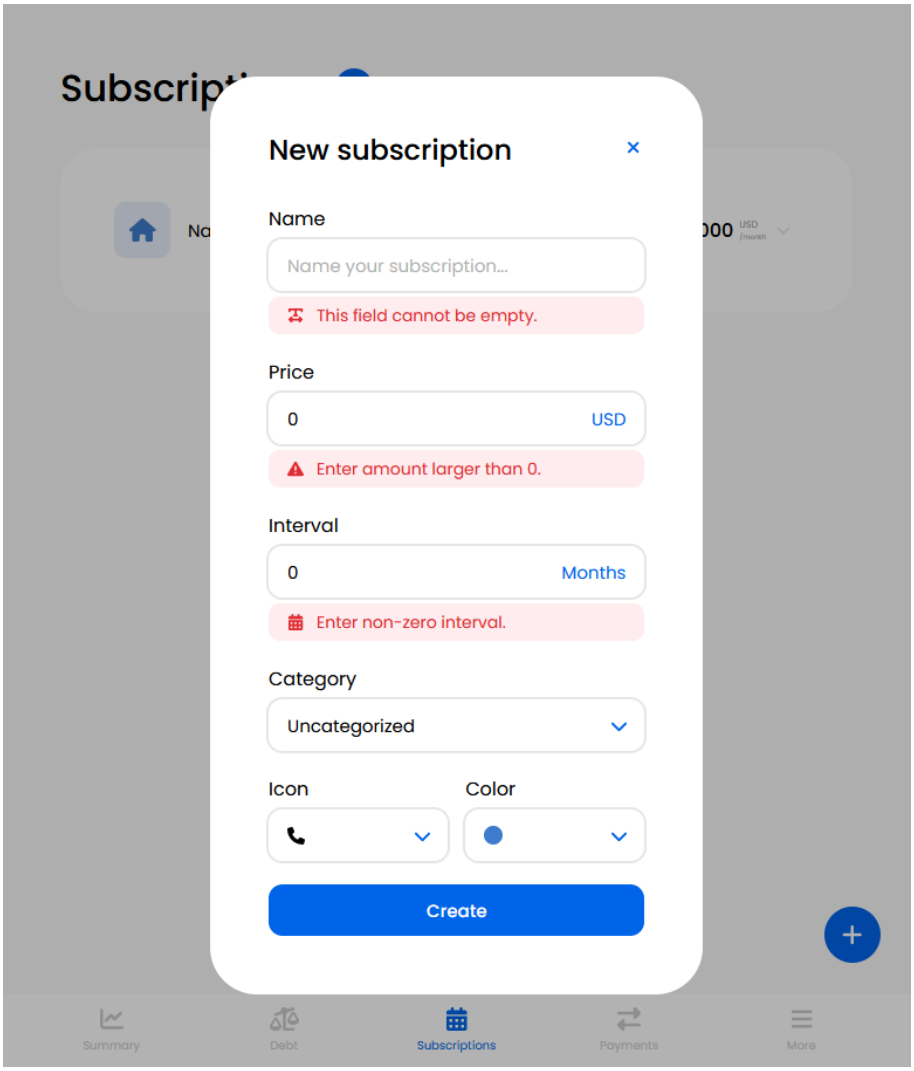
4.2.8 Formulářové chybové zprávy

Pokud uživatel chybně vyplní formulářové pole, je potřeba mu na chybu názorně ukázat, aby ji mohl opravit. Zde pouhé zobrazení globální chybové zprávy nestačí.

Backend zde pracuje ruku v ruce s frontendem. Při validaci dotazu na backendu, server v případě chyby vrací pole nesprávně vyplněných údajů. Každá chyba se skládá

z dvojice údajů, field nese jméno uživatelského pole, ke kterému patří a message obsahuje klíč chyby.

Po obdržení pole chyb se chyby rozdělí k příslušným uživatelským polím, pomocí klíče se vybere přednastavená ikonka popisující význam chyby a z jazykových mutací se vybere překlad zprávy do zvoleného jazyka.



Obrázek 4.14. Schéma databáze.

4.2.9 Lokalizace

Pro implementaci jazykových mutací je použita velmi rozšířená knihovna i18next.

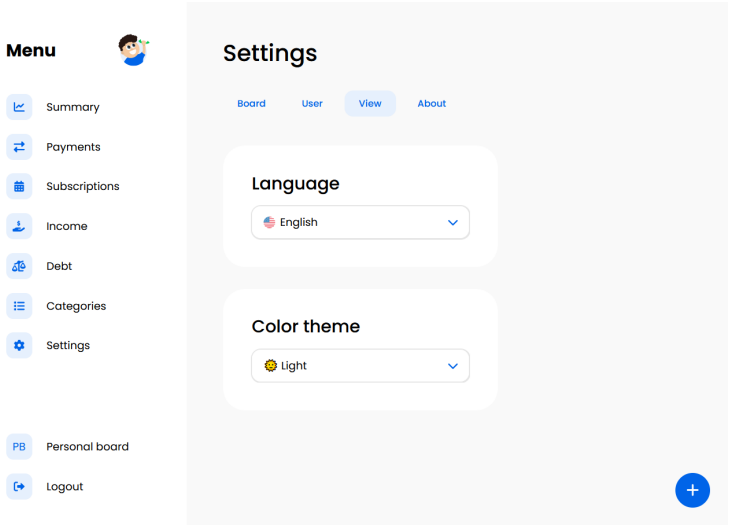
Pro uživatele je také příjemnější, pokud výchozím jazykem bude rovnou jeho jazyk. O detekci jazyka nastaveného ve webovém prohlížeči se stará knihovna i18next-browser-languagedetector. Vybraný jazyk je také propsaný do html tagu pod atributem lang.

Dále je důležité, aby si uživatel mohl jazyk zvolit dle své preference. Tuto volbu má přihlášený uživatel v nastavení aplikace. Vybraný jazyk se ukládá do perzistentního zustand storu a je tedy dostupný i po znovu-načtení stránky.

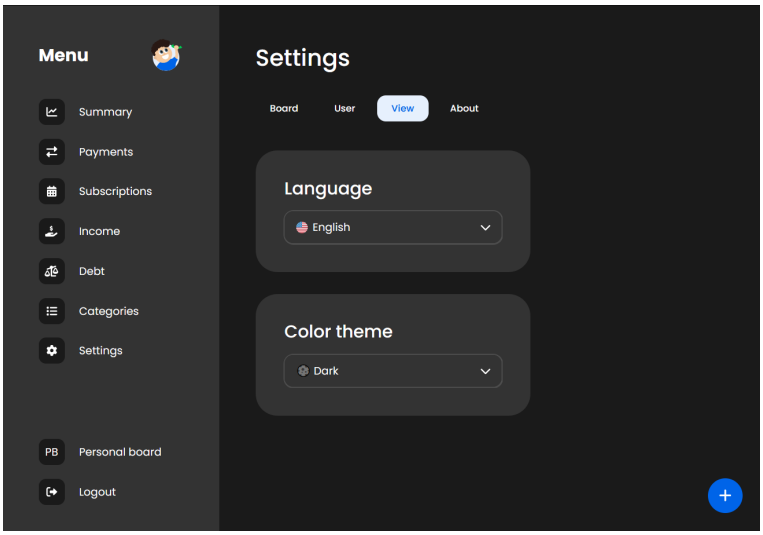
4.2.10 Persistentí data

Ve většině frontendových aplikacích je důležité uchovávat uživatelské preference i po znovu-načtení stránky. Ke správě stavů aplikace je použita knihovna zustand. Zustand také poskytuje jednoduché přepnutí storu na perzistentní. V této aplikaci jsou ukládaná data rozdělená do dvou storů.

- AuthStore
  - Ukládá access token a jméno a id aktuálně vybrané nástěnky.
- SettingsStore
  - Stará se o nastavený jazyk a barevný vzhled stránky.



Obrázek 4.15. Světlý režim.



Obrázek 4.16. Tmavý režim.

