

Internetový platební systém PayScam

Projektový [repozitář](#)

Zpracovali: Artur Khairullin, David Tauchen

Obsah

| | |
|---|----|
| Popis aplikace..... | 3 |
| Motivace..... | 3 |
| Strategický záměr projektu..... | 4 |
| Obchodní přínos..... | 4 |
| SWOT analýza..... | 5 |
| Analýza pěti sil (5F)..... | 6 |
| PEST analýza..... | 8 |
| Funkční požadavky..... | 9 |
| Nefunkční požadavky..... | 10 |
| Seznam uživatelů..... | 11 |
| Případy užití..... | 12 |
| UML diagramy..... | 13 |
| Výběr architektury..... | 18 |
| Rozbor a výběr alternativ návrhu řešení..... | 19 |
| WBS – rozdělení projektu na dílčí procesy..... | 20 |
| Zdroje..... | 22 |
| Normy a standardy..... | 23 |
| Harmonogram GANTT..... | 24 |
| Analýza rizik FMEA..... | 24 |
| Znovupoužitelnost..... | 25 |
| Metriky..... | 26 |
| Plán podpory | |
| Cílem plánu podpory je dosáhnout následujících cílů:..... | 26 |
| Vyhodnocení..... | 27 |

Popis aplikace

PayScam je aplikace platebního styku, která umožňuje snadný převod finančních prostředků z jednoho účtu na druhý nebo do jiných platebních systémů. Aplikace má několik funkcí včetně různých kontrol pro zajištění bezpečnosti a také možnost požádat v případě potřeby o vrácení peněz. Kromě své funkčnosti byla aplikace navržena s uživatelsky přívětivým rozhraním, které uživatelům usnadňuje navigaci a rychlé dokončení transakcí. Celkově aplikace platebního systému nabízí pohodlné a bezpečné řešení pro správu finančních transakcí online.

Motivace

Motivací vývoje této aplikace platebního systému bylo vytvořit pohodlnou a bezpečnou platformu pro uživatele pro správu jejich finančních transakcí. S rostoucí popularitou online plateb se ukázalo, že existuje potřeba spolehlivého platebního systému, který by nabízel snadné použití a špičkové zabezpečení. Tento projekt si kladl za cíl tuto potřebu vyřešit poskytnutím uživatelsky přívětivého rozhraní, které uživatelům usnadnilo převod finančních prostředků a provádění plateb, a zároveň zahrnovalo robustní bezpečnostní opatření na ochranu proti podvodům a jiným škodlivým činnostem.

Strategický záměr projektu

Projekt má potenciál být úspěšný díky rostoucí poptávce po pohodlných a bezpečných platebních systémech. S nárůstem elektronického obchodování a online transakcí roste potřeba spolehlivých a účinných metod převodu finančních prostředků. Různé možnosti šeků a refundací začleněné do tohoto platebního systému navíc přidávají další vrstvu zabezpečení a uživatelskou přívětivost, což z něj činí atraktivnější možnost pro spotřebitele. Řešením těchto potřeb a poskytnutím řešení, které je funkční a uživatelsky přívětivé, má tento platební systém potenciál uspět na trhu.

Obchodní přínos

Obchodním přínosem projektu je potenciál generovat příjmy prostřednictvím platebního systému. Na rozdíl od mnoha existujících platebních systémů nebude tento systém účtovat žádnou provizi za transakce, což z něj činí atraktivní možnost pro uživatele. Nicméně firmám a vývojářům, kteří chtějí integrovat platební systém do svých aplikací, bude účtován poplatek. Tento finanční plán poskytuje pro projekt udržitelný zdroj příjmů a potenciál růstu, jakmile systém zavede více podniků. Navíc tím, že nabízí systém plateb bez provize, může projekt přilákat velkou uživatelskou základnu, což má za následek větší viditelnost a potenciál pro budoucí expanzi.

SWOT analýza



Obrázek č.1 - SWOT analýza

Analýza pěti sil (5F)

| Parameter | Význam | Popis | Řešení |
|--|---------|--|--|
| Hrozby vzniku substitutů | Vysoký | <p>Existuje mnoho náhrad za platební systémy, včetně hotovosti, šeků a dalších forem elektronických plateb.</p> <p>Mohou se objevit nové náhražky, jako jsou kryptoměna nebo jiné platební systémy založené na blockchainu.</p> | <p>Zaměřit se na poskytování jedinečných funkcí a výhod zákazníkům, které nelze nalézt u konkurenčních produktů.</p> <p>Prozkoumat partnerství s doplňkovými službami nebo produkty, abychom mohli poskytnout komplexnější řešení.</p> |
| Stávající konkurenti | Vysoký | <p>Trh platebních systémů je vysoce konkurenční s mnoha zavedenými hráči, kteří soutěží o podíl na trhu.</p> <p>Konkurence může vést k cenovým válkám nebo zvýšeným investicím do inovací a funkcí k odlišení od konkurentů.</p> | <p>Zaměřit se na budování silné pověsti značky a zákaznické základny prostřednictvím cíleného marketingového úsilí a partnerství se zavedenými společnostmi.</p> <p>Odlišit se poskytováním jedinečných funkcí nebo výhod zákazníkům a zkoumáním partnerství s cílem poskytnout komplexnější řešení.</p> |
| Hrozby vstupu nových konkurentů na trh | Střední | <p>Trh platebních systémů je již zaveden s mnoha známými hráči, ale noví účastníci stále mohou vstoupit a konkurovat.</p> <p>Mohou existovat překážky vstupu, jako jsou regulační požadavky a budování důvěry u uživatelů.</p> | <p>Prozkoumat příležitosti k expanzi na nové trhy nebo vertikály, abychom omezili přímou konkurenci</p> |

| | | | |
|--------------------------------|--------|---|--|
| Vyjednávací síla zákazníků | Vysoký | <p>Kupující mají k dispozici mnoho možností platebního systému, což jim usnadňuje přechod ke konkurenci.</p> <p>Kupující mohou mít vyjednávací sílu ke sjednání nižších poplatků nebo lepších služeb od poskytovatelů platebních systémů.</p> | <p>Implementovat flexibilní cenové strategie a poskytnout zákazníkům přidanou hodnotu ke zvýšení loajality. Zaměřit se na budování silné značky a pověsti, abychom přilákali a udrželi zákazníky.</p> |
| Vyjednávací síla dodavatelů | Nízký | <p>Existuje mnoho dodavatelů technologií a poskytovatelů služeb, ze kterých si poskytovatelé platebních systémů mohou vybrat, což jim dává vyjednávací sílu k vyjednávání lepších cen a služeb.</p> | <p>Pracovat na budování silných vztahů s klíčovými dodavateli a vyjednávání výhodných smluv pro snížení nákladů. Prozkoumat alternativní dodavatele, abychom snížili závislost na jednom dodavateli.</p> |

