Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Interaktivní webová platforma pro rezervaci vstupenek na studentské akce

2025

Tuan Anh Nguyen

Vedoucí práce:
Mgr. Jiří Zacpal, Ph.D.

Studijní program: Informační technologie, kombinovaná forma

Bibliografické údaje

Autor:	Tuan Anh Nguyen
Název práce:	Interaktivní webová platforma pro rezervaci vstupenek na studentské akce
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2025
Studijní program:	Informační technologie, kombinovaná forma
Vedoucí práce:	Mgr. Jiří Zacpal, Ph.D.
Počet stran:	27
Přílohy:	elektronická data v úložišti katedry informatiky
Jazyk práce:	český

Bibliographic info

Author:	Tuan Anh Nguyen
Title:	Interactive web platform for ticket reservation on student events
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2025
Study program:	Information Technologies, combined form
Supervisor:	Mgr. Jiří Zacpal, Ph.D.
Page count:	27
Supplements:	electronic data in the storage of department of computer science
Thesis language:	Czech

Anotace

Bakalářská práce se zabývá návrhem a implementací webové aplikace pro rezervaci vstupenek na studentské akce. Aplikace umožňuje organizátorům vytvářet a spravovat akce, uživatelům rezervovat a platit vstupenky online. Systém využívá moderní technologie jako React, Node.js, PostgreSQL a integruje platební bránu Stripe. Součástí je kompletní systém správy rezervací, reklamací, automatických emailových notifikací a generování PDF vstupenek s QR kódy. Práce zahrnuje i administrační rozhraní pro správu uživatelů a akcí.

Synopsis

This bachelor thesis focuses on the design and implementation of a web application for ticket reservation for student events. The application allows organizers to create and manage events, and users to reserve and pay for tickets online. The system uses modern technologies such as React, Node.js, PostgreSQL and integrates the Stripe payment gateway. It includes a complete reservation management system, complaints handling, automatic email notifications and PDF ticket generation with QR codes. The work also includes an administrative interface for user and event management.

Klíčová slova: webová aplikace; rezervační systém; vstupenky; React; Node.js; TypeScript; Stripe; PostgreSQL; studentské akce

Keywords: web application; reservation system; tickets; React; Node.js; TypeScript; Stripe; PostgreSQL; student events

Děkuji vedoucímu práce Mgr. Jiřímu Zacpalovi, Ph.D. za odborné vedení a cenné rady při tvorbě této práce. Dále děkuji své rodině za podporu během celého studia.

Odevzdáním tohoto textu jeho autor/ka místopřísežně prohlašuje, že celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

Obsah

1 Průzkum existujících řešení **8**
- 1.1 Popis jednotlivých řešení . . . 8
  - 1.1.1 Eventbrite . . . 8
  - 1.1.2 Ticketmaster . . . 8
  - 1.1.3 Meetup . . . 8
  - 1.1.4 Eventzilla . . . 8
- 1.2 Hodnocení kladů a záporů . . . 9
- 1.3 Vyhodnocení přínosů . . . 9
- 1.4 Přínosy vlastní práce . . . 9

2 Specifikace řešení **10**
- 2.1 Uživatelské role . . . 10
- 2.2 Správa událostí . . . 10
- 2.3 Rezervace . . . 10
- 2.4 Reklamace a vrácení vstupenek . . . 11
- 2.5 Popis použitých technologií . . . 11
  - 2.5.1 Backend . . . 11
  - 2.5.2 Databáze . . . 12
  - 2.5.3 Frontend . . . 12
  - 2.5.4 Autentizace a autorizace . . . 13
  - 2.5.5 Platby . . . 13
  - 2.5.6 Hosting a nasazení . . . 13
- 2.6 Implementované pokročilé funkce . . . 14
  - 2.6.1 Email notifikace . . . 14
  - 2.6.2 PDF vstupenky s QR kódy . . . 14
  - 2.6.3 Admin dashboard . . . 15
  - 2.6.4 Testování . . . 15
- 2.7 Relační model databáze . . . 15
- 2.8 Diagram tříd . . . 20

3 Programátorská příručka **21**
- 3.1 Klíčové algoritmy a implementace . . . 21
  - 3.1.1 Autentizace pomocí JWT . . . 21
  - 3.1.2 Atomické transakce při rezervaci . . . 21
  - 3.1.3 Zpracování Stripe webhooků . . . 21
  - 3.1.4 Proces refundace . . . 22
  - 3.1.5 Generování PDF vstupenek . . . 22
- 3.2 Možnosti rozšíření aplikace . . . 22
  - 3.2.1 Přidání nové databázové entity . . . 22
  - 3.2.2 Přidání nového typu emailové notifikace . . . 23
  - 3.2.3 Integrace nové platební metody . . . 23
  - 3.2.4 Přidání nové uživatelské role . . . 23
- 3.3 Testování . . . 23

3.4 Deployment 23

4 Uživatelská příručka **24**
4.1 První kroky 24
4.1.1 Registrace 24
4.1.2 Přihlášení 24
4.2 Prohlížení a rezervace akcí 24
4.2.1 Prohlížení akcí 24
4.2.2 Detail a rezervace 24
4.3 Správa rezervací 25
4.4 Reklamace 25
4.5 Funkce pro organizátory 25
4.5.1 Vytvoření akce 25
4.5.2 Správa akcí 25
4.6 Funkce pro administrátory 26
4.6.1 Správa uživatelů 26
4.6.2 Správa akcí a reklamací 26
4.7 Diagram případu použití 26

Seznam obrázků

1 Sekvenční diagram procesu rezervace a platby 16
2 Stavový diagram rezervace 17
3 Třívrstvá architektura aplikace s externími službami 17
4 Sekvenční diagram procesu autentizace s JWT tokeny 18
5 Relační model databáze - 5 tabulek s vazbami 19
6 Diagram tříd - Backend struktura (MVC pattern) 20
7 Diagram případu použití - Rezervační systém 27

