# Požadavky k funkcionalitě

## Funkční požadavky

### 1. **Přehled financí**

* Zobrazení aktuálního zůstatku na všech účtech.
* Přehled příjmů a výdajů za určité časové období.
* Zobrazení příjmů a výdajů
* Grafické zobrazení dat.

### 2. **Správa účtů**

* Možnost přidání více účtů.
* Možnost přidání různých typů účtů.
* Možnost nastavení počátečního stavu účtu nebo získání stavu z banky prostřednictvím API.
* Zobrazení zůstatků jednotlivých účtů.
* Uprava nebo odstranění účtů.
* Podpora více měn, včetně automatických konverzí měn.
* Možnost přiřazení barev k jednotlivým účtům pro lepší přehlednost.
* Možnost přiřazení ikón k jednotlivým účtům.

### 3. **Kategorie výdajů a příjmů**

* Přednastavené kategorie (např. potraviny, doprava, bydlení).
* Možnost vytváření vlastních kategorií.
* Úprava nebo mazání kategorií.
* Možnost přiřazení barev k jednotlivým kategoriím pro lepší přehlednost.
* Možnost přiřazení ikón k jednotlivým kategoriím

### 4. **Přidávání transakcí**

* Ruční zadávání příjmů a výdajů.
* Výběr kategorie transakce.
* Možnost přidat poznámku k transakci.
* Možnost nahrávání účtenek a faktur k transakci.
* Možnost přidat polohu platby.
* Možnost upravit den provedení transakce.

### 5. **Transakční historie**

* Seznam všech transakcí v přehledném zobrazení.
* Možnost filtrovat transakce podle datumu, účtu, kategorie, nebo klíčového slova.
* Export historie transakcí do formátů Excel, XML nebo JSON.
* Možnost mazání nebo úpravy transakcí zpětně.

### 6. **Napojení na bankovní účty**

* Automatický import stavu účtu z banky přes API.

### 7. **Opakované transakce**

* Možnost nastavení opakujících se transakcí (např. měsíční nájem, půjčka).
* Výběr intervalu opakování (denně, týdně, měsíčně, ročně).
* Možnost editace a rušení opakovaných transakcí.

### 8. **Dluhové a spořicí účty**

* Možnost vytváření dluhu.
* Nastavení splátek dluhu za období a procenta.
* Nastavení splátek, procenta a přip. cíle k spořicímu účet

### 9. **Upomínky**

* Nastavení upomínek na platby (např. upomínky před splatností faktur).
* Výběr způsobu notifikací (e-mail, push notifikace).
* Možnost nastavení různých připomínek (den před splatností, týden před splatností apod.).

### 10. **Multijazyčnost**

* Podpora více jazyků.
* Automatická detekce jazyka dle regionu nebo nastavení uživatele.
* Možnost ruční změny jazyka v aplikaci.

### 11. **Bezpečnost**

* Možnost zabezpečení aplikace PIN kódem.
* Šifrování přenašených dat.
* Automatické zálohy dat.

### 12. **Podpora více uživatelů**

* Možnost vytvoření více uživatelských profilů (např. pro rodinu nebo spolubydlící).
* Možnost sdílení účtů nebo rozpočtů mezi uživateli.
* Oddělené přehledy financí pro jednotlivé uživatele.

### 13. **Správa aktiv**

* Možnost přidání aktiv.
* Možnost přidání různých typů aktiv (nemovitost, budova, technické zařízení).
* Zobrazení zůstatků jednotlivých aktiv.
* Uprava nebo odstranění aktiv.
* Možnost přiání odpisu k jednotlivým aktivům.
* Podpora více měn, včetně automatických konverzí měn.
* Možnost přiřazení barev k jednotlivým aktivům pro lepší přehlednost.
* Možnost přiřazení ikón k jednotlivým aktivům.
* Možnost vygenerovat účetní závěrku v podobě Excel tabulky.