PEST analýza

| | |
|---------------------------|--|
| Politické faktory | Platební systém bude podléhat různým regulacím a politikám uloženým vládou a regulačními orgány. Projektový tým bude muset mít aktuální informace o změnách zákonů a předpisů, aby zajistil jejich dodržování a vyhnul se sankcím. |
| Ekonomické faktory | Projekt platebního systému bude pravděpodobně ovlivněn ekonomickými podmínkami, jako je inflace, úrokové sazby a směnné kurzy. Tým bude muset pečlivě sledovat a řídit finanční rizika, aby byla zajištěna udržitelnost projektu. |
| Sociálně-kulturní faktory | Projekt platebního systému bude ovlivněn sociálními faktory, jako jsou měnící se preference spotřebitelů a demografické údaje. Tým bude muset držet krok s trendy v technologiích a platebních metodách, aby zůstal relevantní a vyhovoval potřebám zákazníků. |
| Technologické faktory | Úspěch projektu platebního systému bude do značné míry záviset na technologii a inovacích. Tým bude muset zůstat v obraze s nejnovějšími pokroky v platební technologii a investovat do výzkumu a vývoje, aby si udržel náskok před konkurencí. |

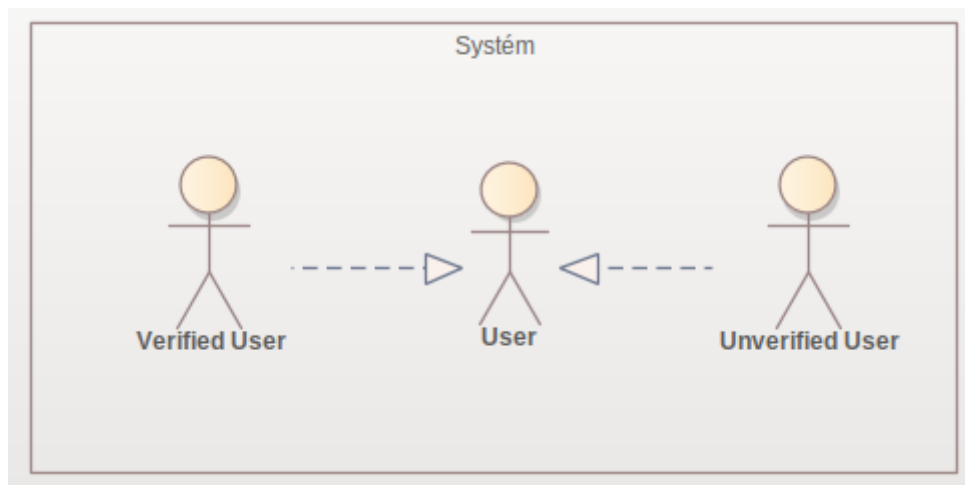
Funkční požadavky

- Systém umožní uživatelům iniciovat interní převody prostředků mezi jejich vlastními účty.
- Systém umožní uživatelům iniciovat externí převody peněžních prostředků jiným uživatelům nebo platebním systémům.
- Systém vygeneruje jedinečné potvrzovací kódy pro každou transakci, aby byla zajištěna bezpečnost a sledovatelnost.
- Systém poskytne uživatelům možnost smazat svůj účet a všechna související data.
- Systém musí uchovávat záznamy o minulých transakcích a umožnit uživatelům je prohlížet.
- Systém poskytne uživatelům informace o zůstatku každého účtu v reálném čase.
- Systém umožní uživatelům propojit se svým účtem více platebních metod.
- Systém bude poskytovat uživatelsky přívětivé rozhraní pro zahájení a správu transakcí.
- Systém zavede přísná bezpečnostní opatření k ochraně uživatelských dat a zabránění podvodným činnostem.
- Systém umožní uživatelům nastavit automatické opakující se transakce pro pravidelné platby nebo převody.

Nefunkční požadavky

- Zabezpečení: Systém by měl zajistit bezpečnost uživatelských dat a transakcí a zabránit neoprávněnému přístupu a podvodům.
- Spolehlivost: Systém by měl mít vysokou dobu provozuschopnosti a měl by být schopen zvládnout velké objemy transakcí bez selhání.
- Výkon: Systém by měl být rychlý a pohotový, s nízkou latencí a vysokou propustností.
- Škálovatelnost: Systém by měl být schopen škálovat horizontálně i vertikálně, aby zvládl rostoucí provoz a uživatelskou základnu.
- Použitelnost: Systém by měl být uživatelsky přívětivý a intuitivní, s jednoduchým a přehledným uživatelským rozhraním.
- Dostupnost: Systém by měl být přístupný uživatelům se zdravotním postižením a měl by podporovat více jazyků a měn.
- Kompatibilita: Systém by měl být kompatibilní s různými zařízeními, platformami a prohlížeči.
- Údržba: Systém by měl být snadno udržovatelný, aktualizovatelný a laditelný, s dobrou dokumentací a organizací kódu.
- Integrace: Systém by měl být schopen integrace s jinými platebními systémy a finančními službami, jako jsou banky a kreditní karty.
- Soulad: Systém by měl být v souladu s příslušnými zákony a předpisy, jako jsou zákony na ochranu údajů a proti praní špinavých peněz.

Seznam uživatelů



Obrázek č.2 – Uživatelé systému

System bude podporovat 2 typy uživatelů: Ověřený uživatel a neověřený uživatel. Oba uživatelé mají stejné možnosti (kromě toho, že neověřený uživatel má limit na všechny své transakce):