Seznam tabulek

1 Průzkum existujících řešení

Pro zkoumání budou vybrány následující platformy:

- Eventbrite
- Ticketmaster
- Meetup
- Eventzilla

1.1 Popis jednotlivých řešení

1.1.1 Eventbrite

je platforma využívající technologie Ruby on Rails, React a PostgreSQL. Mezi hlavní funkce patří vytváření, modifikace a rušení akcí, rezervace a nákup vstupenek online, podpora různých typů vstupenek (volný vstup, placené vstupenky), integrovaný platební systém a možnost refundací. Výhodou Eventbrite je jeho vysoký výkon a škálovatelnost, stejně jako podpora různých platebních bran. Na druhou stranu se jedná o řešení s vyššími náklady na implementaci a údržbu a složitým kódem

1.1.2 Ticketmaster

je postaven na technologiích Java, Spring Boot, Angular a MySQL. Poskytuje funkce správy a propagace akcí, rezervace a prodeje vstupenek, podporu mobilních vstupenek a systém pro správu vstupenek, včetně reklamací a refundací. Ticketmaster je robustní a zabezpečená platforma, která podporuje velký počet uživatelů a transakcí. Nevýhodou jsou vyšší složitost vývoje a údržby a vysoké náklady na provoz.

1.1.3 Meetup

využívá technologie Node.js, Express.js, React a MongoDB. Umožňuje organizaci a správu událostí, potvrzení účasti (RSVP) a podporu plateb za akce. Mezi jeho výhody patří flexibilní a rychlý vývoj díky použitým technologiím a dobrá podpora komunitních funkcí. Omezení spočívají ve škálovatelnosti pro velmi velké akce a méně pokročilých funkcích pro správu plateb.

1.1.4 Eventzilla

je platforma postavená na PHP, Laravel, Vue.js a PostgreSQL. Nabízí funkce pro vytváření a správu událostí, rezervaci a prodej vstupenek, podporu různých typů vstupenek a integraci s různými platebními bránami. Výhodou Eventzilla je snadné přizpůsobení a rozšíření funkcí, stejně jako dobrá podpora různých

platebních systémů. Nevýhodou je méně známá značka a omezené analytické nástroje.

1.2 Hodnocení kladů a záporů

Pokud se podíváme na hodnocení kladů a záporů jednotlivých řešení, Eventbrite je vysoce výkonné a škálovatelné, ale s vyššími náklady a komplexností. Ticketmaster je robustní a zabezpečené, ale drahé a složité. Meetup nabízí flexibilní a rychlý vývoj, ale je omezené pro velké akce. Eventzilla je přizpůsobitelné a cenově dostupné, ale méně známé a s omezenými analytickými nástroji.

1.3 Vyhodnocení přínosů

Co se týče přínosů, Eventbrite inspiruje k vytvoření výkonného a škálovatelného řešení. Ticketmaster ukazuje, jak vybudovat robustní a zabezpečenou platformu. Meetup klade důraz na komunitní funkce a rychlý vývoj. Eventzilla nabízí přístup k přizpůsobitelnému a cenově dostupnému řešení.

1.4 Přínosy vlastní práce

Podpora studentských aktivit bude posílena vytvořením platformy pro rezervaci vstupenek, což usnadní organizaci a propagaci studentských akcí. To přispěje k aktivnímu studentskému životu a komunitě. Zjednodušení správy akcí prostřednictvím automatizace a zjednodušení procesů spojených s organizací akcí ulehčí práci studentským organizátorům, kteří se tak mohou více soustředit na kreativní a obsahovou část akcí.

Zvýšení účasti na akcích je dalším přínosem, neboť snadná dostupnost a rezervace vstupenek může vést ke zvýšenému zapojení studentů do komunitního života. Bezpečnost a pohodlí budou zajištěny implementací moderních platebních systémů a ochranou osobních údajů podle GDPR, což zvýší důvěru uživatelů v používání platformy.

Rozvoj digitálních dovedností u studentů zapojených do vývoje a správy platformy je další významný přínos. Studenti získají praktické zkušenosti s moderními technologiemi a vývojem webových aplikací, což je cenné pro jejich budoucí profesní kariéru. Ekologický přínos se projeví digitalizací vstupenek a snížením potřeby fyzických vstupenek, což přispěje k ochraně životního prostředí.

Zahájení činnosti studentské organizace zaměřené na organizaci a propagaci studentských akcí může být dalším přínosem této práce. Tím se obohatí univerzitní komunita a nabídnou se nové příležitosti pro zapojení studentů.

2 Specifikace řešení

2.1 Uživatelské role

Platforma bude disponovat několika uživatelskými rolemi, z nichž každá bude mít specifické přístupy a oprávnění.

Administrátor bude mít plný přístup ke všem funkcím systému, což mu umožní kontrolovat a spravovat všechny aspekty platformy.

Organizátor bude mít přístup k vytváření a správě vlastních akcí, což zahrnuje možnosti přidávat nové události, upravovat stávající a rušit je podle potřeby.

Běžný **uživatel** bude mít přístup k prohlížení nabízených akcí a rezervaci vstupenek, čímž se usnadní zapojení do studentských aktivit.

2.2 Správa událostí

V rámci správy událostí může organizátor přidávat nové akce vyplněním detailního formuláře, který obsahuje všechny potřebné specifikace události - název, popis, datum a čas (startDate, endDate), místo konání, kategorii, cenu vstupenky a celkový počet dostupných vstupenek.

Po zveřejnění má organizátor možnost upravovat údaje akce, měnit její stav (DRAFT, PUBLISHED, CANCELLED, COMPLETED) a sledovat statistiky - počet prodaných vstupenek, celkový příjem a zbývající kapacitu.

V případě potřeby zrušení události (stav CANCELLED) systém automaticky informuje všechny registrované účastníky emailovou notifikací a v případě placených akcí spustí proces refundace. Aktuálně dostupné vstupenky se automaticky aktualizují při každé nové rezervaci.