4.2.11 Rychlé přidání platby

Jedním z výsledků analýzy bylo poskytnout uživateli rychlé přidání nové platby. Tlačítko pro tuto akci se vyskytuje na všech stránkách aplikace a nachází se v pravém spodním rohu. Po kliknutí se otevře stejné vyskakovací okno jako na stránce plateb, pomocí kterého uživatel dokáže rychle přidat novou jednorázovou platbu.

4.2.12 Stránka přehledu

Stránka přehledu je hlavní stránkou aplikace. Uživatel se zde ocitne po přihlášení nebo zvolení nástěnky. Stránka obsahuje grafy, zobrazující informace o příjmech a výdajích v aktuálním kalendářním měsíci.

Pro zobrazení grafů je převážně využívána knihovna Recharts, která poskytuje různé druhy grafů a mnoho možností pro modifikaci grafů. Všechny grafy byly upraveny tak, aby zapadaly do stylu aplikace.

Logika grafů se převážně nachází na backendu, kde se server rozhoduje, zda graf zobrazit či ne na základě množství dat pro daný graf. Po založení nové nástěnky nejsou k dispozici žádná data a proto se nezobrazí žádné grafy, dokud uživatel nepřidá nějakou platbu.

- Shrnutí

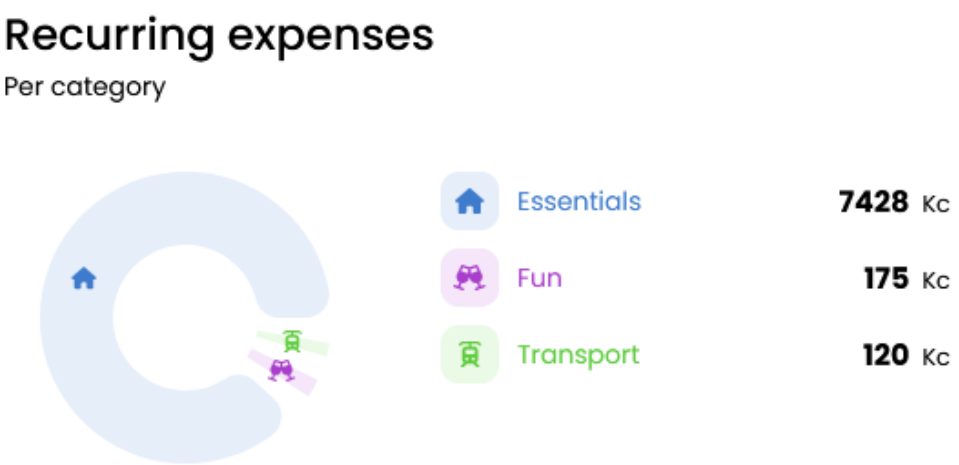
  - Shrnutí obsahuje hlavní informace, které uživatel potřebuje. Proto se nachází v přehledu na pozici prvního grafu.
  - Obsahuje barevně zvýrazněné, zda je uživatel tento měsíc v profitu nebo ve ztrátě.
  - Dále obsahuje přesný součet měsíčních příjmů a výdajů, znázorněný sloupcovým grafem.
  - Tento graf byl stejně jako ostatní grafy původně převzatý z knihovny Recharts, ale v některých situacích docházelo k tomu, že se nezobrazily popisky sloupců a graf v tu chvíli neměl žádnou vypovídací hodnotu.
  - Proto byla vytvořena vlastní komponenta tohoto grafu, která funguje spolehlivě.

Obrázek 4.17. Graf shrnutí při profitu.

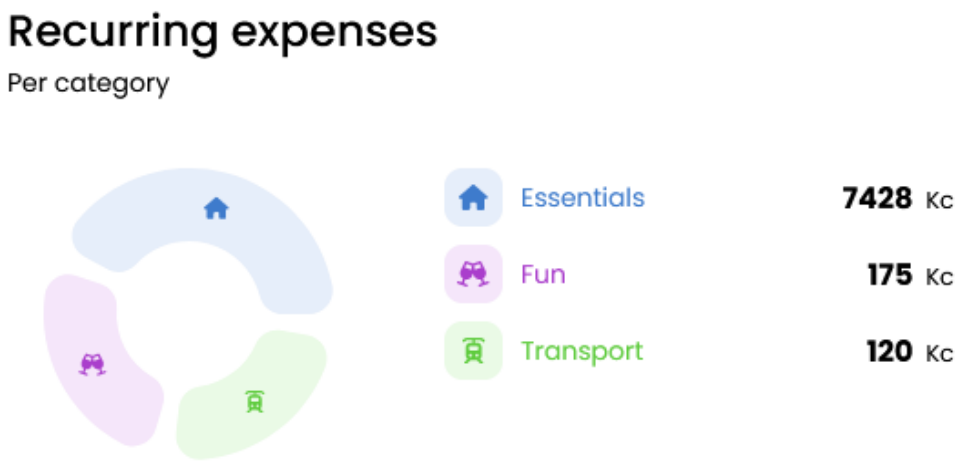
Obrázek 4.18. Graf shrnutí při ztrátě.

- Grafy výdajů podle kategorie

  - Tato sekce je rozdělená na graf pro jednorázové výdaje a graf pro opakující se výdaje.
  - Pokud se v aktuálním měsíci nenachází žádná platba pro daný graf, graf se nezobrazí.
  - Grafy mají mít spíše orientační hodnotu pro přehlednost. K specifickým hodnotám se vedle nich nachází popis. Jednotlivé segmenty koláčových grafů reprezentují kategorie výdajů a mohou nabývat řádově různých hodnot. Grafy v takových případech vypadají nečitelně.
  - Proto byly hodnoty mapované přes logaritmus. Velikosti jednotlivých segmentů to přiblíží a graf je čitelnější.

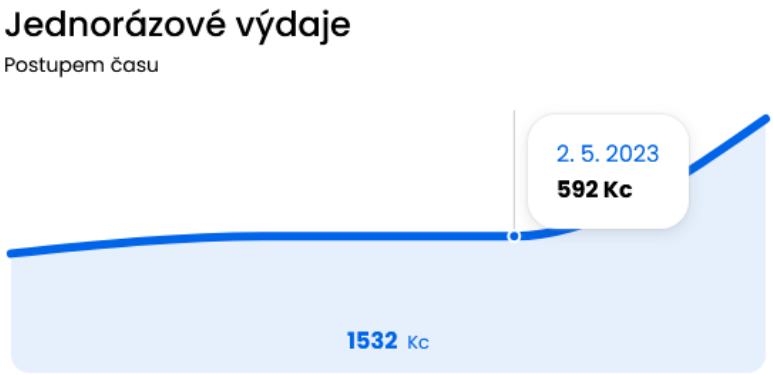


Obrázek 4.19. Koláčový graf bez mapování.



Obrázek 4.20. Koláčový graf s mapováním.

- Kumulativní útrata postupem času
  - Graf zobrazuje nashromážděnou útratu v kalendářním měsíci k aktuálnímu datu.

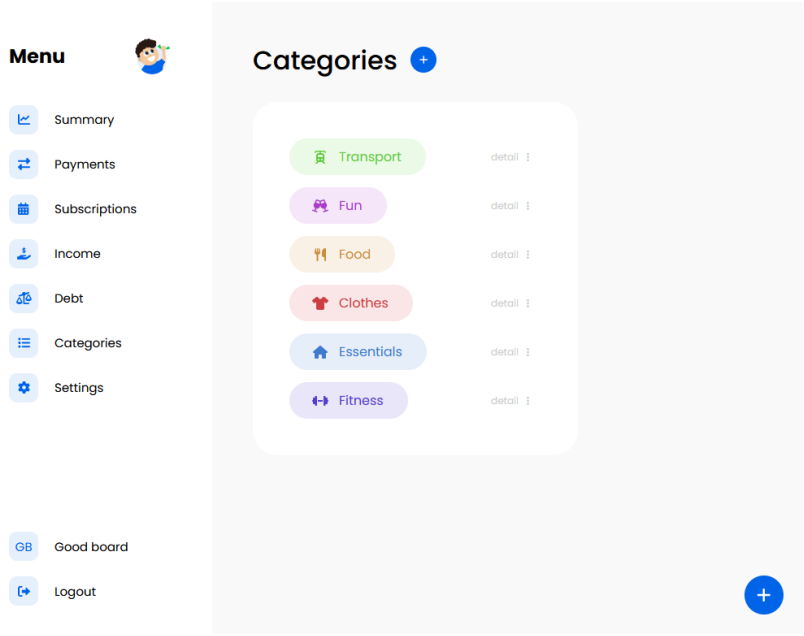


Obrázek 4.21. Graf výdajů postupem času.