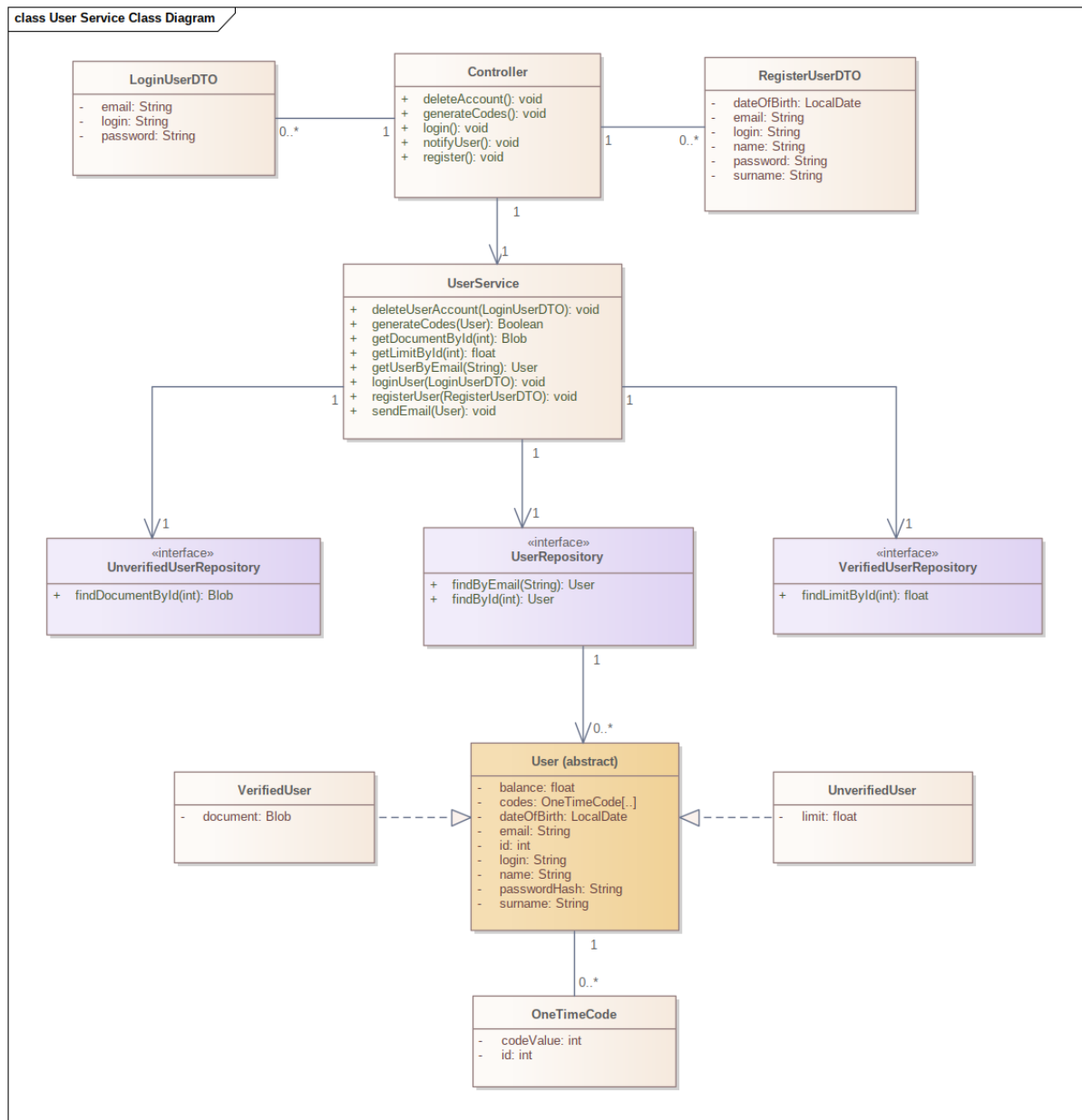
- Zobrazení a aktualizace osobních údajů (e-mail, heslo a login)
- Provádění interních transakcí
- Provádění externích transakcí
- Generování nových kódů pro potvrzení transakcí
- Odstranění účtu
- Zobrazení seznamu minulých transakcí