2.3 Rezervace

Uživatelé mohou prohlížet dostupné akce prostřednictvím přehledného seznamu s možností filtrování podle kategorií (Hudba, Divadlo, Film, Sport, Technologie, Vzdělávání, Jiné) a vyhledávání podle názvu akce. Každá akce zobrazuje základní informace včetně počtu zbývajících vstupenek a ceny.

Detail akce poskytuje kompletní informace o události včetně popisu, data a času, místa konání a statistik. Uživatel si zde může vybrat počet vstupenek (max. 10 na jednu rezervaci) a přejít k platbě.

Při placených akcích je implementován Stripe Payment Intent flow s bezpečným platebním formulářem (Stripe Elements). Po úspěšné platbě potvrzené přes Stripe webhook je rezervace automaticky označena jako PAID a uživatel obdrží emailové potvrzení s rezervačním kódem.

Pro akce zdarma ($\text{ticketPrice} = 0$) je rezervace okamžitě potvrzena bez platebního procesu. Uživatel může své rezervace spravovat na stránce "Moje rezervace", kde může rezervaci zrušit (refundace pro placené rezervace) nebo stáhnout PDF vstupenku s QR kódem.

Obrázek 1 znázorňuje kompletní flow rezervace vstupenek. Proces začíná výběrem akce, pokračuje vytvořením rezervace v databázové transakci (s kontrolou dostupnosti a atomickým snížením počtu vstupenek), následuje platba přes Stripe Payment Intent pro placené akce nebo okamžité potvrzení pro akce zdarma, a končí webhook potvrzením platby a odesláním emailové notifikace.

Obrázek 2 ukazuje přechody mezi stavy rezervace. Rezervace začíná ve stavu PENDING (čeká na platbu), přechází do stavu PAID po úspěšné platbě, a může končit ve stavu CANCELLED (zrušeno bez platby) nebo REFUNDED (zrušeno s vrácením peněz). Refundace může nastat z důvodu schválené reklamace, zrušení uživatelem nebo zrušení celé akce organizátorem.

2.4 Reklamace a vrácení vstupenek

Systém reklamací umožňuje uživatelům podat reklamaci na rezervovanou akci prostřednictvím formuláře, který vyžaduje popis problému. Reklamace prochází těmito stavy:

- **SUBMITTED** - Nově podaná reklamace čekající na posouzení
- **IN_REVIEW** - Reklamace je v procesu posuzování administrátorem
- **RESOLVED** - Reklamace byla schválena, proběhla refundace
- **REJECTED** - Reklamace byla zamítnuta

Admin může v administračním rozhraní () prohlížet všechny reklamace, filtrovat podle statusu a reagovat na ně. Při vyřízení reklamace admin vyplní odpověď a volitelně zaškrtne checkbox pro automatickou refundaci.

Pokud je reklamace schválena s refundací, systém automaticky:

1. Provede Stripe refund (pouze pro placené akce)
2. Aktualizuje stav rezervace na CANCELLED
3. Zvýší počet dostupných vstupenek na akci
4. Odešle emailovou notifikaci uživateli s informací o refundaci

Pro akce zdarma se provede pouze zrušení rezervace bez Stripe refundu. Všechny operace jsou atomické a zajištěné databázovou transakcí.

2.5 Popis použitých technologií

2.5.1 Backend

Pro backend je použit **Node.js** s **TypeScript**, který se vyznačuje rychlým a efektivním asynchronním zpracováním. Tato vlastnost je klíčová pro aplikace, které musí zvládat vysokou zátěž na I/O operace, jako jsou rezervace a platby.

V kombinaci s Node.js je použit **Express.js**, což je jednoduchý a minimalistický framework pro RESTful API. Pro práci s databází slouží **Prisma ORM**, který poskytuje type-safe přístup k datům a automatickou generaci TypeScript typů.

Implementovány jsou i další knihovny: **Nodemailer** pro odesílání emailových notifikací, **PDFKit** pro generování PDF vstupenek, **QRCode** pro vytváření QR kódů na vstupenky a **bcryptjs** pro bezpečné hashování hesel.

Obrázek 3 znázorňuje kompletní architekturu aplikace. Presentation Layer (prezentační vrstva) obsahuje React frontend s komponentami a Stripe Elements pro platební formulář. API Layer (aplikační vrstva) tvoří Express.js backend se službami a middleware. Data Layer (datová vrstva) představuje PostgreSQL databázi. Externí služby zahrnují Stripe API pro platby a Email Service pro notifikace.

2.5.2 Databáze

Pro ukládání dat je použita **PostgreSQL**, což je spolehlivá a výkonná relační databáze. Podporuje pokročilé funkce jako je transakční zpracování, což je klíčové pro zajištění integrity dat při rezervacích a platbách.

Pro práci s databází slouží **Prisma ORM**, které poskytuje:

- Type-safe přístup k datům s automatickým generováním TypeScript typů
- Deklarativní databázové schéma
- Automatické migrace databáze
- Prisma Studio pro vizuální správu dat

Databáze obsahuje 7 hlavních modelů: User, Event, Reservation, Payment, Complaint, Ticket a RefreshToken.

2.5.3 Frontend

Na frontendu je použit **React 18** s **TypeScript**, což je vysoce výkonný a flexibilní frontendový framework. React umožňuje vytváření dynamických uživatelských rozhraní a díky komponentovému přístupu usnadňuje správu a opakované použití kódu.

Jako build tool slouží **Vite**, který zajišťuje rychlý development server a optimalizovaný production build. Pro routing je použit **React Router v6** s lazy loading pro optimalizaci výkonu.

Stav aplikace je řízen pomocí **Zustand** (lightweight state management) a **Axios** pro HTTP komunikaci s backendem. Formuláře využívají **React Hook Form** s **Zod** validací pro type-safe validaci dat.