4.2.13 Stránky seznamů

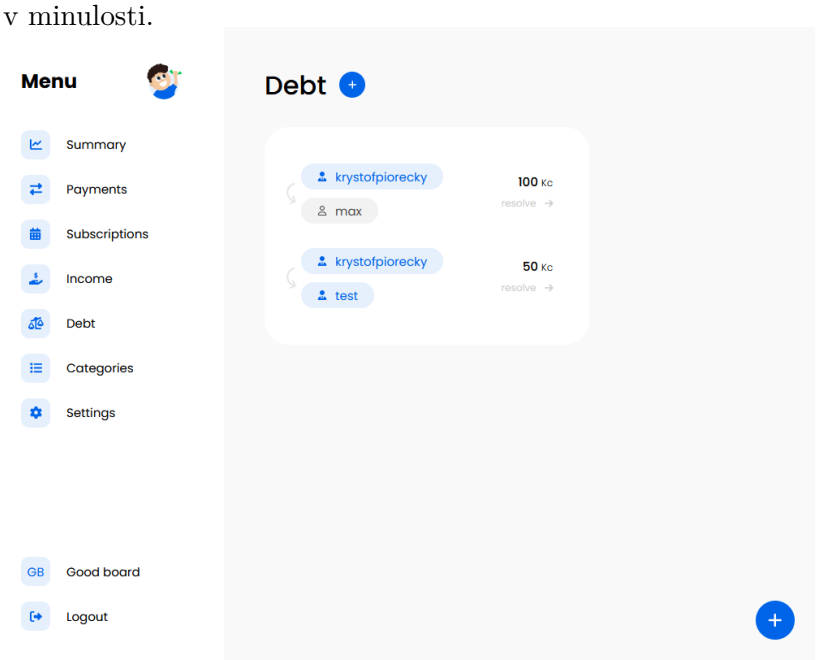
Většina těchto stránek má podobnou charakteristiku a to že, obsahuje seznam položek. U nadpisu se nachází tlačítko pro přidání nové položky. Položky je možné editovat a mazat. K těmto akcím se vždy otevře vyskakovací okno s formulářem pro vyplnění vlastností položky. Před smazáním položky je vždy uživatel dotázán o potvrzení smazání.

- Kategorie
  - Položka kategorie obsahuje ikonu, text a barvu. Stejná komponenta v menší velikosti je pak využívána u plateb, které mají přiřazenou kategorii.



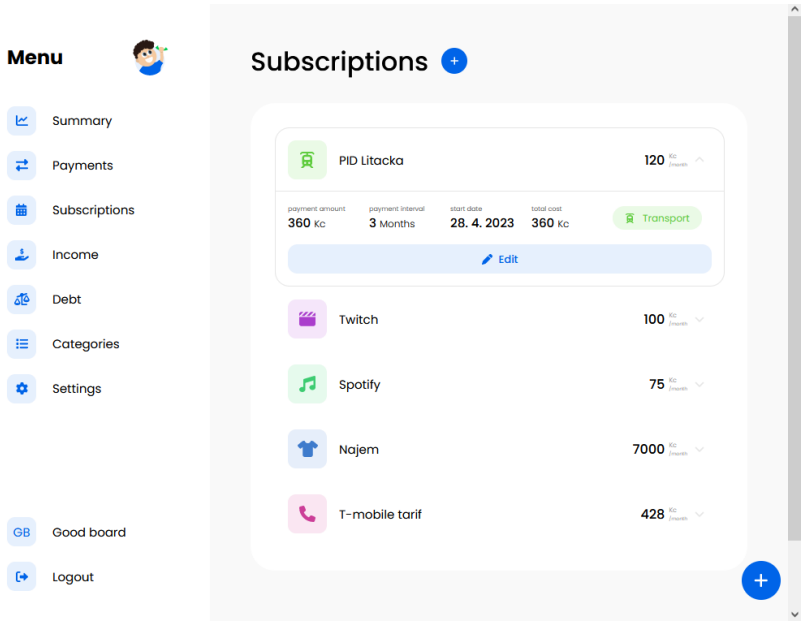
Obrázek 4.22. Stránka kategorií.

- Dluhy
  - Položka seznamu dluhů obsahuje jména uživatelů, kteří mezi sebou mají nějaký dluh.
  - Do položky uživatelského jména je možné zadat i jiného uživatele, než se nachází na nástěnce. Uživatelé na nástěnce jsou zvýrazněni modře.
  - Ve formuláři pro nový dluh se při psaní uživatelského jména zobrazují doporučení. Mezi doporučené jména patří jména uživatel na nástěnce, ale také ostatní jména zadaná v minulosti.



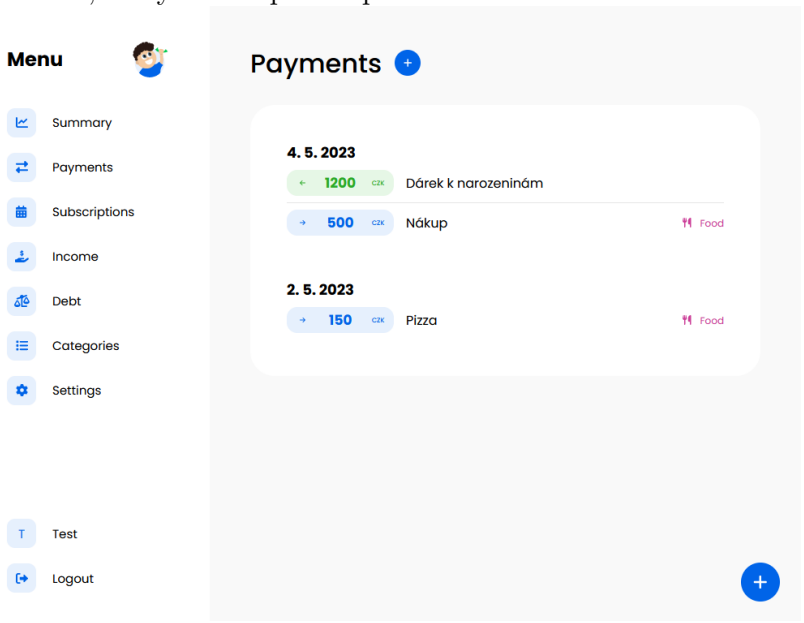
Obrázek 4.23. Stránka dluhů.

- Příjmy a výdaje
  - Obě stránky sdílí stejný grafický návrh.
  - Položky seznamů obsahují příliš mnoho informací. Ve výchozím stavu jsou detaily položky schované, společně s tlačítkem pro editaci. Po kliknutí se detail položky zobrazí.



Obrázek 4.24. Stránka opakujících se výdajů.

- Platby
  - Platby jsou rozdělené do skupin podle data.
  - Platby jsou barevně odlišené na příjmy a výdaje.
  - Pokud se na nástěnce nachází více než jeden uživatel, u plateb bude zobrazené jméno uživatele, který danou platbu přidal.