Případy užití

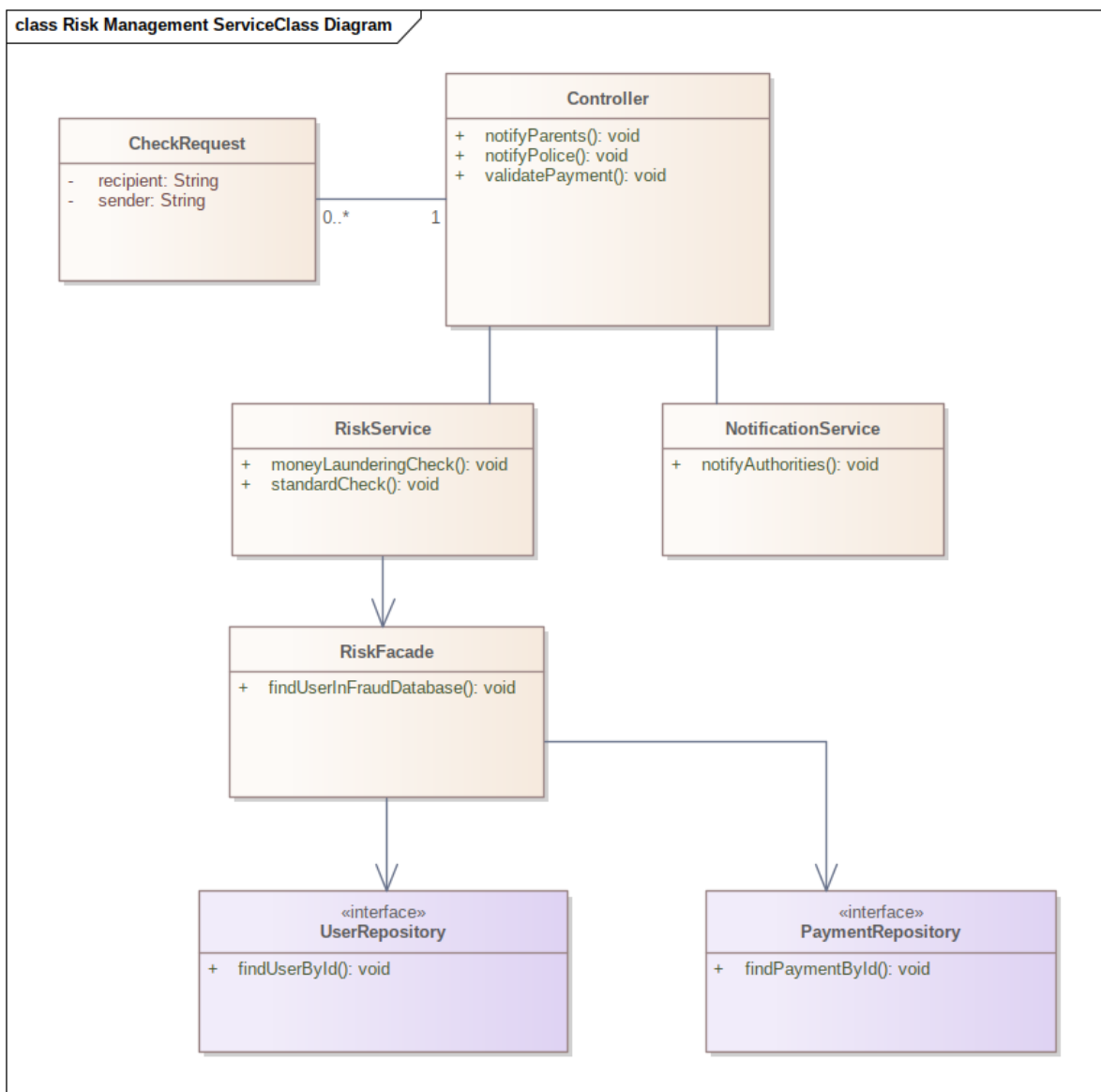


Obrázek č.3 – Use case diagram

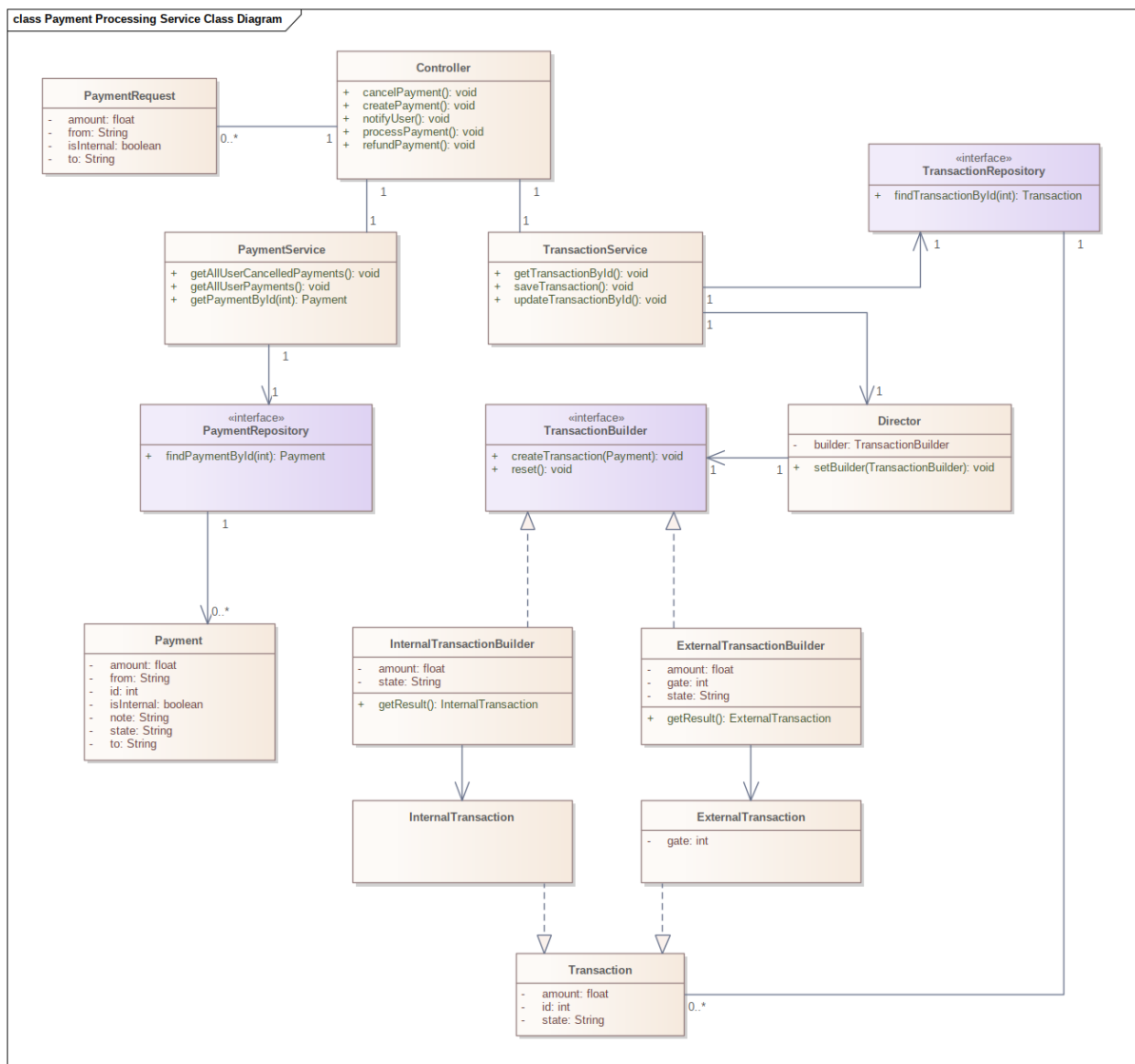
UML diagramy



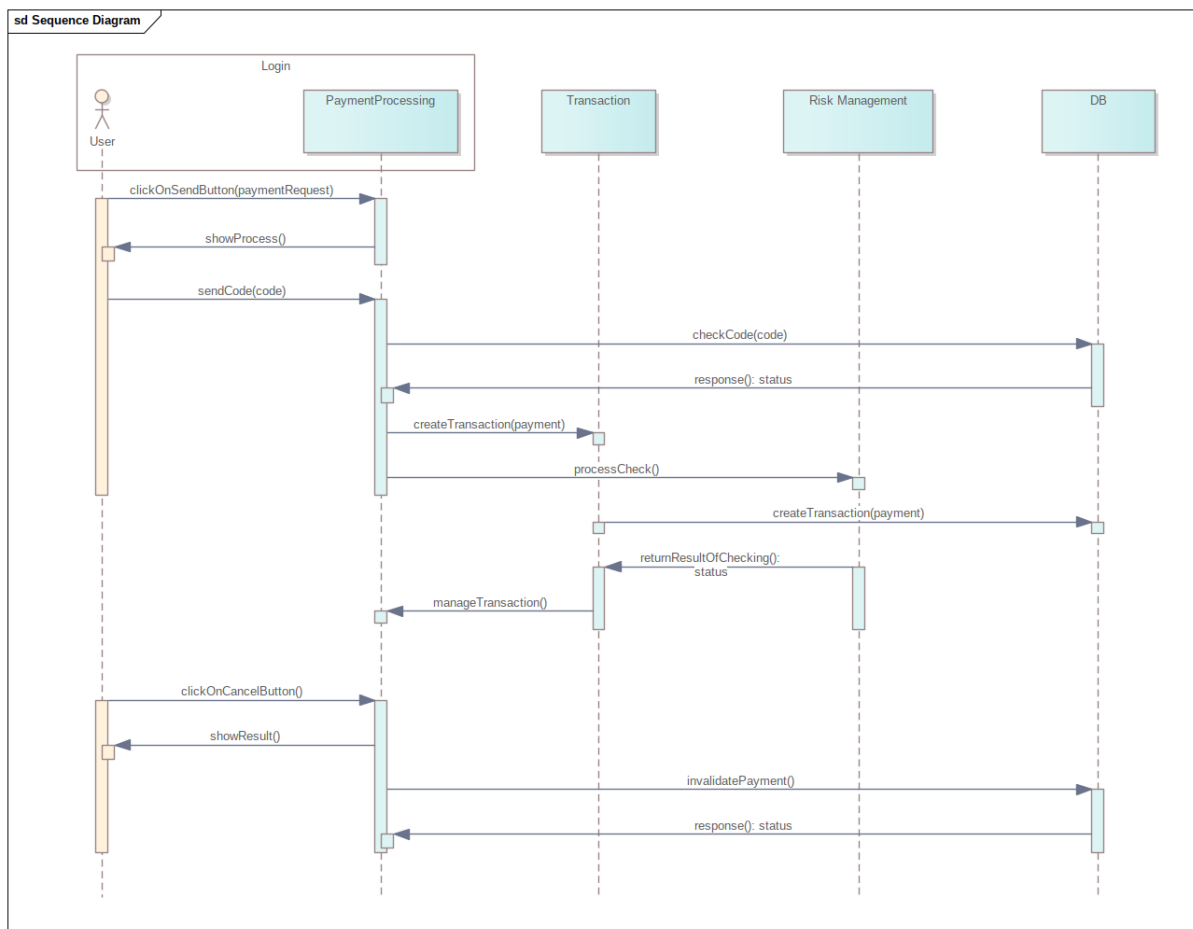
Obrázek č.4 – User Service class diagram