Pro stylování je využit **Tailwind CSS**, což je utility-first CSS framework umožňující rychlé a efektivní stylování s plnou responzivitou. Pro ikony slouží **Lucide React** a pro notifikace **React Hot Toast**.

2.5.4 Autentizace a autorizace

Pro autentizaci a autorizaci uživatelů je implementován systém založený na **JWT** (JSON Web Tokens). Systém využívá dva typy tokenů:

- **Access Token** - krátkodobý (15 minut), používaný pro autentizaci API požadavků
- **Refresh Token** - dlouhodobý (7 dní), uložený v databázi, sloužící k obnovení access tokenu

Hesla jsou bezpečně hashována pomocí **bcryptjs** s salt rounds nastaveným na 10. Middleware chrání všechny chráněné API endpointy a automaticky ověřuje JWT tokeny.

Obrázek 4 ilustruje kompletní proces autentizace od registrace přes přihlášení až po odhlášení. Proces zahrnuje registraci s hashováním hesla, přihlášení s generováním JWT tokenů, autorizované API požadavky s ověřováním tokenu, refresh mechanismus pro obnovení access tokenu a bezpečné odhlášení se smazáním tokenů.

2.5.5 Platby

Pro zpracování plateb je implementována integrace se **Stripe**, což je flexibilní a spolehlivá platební brána. Implementace zahrnuje:

- **Stripe Elements** - bezpečný platební formulář na frontendu
- **Payment Intent API** - vytváření platebních intentů na backendu
- **Webhook** - automatické potvrzování plateb přes Stripe webhook endpoint
- **Refundace** - automatické vrácení peněz při schválení reklamace

Stripe zajišťuje PCI DSS compliance a podporuje různé platební metody. Pro akce zdarma (ticketPrice = 0) se platební proces přeskakuje a rezervace je okamžitě potvrzena.

2.5.6 Hosting a nasazení

Aplikace je připravena pro nasazení na různé platformy. Pro development a lokální testování je připraven **Docker Compose** setup, který automaticky spustí frontend, backend, PostgreSQL databázi a Stripe CLI pro testování webhooků.

Pro production deployment je aplikace připravena pro platformy jako:

- **Heroku** - jednoduchá PaaS platforma s automatickým deploymentem
- **Railway** - moderní alternativa k Heroku
- **AWS/Azure** - pro enterprise deployment s vlastní konfigurací

- **Vercel** (frontend) + **Render** (backend) - serverless architektura

V dokumentaci je připraven kompletní production checklist včetně nastavení environment proměnných, SSL certifikátů a databázových migrací.

2.6 Implementované pokročilé funkce

2.6.1 Email notifikace

Systém automaticky odesílá emailové notifikace při klíčových událostech:

- **Welcome email** - při registraci nového uživatele
- **Reservation confirmation** - po úspěšné rezervaci s detaily akce
- **Payment confirmation** - po potvrzení platby přes Stripe webhook
- **Reservation cancellation** - při zrušení rezervace s informací o refundaci
- **Event status change** - při změně stavu akce (pro organizátora)
- **Complaint response** - při vyřízení reklamace

Všechny emaily mají profesionální HTML design s gradientním headerem, jsou responzivní a obsahují call-to-action buttony s odkazy do aplikace. Pro odesílání je použita knihovna Nodemailer s SMTP konfigurací (podporuje Gmail, Outlook, vlastní SMTP server).

2.6.2 PDF vstupenky s QR kódy

Každá potvrzená rezervace (status PAID) umožňuje stažení PDF vstupenky, která obsahuje:

- Název a logo aplikace
- Detaily akce (název, datum, místo)
- Informace o uživateli (jméno, email)
- Počet vstupenek a celková cena
- QR kód s rezervačním kódem pro ověření při vstupu
- Rezervační kód v textové podobě

PDF je generováno na backendu pomocí knihovny PDFKit s podporou Unicode znaků (Arial TTF font) a QR kód je vytvořen knihovnou qrcode. Vstupenka je automaticky stažena jako .

2.6.3 Admin dashboard

Administrátoři mají přístup ke komplexnímu dashboardu s těmito funkcemi:

- **Správa uživatelů** - zobrazení všech uživatelů, filtrování podle role (ADMIN, ORGANIZER, USER), aktivace/deaktivace účtů, ochrana admin účtů před deaktivací, statistiky počtu uživatelů podle rolí
- **Správa všech akcí** - přehled akcí od všech organizátorů, možnost úpravy nebo smazání jakékoliv akce
- **Správa reklamací** - centralizovaný přehled všech reklamací s možností filtrování, schvalování/zamítání, automatická refundace