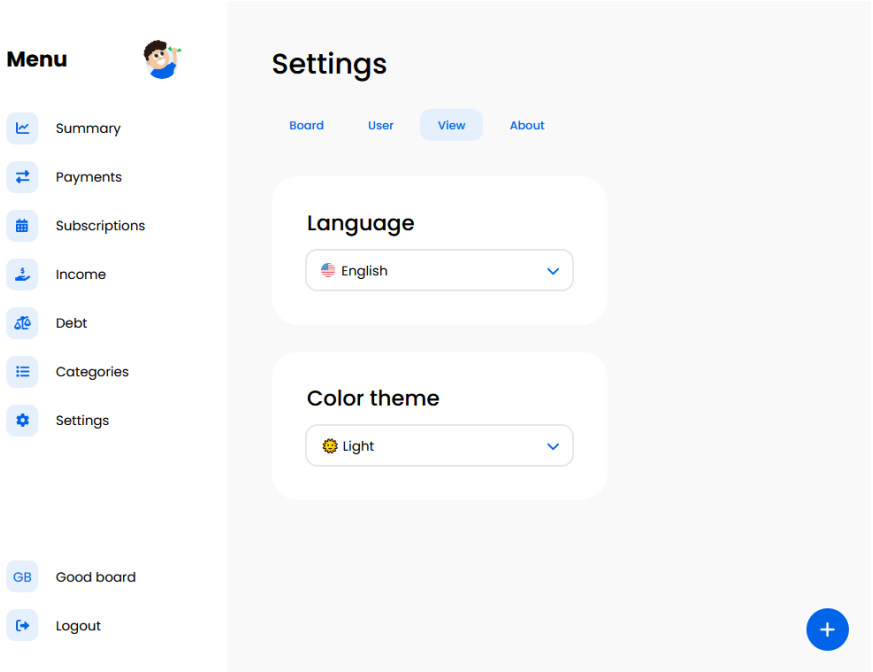


Obrázek 4.25. Stránka jednorázových plateb.

4.2.14 Stránka nastavení

Nastavení se dělí na podstránky a přepínání mezi nimi je zařízeno tlačítky pod nadpisem.

- V nastavení nástěnky je možné spravovat seznam uživatel, měnit zkratku měny, nebo případně nástěnku smazat.
- V nastavení uživatele, je možné změnit heslo, nebo mailovou adresu.
- V nastavení zobrazení, je možné změnit jazyk a barvu prostředí.



Obrázek 4.26. Stránka nastavení.

4.3 Backend

Backend aplikace slouží jako propojení mezi frontendovou částí a databází. Poskytuje informace na základě uživatelských práv tak, aby byla soukromá data uživatelů v bezpečí.

Propojení s databází je řešené pomocí oficiální knihovny MongoDB. Při spouštění backendu na serveru se nejdříve zkontroluje dostupnost databáze a při úspěšném připojení pokračuje další inicializace backendu.

Komunikace s frontendovou částí probíhá pomocí REST api. U endpointů, které vyžadují přihlášení uživatele, se identita uživatele kontroluje pomocí autorizačního bearer tokenu.

Pro bezpečnou komunikaci je využita dvojice JWT tokenů. Každý token má vlastní secret. Část tokenu payload je možné dekódovat a tedy v ní udržovat důležité informace pro rozpoznání uživatele. Díky tomu není třeba tokeny udržovat v databázi. Stačí ověřit platnost tokenu pomocí příslušného secretu a z dekódované části získat id uživatele.

- Refresh token
  - Refresh token je token s delší platností (7 dní). Token je dostupný pouze pro backend a je generovaný nový při každém přihlášení.
  - Po vypršení platnosti refresh tokenu je uživatel odhlášen a musí znovu zadat přihlašovací údaje, aby měl přístup do aplikace.

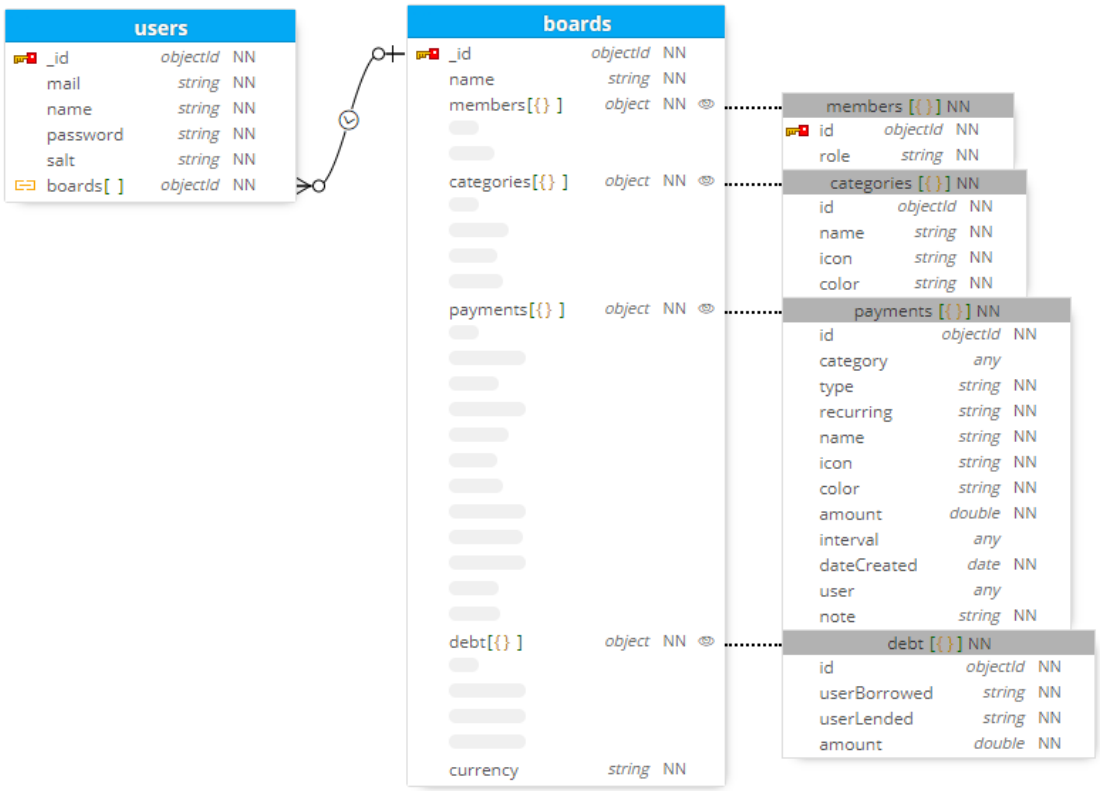
- Access token
  - Access token je token s krátkou platností (15 minut).
  - Tento token uživatel obdrží při přihlášení a je dostupný na frontendu.
  - Access token se posílá v každém dotazu na backend k identifikaci uživatele.
  - Po vypršení platnosti se využívá endpoint refresh, který pomocí refresh tokenu (pokud je stále platný) vygeneruje uživateli nový access token.

4.3.1 Databáze

Databáze je hostovaná na oficiální hostovací službě MongoDB Atlas. Databáze je pro bezpečnost nastavená tak, aby přijímala dotazy pouze z IP adresy serveru, na kterém běží backend aplikace.

Jelikož MongoDB spadá mezi NoSQL databáze, je přístup k ukládání dat jiný než u klasických SQL databází. Práce s daty je převážně řešena na straně backendu.

- V architektuře jsou také využívány embedded dokumenty, které nesídlí ve své vlastní collection, ale jsou součástí jejich mateřského dokumentu.
  - Například dokumenty plateb a dokumenty kategorií, patří vždy pouze k jedné nástěnce, jedná se tedy o embedded dokumenty v dokumentu nástěnky.
- Také je zde pro rychlejší přístup k datům využita redundance informací.
  - Mezi uživateli a nástěnkami je M:N relace a v aplikaci se nachází využití pro zobrazení uživatelů v nástěnce i zobrazení nástěnek jednoho uživatele.
  - Uživatel u sebe má uvedený seznam nástěnek a nástěnky u sebe mají uvedený seznam uživatel.



Obrázek 4.27. Schéma databáze.