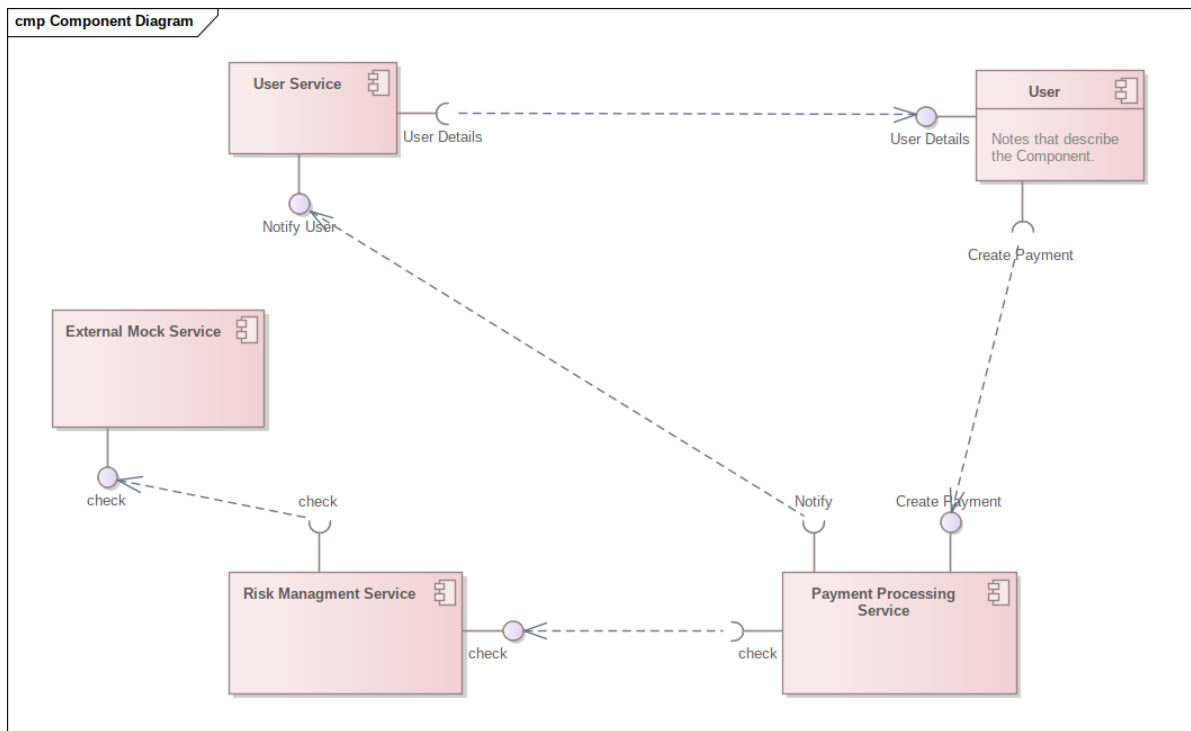
Obrázek č.5 – Risk Management Service class diagram



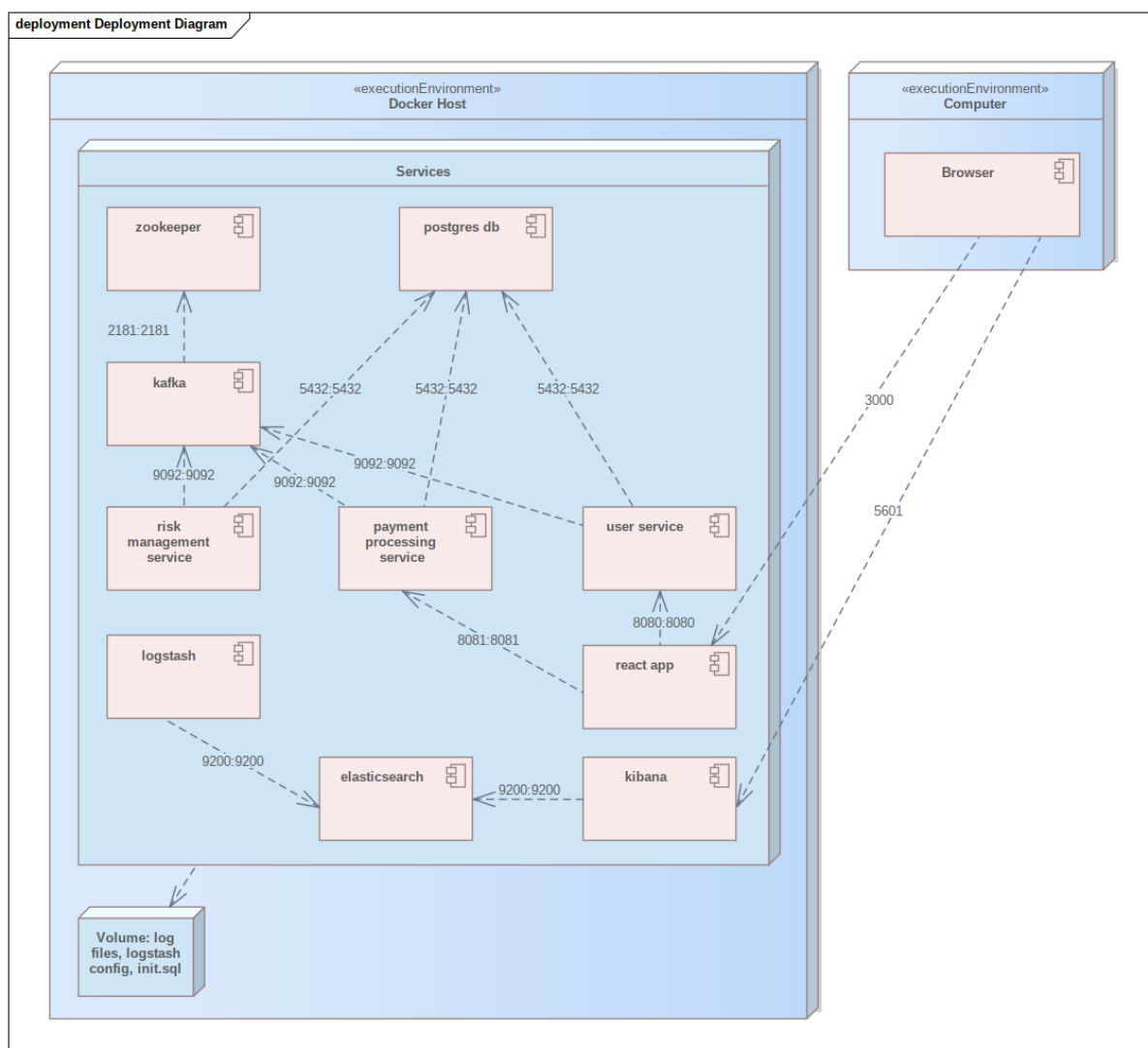
Obrázek č.6 – Payment Processing Service class diagram