2.6.4 Testování

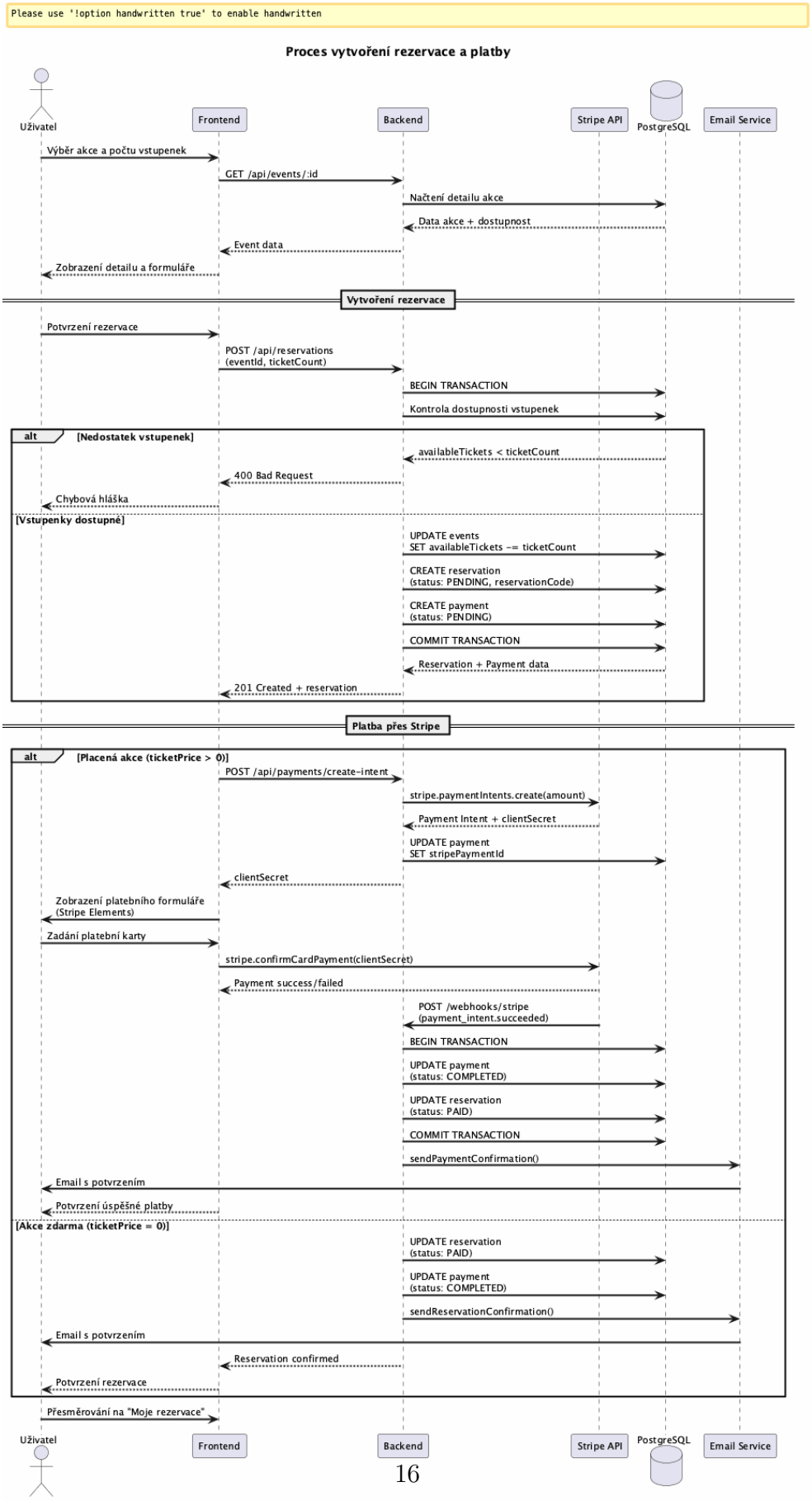
Projekt obsahuje komplexní testovací pokrytí:

- **Frontend testy** - rozsáhlá sada unit testů (Vitest, React Testing Library) pokrývající komponenty, utility funkce a konstanty
- **Backend testy** - integration testy (Jest, Supertest) testující všechny API endpointy s testovací PostgreSQL databází
- **E2E testy** - Playwright testy pokrývající kompletní user flows (registrace, rezervace, platba)
- **Performance testy** - Node.js skripty pro zátěžové testování API endpointů
- **Security testy** - Testy pro SQL injection, XSS, CSRF, rate limiting

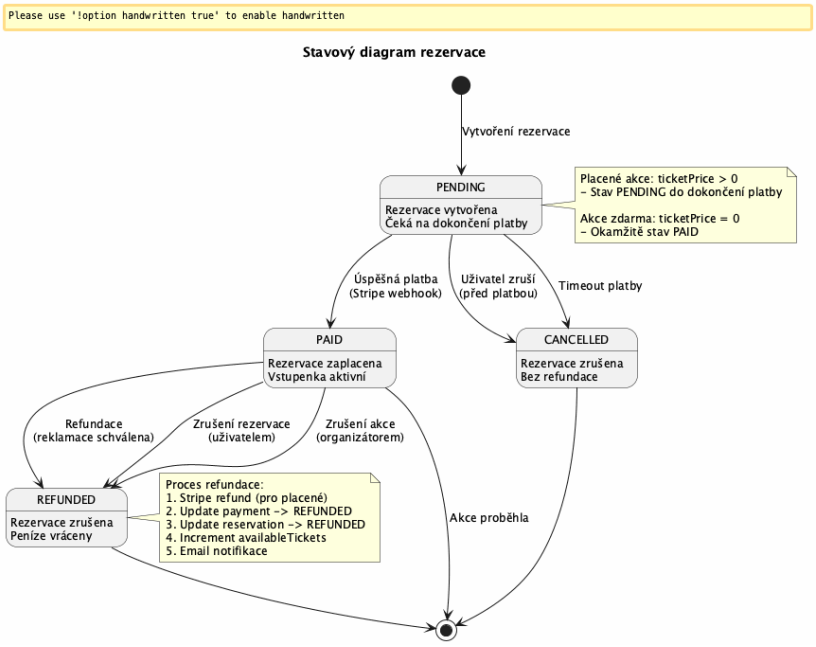
Celkové testovací pokrytí dosahuje vysokých hodnot na straně backendu i frontendu.