4.3.2 Šifrování

Hesla jsou šifrována pomocí sha512. Každému uživateli je generován vlastní salt řetězec.

Když se uživatel registruje, uloží se do databáze salt řetězec a samotné zašifrované heslo. Při následných dotazech na přihlášení, se zadané heslo znovu zašifruje a zašifrovaný řetězec se porovná s uloženým otiskem hesla v databázi.

4.3.3 API

Aplikační rozhraní backendové částí se skládá z endpointů reprezentujících jednotlivé funkcionality. Rozdělené jsou do kategorií podle vyžadovaných práv. Parametry se nacházejí v těle dotazu ve formátu JSON.

Zkratka	Popis
U	Nepřihlášený uživatel
A	Přihlášený uživatel (platný access token).
AA	Uživatel s přistupem k nástěnce
AAA	Uživatel, který je vlastníkem nástěnky

Tabulka 4.2. Tabulka druhů endpointů.

REST api metody a jejich využití

- POST
  - Založení nové entity, registrace a přihlášení.
- GET
  - Čtení informací z backendu.
- PUT
  - Editace existující entity.
- DELETE
  - Mazání existující entity.

Práva	REST	Název	Popis
U	POST	login	Přihlášení uživatele.
U	POST	registration	Vytvoření nového uživatelského účtu.
U	GET	refresh	Generování nového access tokenu.
A	PUT	changePassword	Změna hesla.
A	PUT	changeMail	Změna mailu.
A	GET	baords	Seznam nástěnek uživatele.
A	PUT	baords	Založení nové nástěnky.
AAA	DELETE	baords	Smazání nástěnky.
AA	GET	summary	Generování dat pro grafy souhrnu.
AA	GET	categories	Seznam kategorií nástěnky.
AA	POST	categories	Nová kategorie.
AA	PUT	categories	Editace kategorie.
AA	DELETE	categories	Smazání kategorie.
AA	GET	payments	Seznam plateb nástěnky.
AA	POST	payments	Nová platba.
AA	PUT	payments	Editace platby.
AA	DELETE	payments	Smazání platby.
AA	GET	debtUsernames	Seznam uživatelů, vyskytujících se u dluhů.
AA	GET	debt	Seznam dluhů na nástěnce.
AA	POST	debt	Přidání dluhu.
AA	DELETE	debt	Smazání dluhů mezi dvěma uživateli.
AAA	GET	generateBoardInvite	Vytvoření nové pozvánky k nástěnce.
A	POST	acceptBoardInvite	Přijmutí pozvánky k nástěnce.
AA	GET	members	Seznam členů nástěnky.
AA/AAA	DELETE	members	Opuštění/odebrání člena nástenky.
AA	GET	currency	Nastavená zkratka měny.
AA	PUT	currency	Změna zkratky měny.

Tabulka 4.3. Tabulka seznamu endpointů.

4.3.4 Struktura

- Handle
  - Složka handle obsahuje handlery jednotlivých endpointů.
- Entity
  - Složka entity obsahuje třídy jednotlivých entit, v nich se pak nacházejí metody pro manipulaci s danou entitou.
- Utilities
  - Složka utilities obsahuje pomocné funkce pro práci s daty.
- Třída Check
  - Kontroluje správnost vstupu jednotlivých endpointů.
  - Každý endpoint specifikuje své povinné parametry a případně přijatelné hodnoty těchto parametrů.
  - Check zkontroluje správnost vstupu a případně vrátí chybovou zprávu.
- Třída Db

- Kontroluje dostupnost databáze při spuštění serveru.

4.3.5 Životní cyklus dotazu

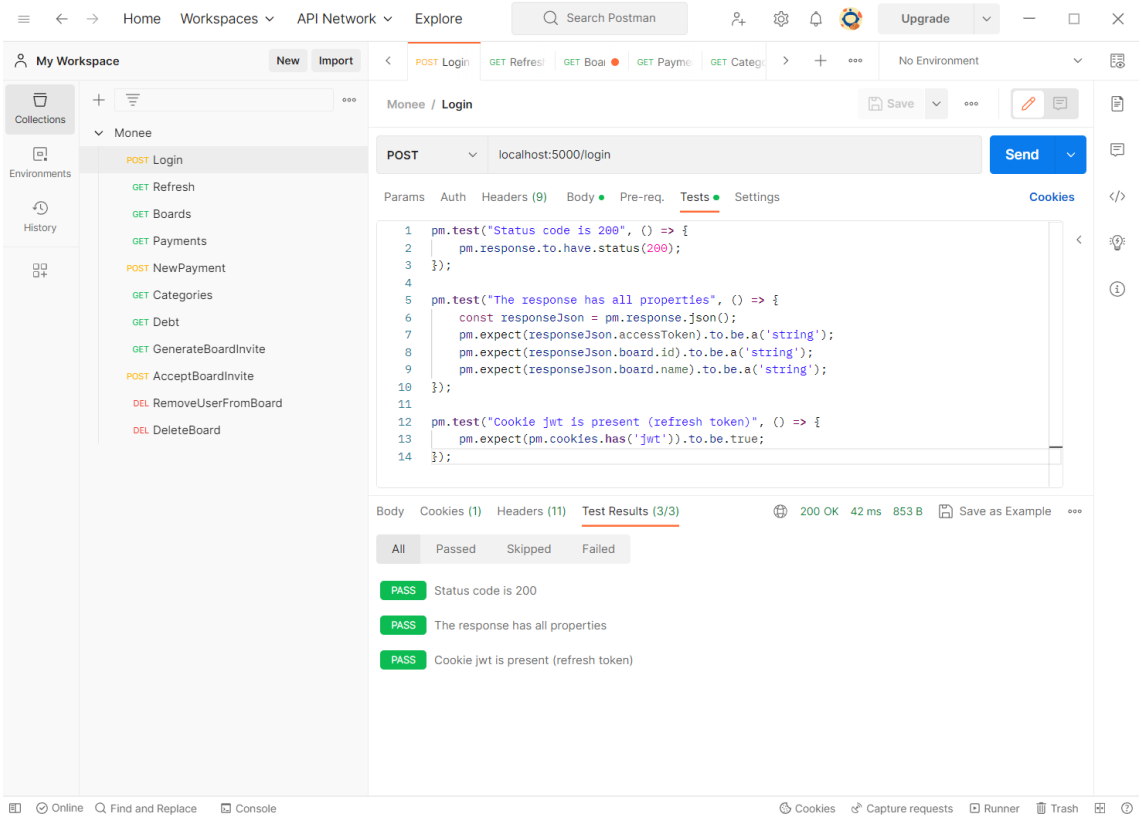
- Výběr handleru
  - Podle URL a metody dotazu a se vybere příslušný handler.
- Autorizace
  - Pokud se jedná o handler s povinnou autorizací, zkontroluje se, zda se v dotazu nachází access token a zda je platný.
  - V případě platného tokenu se dekóduje id uživatele, které je následně předané handleru.
- Kontrola parametrů
  - Pomocí třídy Check se zkontrolují povinné parametry.
- Logika
  - Proběhne logika handleru. Může se jednat o dotazy na databázi, nebo třeba výpočty.
  - Zde může dojít k chybám, v takovém případě handler vrací chybovou zprávu.
- Odpověď
  - Celá logika dotazu byla správně vykonána a handler vrací odpověď ve tvaru JSON.

Kapitola 5
Testování

Testování aplikace při vývoji softwaru slouží k ověření, zda aplikace pracuje dle očekávání a splňuje požadované specifikace a požadavky. Pomocí testování aplikace již při jejím vývoji je možné dříve odhalit chyby či nedostatky a minimalizovat takové chyby v produkci aplikace.

5.1 Testování backendu

Při vývoji backendu aplikace byl použit nástroj pro testování API Postman. Díky tomuto nástroji bylo možné vyvíjet nejdříve samotný endpoint backendu, bez potřeby hotového frontendu.