Obrázek č.7 – Sekvenční diagram



Obrázek č.8 – Component diagram



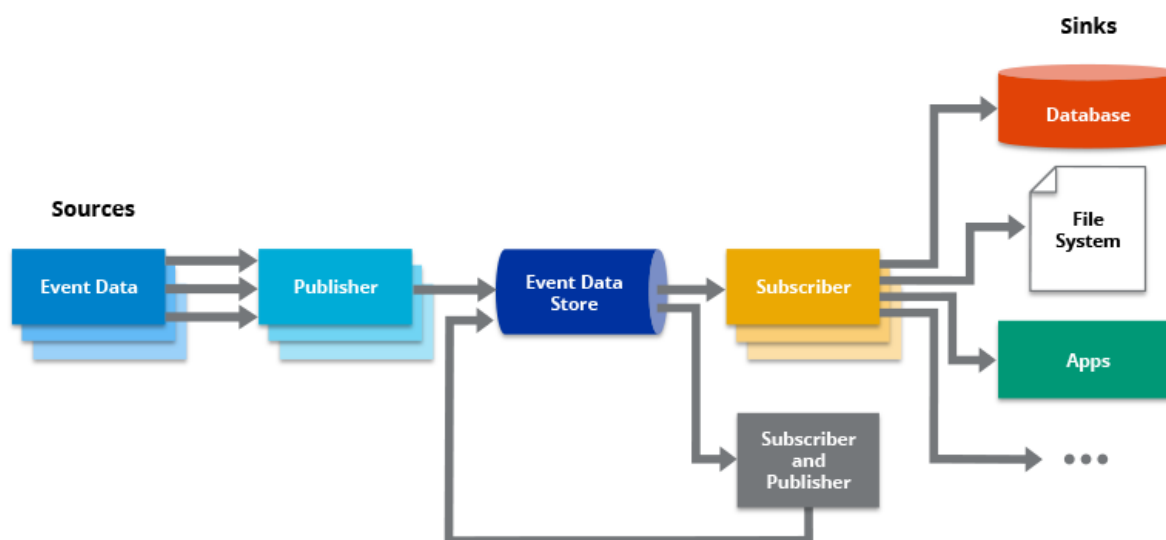
Obrázek č.9 – Deployment diagram

Výběr architektury

Jako architektonický vzor pro tento projekt byla vybrána Event Driven Architecture (EDA). EDA je návrhový vzor softwaru, který umožňuje produkci, detekci, spotřebu a reakci na události včasným a škálovatelným způsobem. Tento vzor je vhodný pro aplikace, které vyžadují zpracování událostí nebo datových toků v reálném čase, jako jsou platební systémy.

V tomto případě je EDA vhodná, protože umožňuje včasné a efektivní zpracování platebních událostí a transakcí. Když je například provedena platba, je vygenerována událost, která spustí sadu akcí, které mají být

provedeny, jako je aktualizace zůstatků na účtu, zasílání upozornění uživatelům a protokolování transakčních dat. EDA také podporuje oddělení různých komponent systému, což usnadňuje přidávání nebo odebírání funkcí podle potřeby, aniž by to ovlivnilo celý systém. Celkově použití EDA pomáhá zajistit, aby byl platební systém škálovatelný, spolehlivý a reagoval na události v reálném čase.



Obrázek č.10 - Event Driven Architecture

Rozbor a výběr alternativ návrhu řešení

Při vývoji platebního systému pro náš semestrální projekt jsme použili architekturu řízenou událostmi (EDA), abychom zajistili škálovatelné a robustní řešení. Během fáze návrhu jsme se však setkali s různými alternativami návrhu, které vyžadovaly pečlivou analýzu a vyhodnocení. V této kapitole probereme různé zvažované alternativy návrhu řešení, analyzujeme jejich silné a slabé stránky a zdůvodníme výběr architektury řízené událostmi pro náš platební systém.

Monolitická architektura:

Jednou z původních alternativ návrhu byla monolitická architektura. V tomto přístupu by byl platební systém vyvinut jako jediná, těsně provázaná aplikace. Hlavní výhodou monolitické architektury je její jednoduchost, protože všechny komponenty jsou nasazeny a spravovány společně. Tento přístup však přináší problémy, pokud jde o

škálovatelnost, udržitelnost a odolnost proti chybám. Kromě toho mohou jakékoli změny nebo aktualizace konkrétního modulu potenciálně ovlivnit celý systém, což vede k nedostatečné flexibilitě. Proto jsme došli k závěru, že monolitická architektura by pro náš platební systém nebyla vhodná.

Architektura orientovaná na služby (SOA):

Další hodnocenou alternativou návrhu byla architektura orientovaná na služby (SOA). SOA obhájí rozklad systému na volně propojené a nezávislé služby, komunikující prostřednictvím standardizovaných rozhraní. Tento přístup poskytuje lepší modularitu a umožňuje škálovatelnost a flexibilitu. Správa složitých interakcí a závislostí mezi službami však může být náročná. Povaha požadavku-odezva SOA navíc může zavést latenci a vytvořit potenciální jediný bod selhání. Po pečlivém zvážení jsme usoudili, že architektura SOA není pro náš platební systém nejvhodnější vzhledem k potenciálním problémům s výkonem a škálovatelností.

Po vyhodnocení různých alternativ návrhu řešení jsme zjistili, že našim požadavkům na platební systém nejlépe vyhovuje architektura řízená událostmi (EDA). EDA nabízí škálovatelnost, flexibilitu, volné propojení a odolnost, které jsou klíčové pro vybudování robustního a adaptabilního platebního systému.