2.7 Relační model databáze

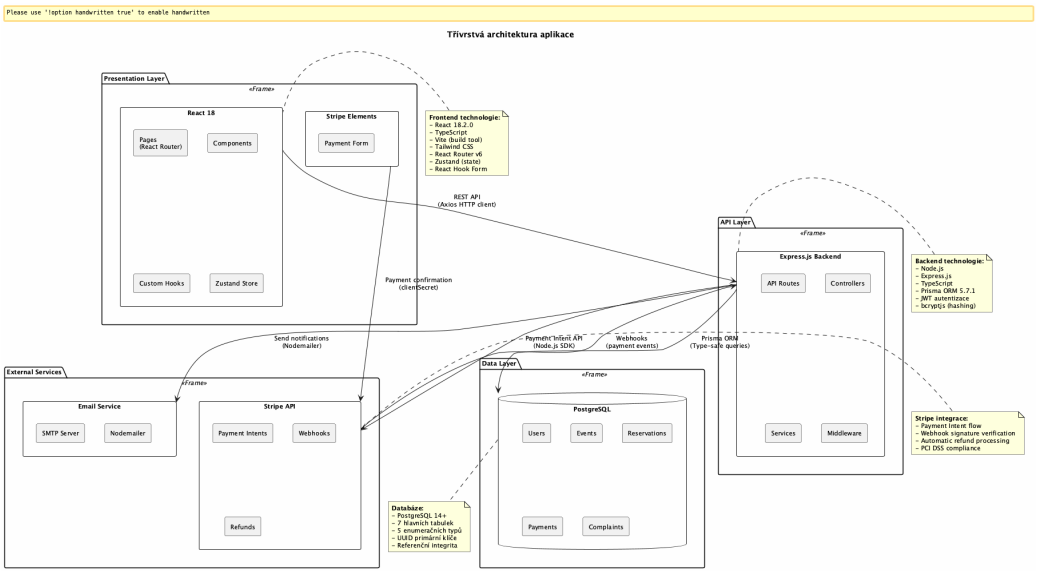
Databázový model obsahuje 5 hlavních entit (User, Event, Reservation, Payment, Complaint) a 5 enumeračních typů pro stavy. Všechny tabulky využívají UUID primární klíče a obsahují časové razítko pro sledování vytvoření a aktualizace záznamů. Vztahy mezi tabulkami zajišťují referenční integritu a podporují kaskádové operace.



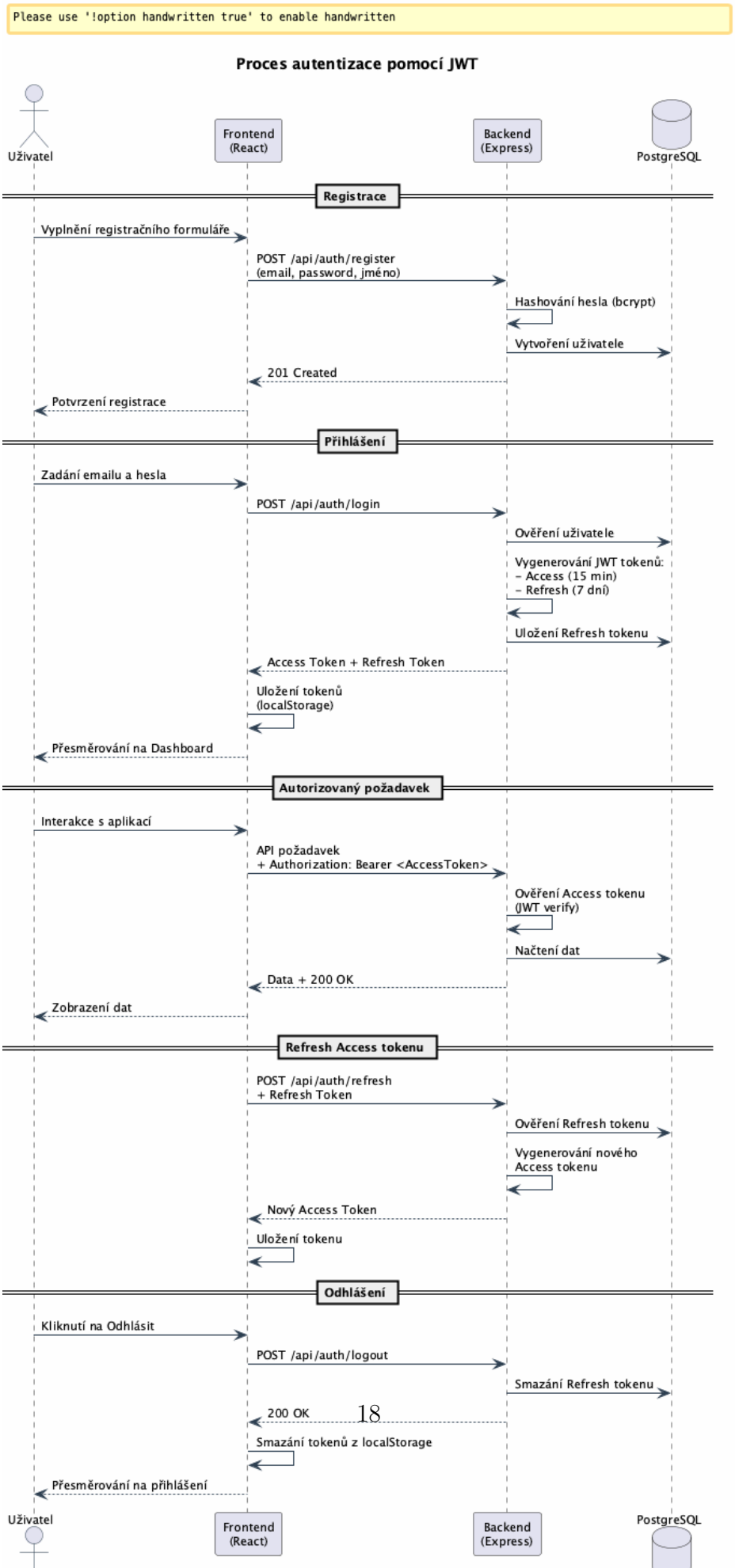
Obrázek 1: Sekvenční diagram procesu rezervace a platby