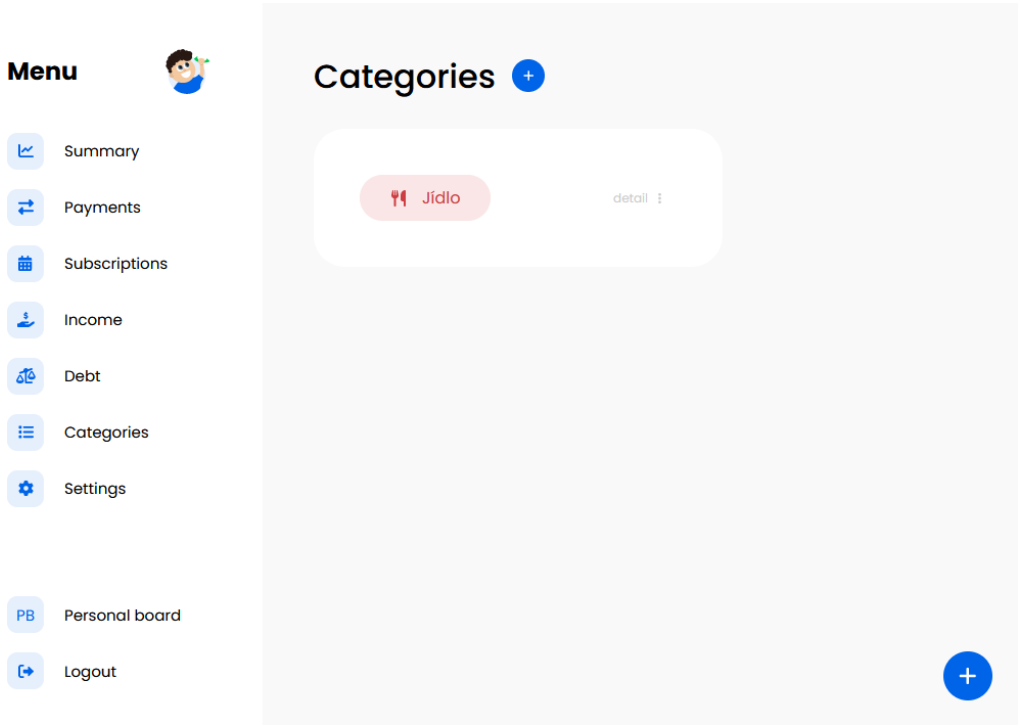
Obrázek 5.1. Ukázka testu v nástroji Postman.

5.2 Testování frontendu

Pro otestování frontendu aplikace byly použity uživatelské testy. Uživatelské testování je proces ověření chování uživatelského rozhraní a chování aplikace.

5.2.1 Testování stránek

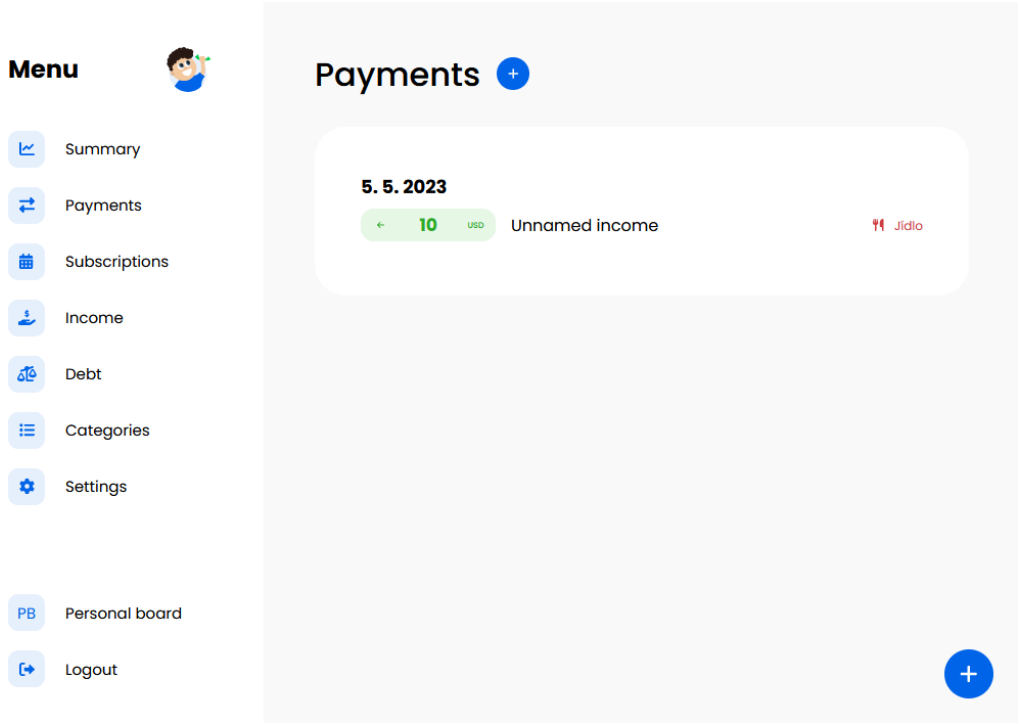
- Uživatel se zaregistruje do aplikace s čistou databází pod uživatelským jménem admin.
- Uživatel přidá novou kategorii. Postup:
  - Uživatel v menu vybere položku kategorie.
  - Uživatel klikne na tlačítko plus u nadpisu stránky. Musí se otevřít vyskakovací okno.
  - Uživatel zadá jméno kategorie `Jídlo`, vybere ikonu příboru, červenou barvu a potvrdí formulář kliknutím na tlačítko Vytvořit.
  - Vyskakovací okno se musí zavřít a stránka kategorie musí obsahovat právě jednu kategorii se jménem `Jídlo`, ikonou příboru a červenou barvou.



Obrázek 5.2. Test přidání kategorie.

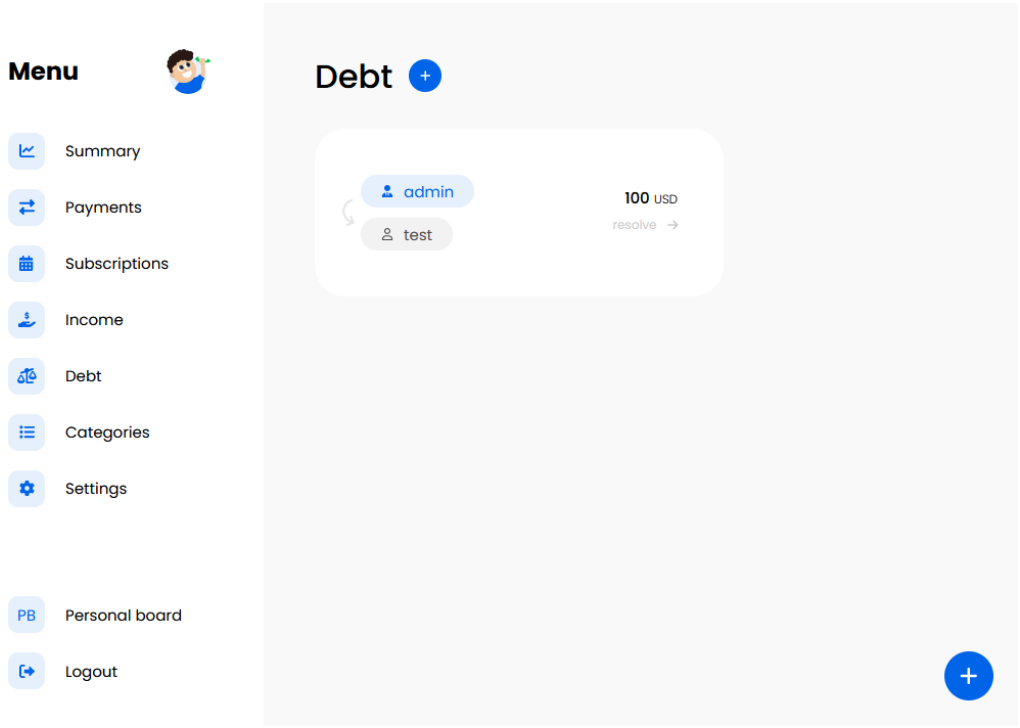
- Uživatel přidá novou platbu. Postup:
  - Uživatel klikne na ikonu plus v pravém spodním rohu. Musí se otevřít formulář s přidáním nové platby.
  - Uživatel klikne v přepínači na variantu Příjem. Přepínač musí zvýraznit vybranou položku.
  - Uživatel klikne do vstupního pole pro Množství. Musí se zobrazit pomocná tlačítka pro zadání množství. Uživatel na klávesnici zadá `2*` a klikáním postupně zvolí `3+4`. Pole musí mít text `2*3+4` a na pravé straně musí být výsledek 10.
  - Uživatel klikne do vstupního pole pro Množství. Musí se zobrazit pomocná tlačítka pro zadání množství. Uživatel na klávesnici zadá `2*` a klikáním postupně zvolí `3+4`. Pole musí mít text `2*3+4` a na pravé straně musí být výsledek 10.