## Nefunkční požadavky

### 1. **Výkon**

* **Rychlá odezva**: Aplikace musí reagovat na uživatelské vstupy do 5 sekund pro běžné operace, jako je přidání transakce nebo zobrazení přehledu.
* **Optimalizace pro velký objem dat**: Aplikace musí být schopná pracovat s velkým množstvím transakcí a uživatelských účtů (např. desítky tisíc transakcí) bez výrazného zpomalení.
* **Nízká spotřeba systémových prostředků**: Aplikace by měla být optimalizovaná tak, aby nezatěžovala CPU, paměť (maximálně do 100 Mb) ani síťové zdroje nadměrně.

### 2. **Dostupnost**

* **Vysoká dostupnost**: Aplikace musí být dostupná minimálně 99 % času během roku. To znamená minimalizaci výpadků a plánování údržby mimo špičku užívání.
* **Obnovení po výpadku**: Aplikace musí být schopná automaticky obnovit provoz po výpadku nebo selhání serveru s minimální ztrátou dat.
* **Offline režim**: Aplikace by měla umožňovat omezenou práci offline s automatickou.

### 3. **Škálovatelnost**

* **Podpora růstu uživatelské základny**: Aplikace musí být škálovatelná, aby zvládla růst uživatelů a objemu dat bez poklesu výkonu.
* **Horizontální škálování**: Možnost rozšířit aplikaci na více serverů nebo cloudových instancí, pokud bude potřeba zvýšit výkon.

### 4. **Bezpečnost**

* **Ochrana osobních údajů**: Aplikace musí splňovat nařízení o ochraně osobních údajů a možnosti smazání uživatelských dat.
* **Ochrana proti útokům**: Aplikace musí být chráněna proti běžným bezpečnostním hrozbám, jako jsou SQL injection, XSS (cross-site scripting), brute-force útoky apod.
* **Šifrovaná komunikace**: Veškerá komunikace mezi aplikací a servery musí být šifrovaná pomocí SSL/TLS.

### 5. **Udržovatelnost**

* **Modulární architektura**: Kód aplikace by měl být strukturován modulárně, aby bylo snadné přidávat nové funkce nebo měnit stávající bez vlivu na jiné části aplikace.
* **Dobře zdokumentovaný kód**: Kód aplikace musí být dobře dokumentovaný, aby byl snadno pochopitelný pro vývojáře, kteří na projektu pracují nebo na něm budou pracovat v budoucnu.
* **Automatizované testování**: Aplikace musí mít jednotkové a integrační testy, aby bylo možné rychle a bezpečně provádět úpravy a aktualizace.

### 6. **Uživatelská zkušenost (UX)**

* **Konzistence designu**: Aplikace musí mít konzistentní design a ovládací prvky, aby uživatel snadno pochopil, jak používat různé funkce.
* **Minimální počet kroků**: Ovládání aplikace musí být intuitivní a minimalizovat počet kroků potřebných k dosažení cíle.

### 7. **Zálohování a obnova dat**

* **Automatizované zálohy**: Aplikace musí provádět pravidelné zálohy dat, ideálně každých 24 hodin.

### 8. **Kompatibilita**

* **Podpora více platforem**: Aplikace musí být dostupná na různých platformách (Android, iOS).
* **Integrace s externími službami**: Aplikace by měla být schopná integrace s externími API.

### 9. **Rozšiřitelnost**

* **Podpora pluginů a rozšíření**: Aplikace by měla být navržena tak, aby bylo možné do budoucna přidávat pluginy a rozšíření (např. nové způsoby sledování výdajů, propojení s dalšími finančními službami).
* **API pro třetí strany**: Poskytnutí veřejného API, které umožní externím vývojářům vytvářet doplňky a rozšíření aplikace.

### 10. **Podpora a údržba**

* **Technická podpora**: Uživatelé musí mít možnost kontaktovat technickou podporu a očekávat odpověď.
* **Pravidelné údržby**: Plánované údržby systému musí být oznámeny uživatelům s dostatečným předstihem a prováděny mimo hlavní pracovní hodiny.