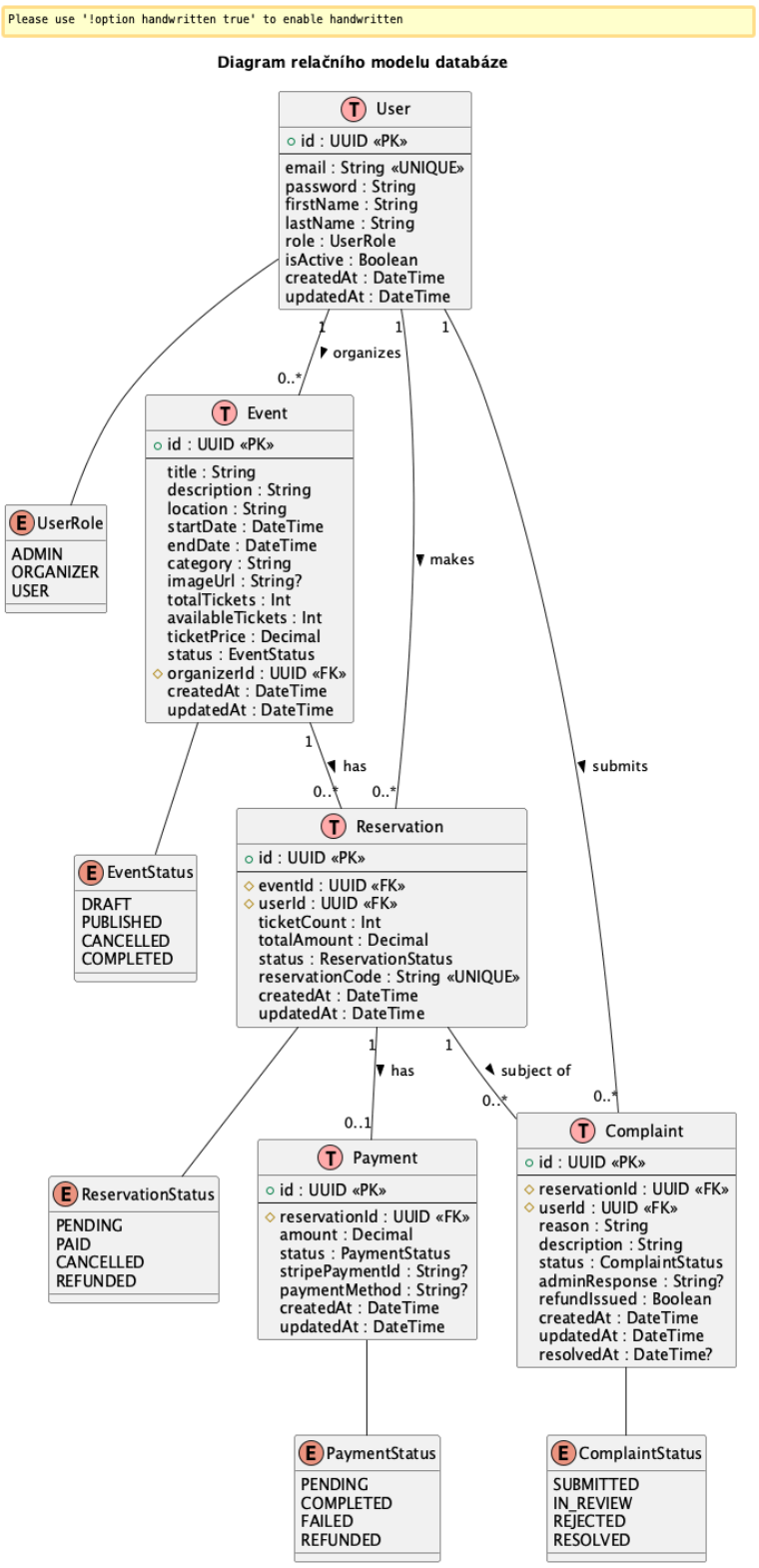


Obrázek 2: Stavový diagram rezervace



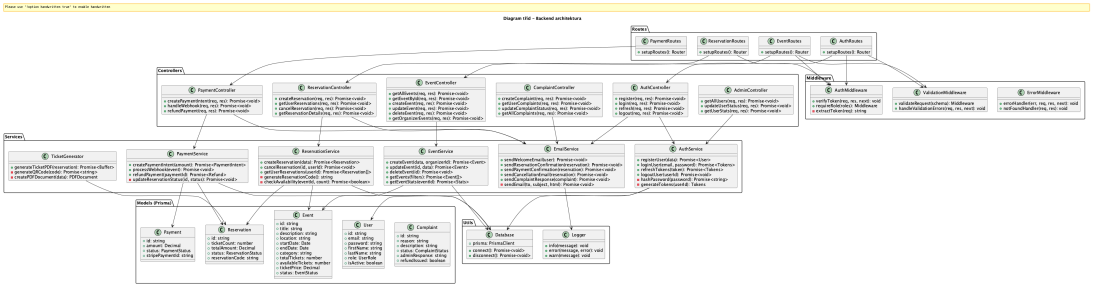
Obrázek 3: Třívrstvá architektura aplikace s externími službami


Please use '!option handwritten true' to enable handwritten
Proces autentizace pomocí JWT
Uživatel
Frontend (React)
Backend (Express)
PostgreSQL
Registrace
Vyplnění registračního formuláře
POST /api/auth/register (email, password, jméno)
Hashování hesla (bcrypt)
Vytvoření uživatele
201 Created
Potvrzení registrace
Přihlášení
Zadání emailu a hesla
POST /api/auth/login
Ověření uživatele
Vygenerování JWT tokenů: – Access (15 min) – Refresh (7 dní)
Uložení Refresh tokenu
Access Token + Refresh Token
Uložení tokenů (localStorage)
Přesměrování na Dashboard
Autorizovaný požadavek
Interakce s aplikací
API požadavek + Authorization: Bearer <AccessToken>
Ověření Access tokenu (JWT verify)
Načtení dat
Data + 200 OK
Zobrazení dat
Refresh Access tokenu
POST /api/auth/refresh + Refresh Token
Ověření Refresh tokenu
Vygenerování nového Access tokenu
Nový Access Token
Uložení tokenu
Odhlášení
Kliknutí na Odhlásit
POST /api/auth/logout
Smazání Refresh tokenu
200 OK
Smazání tokenů z localStorage
Přesměrování na přihlášení
Uživatel
Frontend (React)
Backend (Express)
PostgreSQL



Obrázek 5: Relační model databáze - 5 tabulek s vazbami

2.8 Diagram tříd



Obrázek 6: Diagram tříd - Backend struktura (MVC pattern)

Architektura backendu je rozdělena do čtyř hlavních vrstev: Controllers (zpracování HTTP požadavků), Services (business logika), Middleware (autentizace, validace, error handling) a Database (Prisma ORM). Všechny služby sdílejí přístup k Prisma klientovi pro práci s databází. EmailService a TicketGenerator jsou pomocné třídy volané z různých služeb pro odesílání notifikací a generování PDF vstupenek.