- Uživatel klikne do vstupního pole pro Kategorii. Musí se otevřít možnosti obsahující právě jednu kategorii. Kategorie musí mít jméno `Jídlo`, ikonu příboru a červenou barvou. Uživatel zvolí tuto kategorii.
- Uživatel potvrdí formulář kliknutím na tlačítko Vytvořit. Vyskakovací okno se musí zavřít.
- Uživatel v menu vybere položku platby.
- Stránka plateb musí obsahovat právě jednu platbu. Platba musí mít zvolenou kategorii `Jídlo`. Částka platby musí být 10 a musí být zeleně zvýrazněna.



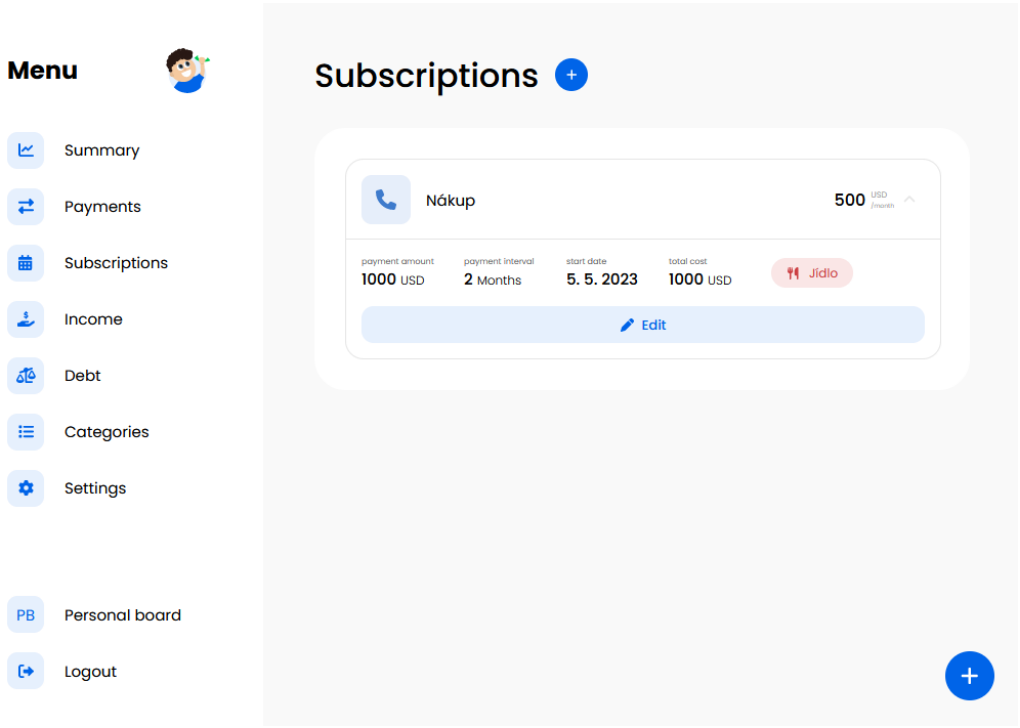
Obrázek 5.3. Test přidání platby.

- Uživatel přidá nový dluh. Postup:
  - Uživatel v menu vybere položku Dluhy.
  - Uživatel klikne na tlačítko plus u nadpisu stránky. Musí se otevřít vyskakovací okno.
  - Uživatel zadá množství 100.
  - Uživatel klikne do položky Kdo si půjčil. Musí se otevřít nápověda obsahující právě jednu položku, která je jméno přihlášeného uživatele. Uživatel zvolí své jméno.
  - Uživatel vyplní jméno položky Kdo půjčil textem `test`.
  - Uživatel potvrdí formulář kliknutím na tlačítko Vytvořit. Vyskakovací okno se musí zavřít.
  - Stránka dluhů musí obsahovat právě jednu položku.
  - Jméno před šipkou musí obsahovat jméno přihlášeného uživatele a musí být zvýrazněno modře.
  - Jméno za šipkou musí obsahovat jméno `test` a musí mít šedé pozadí.
  - Částka položky musí být 100.



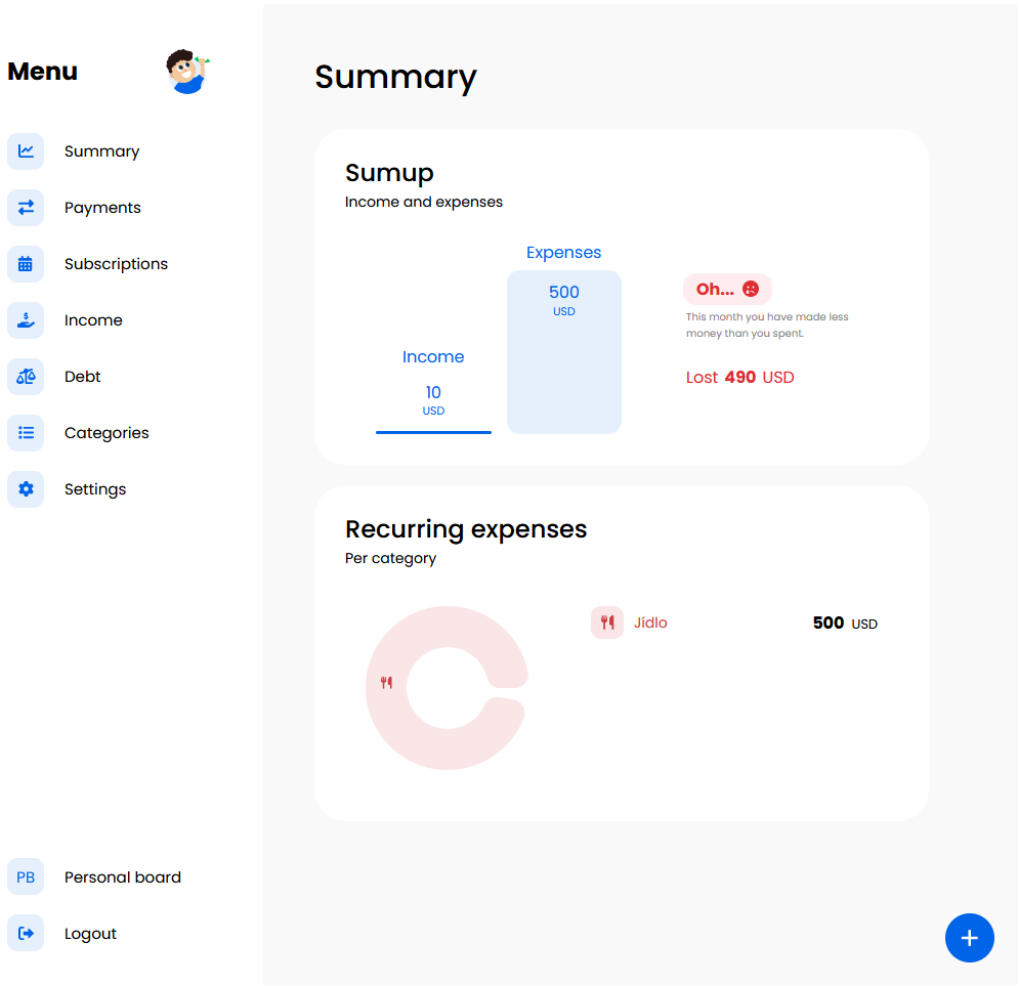
Obrázek 5.4. Test přidání dluhu.