WBS – rozdělení projektu na dílčí procesy

Abychom mohli efektivně plánovat a řídit vývoj našeho projektu platebního systému, je nezbytné jej rozdělit na menší, zvládnutelné dílčí procesy. Tato kapitola se zaměřuje na vytvoření struktury rozpisu práce (WBS), která je hierarchickou reprezentací výstupů, úkolů a dílčích procesů projektu. WBS slouží jako základ pro organizaci a kontrolu projektových aktivit. V této kapitole nastíníme WBS pro náš projekt platebního systému a rozdělíme jej na logické dílčí procesy.

WBS bude vyvinut s použitím přístupu shora dolů, počínaje hlavními výstupy a rozdělit je na menší, zvládnutelné úkoly a dílčí procesy. WBS bude organizována v hierarchické struktuře, přičemž každá úroveň představuje jinou úroveň podrobností.

Níže je uveden příklad WBS na vysoké úrovni pro náš projekt platebního systému:

Úroveň 1: Projekt

- Vyvinout platební systém

Úroveň 2: Dílčí procesy

- 2.1 Shromažďování a analýza požadavků
- 2.2 Návrh systému
- 2.3 Správa uživatelů
- 2.4 Zpracování plateb
- 2.5 Historie transakcí
- 2.6 Testování a zajištění kvality
- 2.7 Správa nasazení a vydání
- 2.8 Dokumentace a uživatelské příručky
- 2.9 Řízení a koordinace projektů

Úroveň 3: Úkoly a dílčí úkoly

- 2.1.1 Definovat funkční a nefunkční požadavky
- 2.1.2 Provádějte rozhovory a workshopy se zúčastněnými stranami
- 2.1.3 Analyzovat a stanovovat priority požadavků
- 2.1.4 Vytvořit dokument specifikace požadavků
- 2.2.1 Architektonický návrh
- 2.2.2 Návrh databáze
- 2.2.3 Návrh uživatelského rozhraní
- 2.2.4 Návrh integrace
- 2.3.1 Modul registrace uživatele
- 2.3.2 Autentizace a autorizace uživatele
- 2.3.3 Správa uživatelských profilů
- 2.4.1 Protokolování transakcí
- 2.4.2 Načítání historie transakcí
- 2.4.3 Vykazování a analýzy
- 2.5.1 Plánování testů a vývoj strategie
- 2.5.2 Testování jednotky
- 2.5.3 Testování integrace
- 2.5.4 Testování systému
- 2.5.5 Testování výkonu

- 2.5.6 Bezpečnostní testování
- 2.6.1 Plánování nasazení
- 2.6.2 Správa konfigurace
- 2.6.3 Koordinace uvolnění
- 2.7.1 Uživatelská dokumentace
- 2.7.2 Administrátorská dokumentace
- 2.7.3 Uživatelské příručky a výukové programy
- 2.8.1 Plánování a rozvrhování projektu

Zdroje

Při vývoji projektu platebního systému jsme spolupracovali, převzali jsme více rolí a odpovědností. Tato kapitola se zaměřuje na zdroje využívané během projektu, včetně softwarových nástrojů, infrastruktury a technologií. Konkrétně mezi použité prostředky patří Docker pro kontejnerizaci, infrastruktura Azure pro nasazení a různé softwarové technologie, jako je Java, React, Kafka, Elasticsearch a PostgreSQL.

Jako tým pracující na projektu jsme mezi sebou alokovali zdroje na základě našich individuálních dovedností a odborných znalostí. Každý člen týmu převzal specifické role a odpovědnosti, včetně vývoje softwaru, návrhu systému, testování a projektového řízení. Tento přístup založený na spolupráci nám umožnil efektivně využívat dostupné zdroje a efektivně rozdělit pracovní zátěž.

Použili jsme Docker jako zdroj pro usnadnění kontejnerizace komponent aplikace. Využitím Dockeru jsme zabalili různé služby, knihovny a závislosti platebního systému do izolovaných kontejnerů. Tento přístup zjednodušil nasazení, zajistil konzistentní běhová prostředí v různých fázích vývoje a zlepšil škálovatelnost a přenositelnost aplikace.

Pro nasazení jsme využili infrastrukturu Azure a využili její cloudové služby a možnosti platformy. Azure poskytl spolehlivé a škálovatelné prostředí pro hostování aplikace platebního systému, což nám umožnilo soustředit se spíše na vývoj než na správu hardwaru nebo infrastruktury. Funkce Azure, jako jsou virtuální počítače, kontejnerové služby a bezserverové výpočty, podporovaly požadavky projektu na nasazení a škálování.

Normy a standardy

Dodržovali jsme následující normy a standardy pro používané programovací jazyky:

Java 17: Dodrželi jsme konvence kódování uvedené v Konvencích kódu Java, abychom zajistili konzistenci a čitelnost našeho kódu. Pro třídy, metody, proměnné a balíčky byly použity správné konvence pojmenování, aby se zlepšila srozumitelnost kódu. Použili jsme také funkce jazyka Java 17 a osvědčené postupy pro lepší efektivitu kódu a výkon.

Rámcové normy a standardy: Pro použité rámce jsme dodržovali následující normy a standardy: Spring Boot 3: Dodrželi jsme konvence Spring Framework, včetně správné struktury balíčků, pojmenování tříd a použití anotací. Využili jsme funkce automatické konfigurace a správy závislostí Spring Boot ke zjednodušení nastavení projektu a omezení ruční konfigurace. Kromě toho jsme přijali principy inverze řízení (IoC) a vkládání závislostí (DI), abychom podpořili volně spojený a modulární kód.