- Uživatel přidá nový výdaj. Postup:
  - Uživatel v menu vybere položku Výdaje.
  - Uživatel zadá jméno výdaje `Nákup`.
  - Uživatel zadá cenu 1000.
  - Uživatel zadá interval 2 měsíce.
  - Uživatel vybere kategorii Jídlo.
  - Uživatel potvrdí formulář kliknutím na tlačítko Vytvořit. Vyskakovací okno se musí zavřít.
  - Stránka výdajů musí obsahovat právě jednu položku.
  - Uživatel klikne na tuto položku.
  - Ikona položky musí být telefon, barva musí být modrá a jméno výdaje musí byt `Nákup`.
  - Měsíční cena výdaje v pravé části položky musí být 500.
  - V detailu položky musí být hodnoty údajů odpovídající údajům zadaným ve formuláři při vytváření výdaje.



Obrázek 5.5. Test přidání výdaje.

- Kontrola grafů:
  - Uživatel v menu vybere položku Přehled.
  - Musí být zobrazené grafy pouze pro Shrnutí a pro Opakující se výdaje.
  - Graf shrnutí musí obsahovat sloupec s popisem Příjmy a hodnotou 10, sloupec s popisem příjmy a hodnotou 500.
  - Shrnutí musí obsahovat červeně zvýrazněný text Ztráta 490.
  - Opakující se výdaje musí obsahovat právě jednu položku pro kategorii `Jídlo` s ikonou příboru a červenou barvou. Hodnota kategorie `Jídlo` musí být 500.



Obrázek 5.6. Test kontroly grafů.

5.2.2 Testování navigace

- Uživatel se zaregistruje do aplikace.
- Uživatel změní jméno stránky v url adrese ze summary na neplatne-jmeno. Musí se zobrazit stránka pro chybu 404 s chybovou ikonou aplikace.
- Uživatel změní jméno stránky v url adrese na jakoukoliv platnou adresu aplikace. Musí být přesměrován na danou stránku.
- Uživatel klikne položku ve spodním nebo postranním menu aplikace. Musí být přesměrován na danou stránku.

5.2.3 Testování responzivity

- Uživatel se zaregistruje do aplikace.
- Uživatel otevře aplikaci na mobilním telefonu. Žádné prvky uživatelského rozhraní nesmí přesahovat obrazovku. Spodní navigační menu musí být viditelné.
- Uživatel otevře aplikaci na počítači s maximalizovaným oknem prohlížeče. Spodní navigační menu určené pouze pro telefon nesmí být viditelné. Musí být viditelné postranní menu.

5.3 Závěr

Pomocí nástroje Postman byl otestován backend a nebyly nalezeny žádné chyby. Na frontendové částí za pomoci uživatelského testování nebyly objeveny žádné závažné chyby. Bylo objeveno několik drobných nedostatků, které byly následně opraveny.

Kapitola 6
Závěr

Cílem této bakalářské práce bylo vytvořit aplikaci, která bude pomáhat jednotlivcům, nebo domácnostem udržovat přehled o jejich výdajích a příjmech. Nejprve byla provedena analýza nutná pro tvorbu takové aplikace. Následně byla aplikace naimplementována a otestována. Vzhledem k důkladné analýze probíhala implementace bez větších zásek̊u a potřeby předělávat části aplikace. Testování již při vývoji pomohlo odladit chyby včas.

Aplikace je plně funkční, připravena k používání.

6.1 Další možná rozšíření aplikace

6.1.1 Ukončení opakujících se plateb

- V aktuálním stádiu aplikace není možné ukončit opakující se platbu a stále ji zachovat v systému.
- Kvůli tomu není možné správně zobrazovat shrnutí za minulé měsíce.
- Opakující se platby by mohli mít nepovinný parametr data ukončení a v seznamu plateb by bylo třeba odlišit aktivní a ukončené platby.
- Backend by poté mohl správně vypočítat příjmy a výdaje za předchozí období a stránka shrnutí by tak mohla nabídnout uživateli výběr počátečního a koncového data intervalu a zobrazit vypočtená data.

6.1.2 Offline režim

- Pro offline režim aplikace bude v prvé řadě potřeba správně nastavit PWA integraci, aby ukládala soubory frontedu a ten tak byl dostupný i bez připojení k internetu.
- Následně bude třeba uchovávat všechna data o platbách i na frontendu.
- Při připojení k internetu bude třeba aktualizovat všechny provedené změny i na backendu a vyřešit případné kolize.

Literatura

[1] *ABRAN A. Moore J. W., Bourque P. Guide to the software engineering body of knowledge. In: IEEE Intellectual Property Rights Office, 445 Hoes Lane, Piscataway: Los Alamitos, CA IEEE Computer Society Press, 2004. isbn 978-07-695-5166-1 [Online] [Cit. 13. 5. 2023].* https://www.math.unipd.it/~tullio/IS-1/2007/Approfondimenti/SWEBOK.pdf.

[2] *React, A JavaScript library for building user interfaces [Online] [Cit. 13. 5. 2023].* https://reactjs.org/.

[3] *2022 Developer Survey [Online] [Cit. 13. 5. 2023].* https://survey.stackoverflow.co/2022/.

[4] *What is Angular? [Online] [Cit. 13. 5. 2023].* https://angular.io/guide/what-is-angular.

[5] *What is Vue? [Online] [Cit. 13. 5. 2023].* https://vuejs.org/guide/introduction.html.

[6] *What is Xamarin? [Online] [Cit. 13. 5. 2023].* https://learn.microsoft.com/en-us/xamarin/get-started/what-is-xamarin.

[7] *What is Flutter? [Online] [Cit. 13. 5. 2023].* https://docs.flutter.dev/resources/faqwhat-is-flutter.

[8] *Introduction to progressive web apps [Online] [Cit. 13. 5. 2023].* https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Introduction.

[9] *What is PHP? [Online] [Cit. 13. 5. 2023].* https://www.php.net/manual/en/intro-whatis.php.

[10] *What is Java technology and why do I need it? [Online] [Cit. 13. 5. 2023].* https://www.java.com/en/download/help/whatis_java.html.

[11] *About Node.js [Online] [Cit. 13. 5. 2023].* https://nodejs.org/en/about/.

[12] *About MySQL [Online] [Cit. 13. 5. 2023].* https://www.mysql.com/about/.

[13] *What Is MongoDB? [Online] [Cit. 13. 5. 2023].* https://www.mongodb.com/what-is-mongodb.

[14] *Roger S. Pressman, Bruce R. Maxim: Software Engineering: A Practitioner's Approach, McGraw-Hill Education, Edition 8, ISBN 1259253155, 9781259253157.*

[15] *Financer, Aplikace správce financí [Online] [Cit. 13. 5. 2023].* https://financer.com/cz/blog/aplikace-spravce-financi/.

[16] *Správa financí - výdajů, peněz [Online] [Cit. 13. 5. 2023].* https://play.google.com/store/apps/details?id=ru.innim.my_finance&hl=cs&gl=US&pli=1.

Příloha A
Zkratky a symboly

A.1 Zkratky

iOS	iPhone Operating System
SSO	Single Sign-On
HTML	HyperText Markup Language
CSS	Cascading Style Sheet
JS	JavaScript
XSS	Cross-Site Scripting
JSX	JavaScript XML
URL	Uniform Resource Locator
PwA	Progressive Web App
MERN	MongoDB, Express, React, Node
PHP	PHP (rekurzivní zkratka PHP: Hypertext Preprocessor, česky „PHP: Hypertextový preprocesor“, původně Personal Home Page).