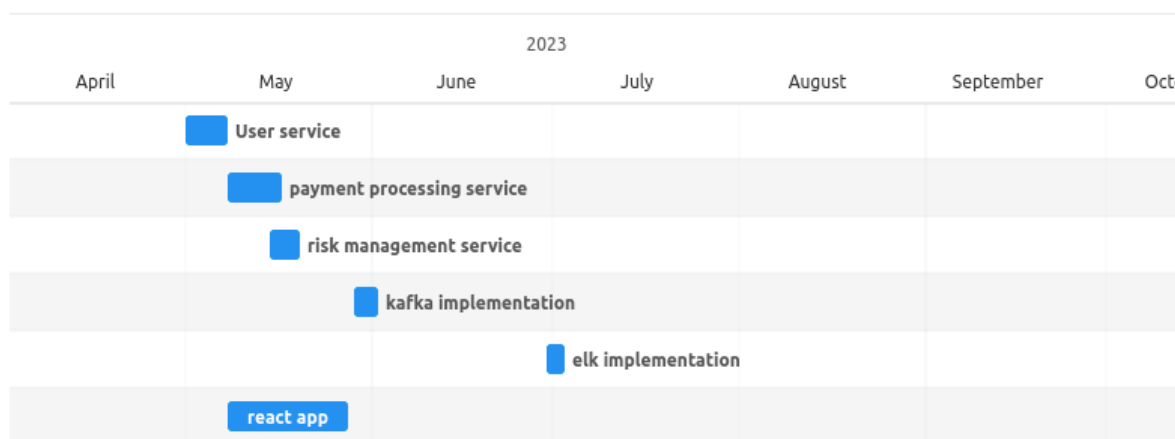
React 18.2.0 + Vite: Dodržovali jsme rámcové konvence a osvědčené postupy React, včetně vývoje založeného na komponentách a řízení stavu. S využitím funkčních komponent, háčků a kontextového API jsme vytvořili opakovaně použitelné a škálovatelné komponenty uživatelského rozhraní. Dodrželi jsme syntaxi JSX a správné odsazení, abychom zlepšili čitelnost kódu. Kromě toho jsme využili virtuální DOM společnosti React a algoritmus odsouhlasení pro efektivní vykreslování a optimální výkon.

Normy a standardy pro vývoj webu: Dodržovali jsme následující normy a standardy pro vývoj webových aplikací: HTML 5: Použili jsme postupy sémantického značkování HTML 5 ke zlepšení struktury a přístupnosti webových stránek. CSS: Dodržovali jsme osvědčené postupy CSS, včetně použití externích šablon stylů, správných konvencí pojmenování tříd a vyhýbání se inline stylům.

Kontrola verzí a spolupráce: Ke správě zdrojového kódu projektu a efektivní spolupráci jsme použili systém správy verzí, jako je Git. Dodržování osvědčených postupů Git, včetně správného větvení, zpráv

odevzdání a pravidelných kontrol kódu, zajistilo kvalitu kódu a efektivní správu správy verzí.

Harmonogram GANTT



Obrázek č.11 - GANTT harmonogram

Analýza rizik FMEA

Při vývoji projektu platebního systému je provedení analýzy rizik klíčové pro identifikaci a zmírnění potenciálních selhání. Analýza režimů a důsledků selhání (FMEA) je běžně používaná technika pro stanovení priority režimů selhání a jejich účinků. Tato kapitola poskytuje stručný přehled analýzy rizik FMEA provedené pro projekt platebního systému, zdůrazňuje související kroky a identifikovaná rizika.

Přehled procesu FMEA:

Proces FMEA zahrnoval následující kroky:

- Sestavení týmu: Byl sestaven mezifunkční tým pro analýzu platebního systému.
- Porozumění systému a procesů: Tým získal komplexní porozumění komponentám a procesům systému.
- Identifikace režimu selhání: Byly probrány a zdokumentovány možné režimy selhání.
- Posouzení závažnosti: Byl vyhodnocen dopad každého režimu selhání na funkčnost, integritu dat a uživatelskou zkušenost.

- Analýza příčin: Byly zkoumány hlavní příčiny každého režimu selhání.
- Posouzení rizik: Byla posouzena pravděpodobnost výskytu a účinnost stávajících kontrol.
- Výpočet čísla priority rizika (RPN): Hodnoty RPN byly vypočteny na základě závažnosti, výskytu a hodnocení detekce.
- Zmírňování rizik: Rizika byla upřednostněna a byly vyvinuty strategie zmírňování. Identifikovaná rizika: Analýza FMEA identifikovala několik rizik specifických pro projekt platebního systému, včetně porušení dat, selhání zpracování plateb, snížení výkonu, problémů s integrací a ztráty dat.

Strategie zmírnění rizik: K řešení identifikovaných rizik tým implementoval strategie, jako je posílení bezpečnostních opatření, zavedení mechanismů redundance a převzetí služeb při selhání, optimalizace výkonu systému, provádění důkladného testování a zajištění kvality a implementace postupů zálohování dat a obnovy po havárii.

Znovupoužitelnost

Při vývoji projektu platebního systému může podpora opětovné použitelnosti kódu a komponent výrazně zvýšit produktivitu, udržitelnost a škálovatelnost. Tato kapitola se zaměřuje na důležitost opětovné použitelnosti a strategií použitých k maximalizaci opětovného použití kódu a komponent v rámci projektu. Pro maximalizaci opětovné použitelnosti kódu v rámci projektu platebního systému byly použity následující strategie:

- Modulární design: Rozdělení systému na modulární komponenty podporuje zapouzdření a oddělení zájmů. Každý modul lze nezávisle znovu použít a snadno integrovat do jiných částí systému. Knihovny a rámce:
- Využití knihoven a rámců třetích stran umožňuje využít předem vytvořené a otestované funkce, což snižuje čas a úsilí na vývoj.
- Návrhové vzory: Implementace známých návrhových vzorů, jako je Singleton, Factory nebo Observer, zvyšuje znovupoužitelnost kódu

tím, že poskytuje osvědčená řešení běžných návrhových problémů. Standardizované postupy kódování:

- Dodržování konzistentních postupů kódování, jako je dodržování konvencí pojmenování, použití správné dokumentace a používání jasných a stručných stylů kódování, zlepšuje čitelnost a srozumitelnost kódu a usnadňuje opětovné použití kódu.

Metriky

V projektu platebního systému byly použity následující klíčové metriky:

- Rychlost vývoje: Tato metrika měří rychlost a efektivitu procesu vývoje, včetně počtu implementovaných funkcí, vyřešených chyb a úkolů dokončených v určitém časovém rámci.
- Kvalita kódu: Metriky kvality kódu posuzují udržitelnost, čitelnost a dodržování standardů kódování ve projektu. Mezi běžné metriky kvality kódu patří složitost kódu, pokrytí kódem a zjištění analýzy statického kódu.
- Zapojení uživatelů: Metriky zapojení uživatelů sledují interakce uživatelů v rámci platebního systému, jako jsou aktivní uživatelé, trvání návštěv, konverzní poměry a zpětná vazba od uživatelů.

Plán podpory

Cílem plánu podpory je dosáhnout následujících cílů:

- Řešení problémů: Okamžitě řešit jakékoli technické nebo funkční problémy, se kterými se setkají uživatelé nebo správci systému.
- Údržba a aktualizace: Pravidelně používat potřebné aktualizace, záplaty a opravy chyb, abychom zajistili stabilitu a bezpečnost systému.
- Komunikační kanály: Vytvořit jasné komunikační kanály, jako je vyhrazený e-mail podpory, systém prodeje vstupenek nebo znalostní báze, abychom usnadnili dotazy uživatelů a zpětnou vazbu.

Vyhodnocení

Při práci na projektu jsem se naučil spoustu nových technologií.