3 Programátorská příručka

Tato kapitola slouží jako technická dokumentace pro vývojáře, kteří chtějí rozumět struktuře aplikace, upravit ji nebo ji rozšířit.

3.1 Klíčové algoritmy a implementace

3.1.1 Autentizace pomocí JWT

Systém využívá JSON Web Tokens pro autentizaci uživatelů. Implementace zahrnuje dva typy tokenů:

Access Token je krátkodobý token s platností 15 minut, používaný pro autentizaci API požadavků. Token obsahuje identifikátor uživatele (userId) a je podepsán tajným klíčem uloženým v prostředí (JWT_SECRET).

Refresh Token je dlouhodobý token s platností 7 dní, uložený v databázi. Slouží k obnovení access tokenu bez nutnosti opětovného přihlášení. Token je podepsán samostatným tajným klíčem (JWT_REFRESH_SECRET).

Middleware pro ověření tokenu extrahuje token z Authorization hlavičky HTTP požadavku, ověří jeho platnost a dekóduje obsah. V případě úspěšného ověření přidá identifikátor uživatele do požadavku a pokračuje v zpracování. Při neplatném nebo chybějícím tokenu vrací chybu 401 (Unauthorized) nebo 403 (Forbidden).

3.1.2 Atomické transakce při rezervaci

Vytvoření rezervace vyžaduje několik databázových operací, které musí být provedeny atomicky. Prisma ORM poskytuje transakční mechanismus zajišťující, že buď proběhnou všechny operace, nebo žádná.

Proces rezervace zahrnuje tyto kroky: Nejprve se kontroluje dostupnost vstupenek načtením akce z databáze. Pokud je dostupný počet menší než požadovaný, transakce se zruší s chybovou hláškou. Následně se snižuje počet dostupných vstupenek aktualizací pole availableTickets. Poté se vytváří nový záznam rezervace s vygenerovaným unikátním rezervačním kódem. Status rezervace je nastaven na PENDING pro placené akce nebo PAID pro akce zdarma. Nakonec se vytváří záznam platby propojený s rezervací.

Pokud jakýkoliv krok selže, celá transakce se automaticky vrátí zpět (rollback) a databáze zůstane v konzistentním stavu.

3.1.3 Zpracování Stripe webhooků

Stripe odesílá webhooky pro informování o změnách stavu plateb. Implementace webhook endpointu zajišťuje automatické potvrzení plateb.

Při přijetí webhooku se nejprve ověří podpis požadavku pomocí Stripe webhook secret klíče. Tím se zajistí, že požadavek skutečně pochází ze Stripe a nebyl modifikován.

Pro událost payment_intent.succeeded se v databázové transakci vyhledá platba podle Stripe Payment Intent ID, aktualizuje se její status na COMPLETED, aktualizuje se status rezervace na PAID a odešle se potvrzující email uživateli s detaily rezervace a odkazem na stažení vstupenky.

Pro událost payment_intent.payment_failed se analogicky zpracovává neúspěšná platba aktualizací statusu na FAILED a odesláním informačního emailu.

3.1.4 Proces refundace

Refundace je komplexní proces vyžadující koordinaci mezi Stripe API, databází a emailovým systémem. Celý proces probíhá v databázové transakci.

Proces začíná načtením rezervace včetně propojené platby a akce. Pokud je rezervace placená (totalAmount > 0), provede se Stripe refund voláním Stripe API s Payment Intent ID. Po úspěšné refundaci se aktualizuje status platby na REFUNDED.

Následně se aktualizuje status rezervace na REFUNDED nebo CANCELLED podle kontextu. Vrátí se vstupenky do akce zvýšením pole availableTickets o počet vstupenek z rezervace. Nakonec se odešle email uživateli s potvrzením refundace a informací o vrácení peněz.

Pro akce zdarma se Stripe refund přeskočí a provádí se pouze zrušení rezervace a vrácení vstupenek.

3.1.5 Generování PDF vstupenek

PDF vstupenka je generována na backendu pomocí knihovny PDFKit s integrovaným QR kódem.

Proces začíná vygenerováním QR kódu obsahujícího JSON data s rezervačním kódem, ID akce a ID uživatele. QR kód je vytvořen jako Data URL obrázek pomocí knihovny qrcode.

Poté se vytváří PDF dokument s nastavenou velikostí A4 a okraji 50 bodů. Dokument používá Arial font pro podporu českých znaků (Unicode). Obsah PDF zahrnuje hlavičku s nadpisem "Vstupenka"a názvem akce, detaily události (místo, datum, počet vstupenek, celkovou cenu), vložený QR kód v rozměrech 200x200 bodů a rezervační kód v textové podobě pro ruční ověření.

PDF je generováno do paměťového bufferu a vráceno jako binární data připravená ke stažení. Soubor je pojmenován podle vzoru vstupenka-XXXXXXXX.pdf, kde XXXXXXXX je rezervační kód.

3.2 Možnosti rozšíření aplikace

3.2.1 Přidání nové databázové entity

Pro přidání nové entity je třeba nejprve aktualizovat Prisma schema soubor přidáním nového modelu s definicí polí a relací. Následně se spustí migrace databáze příkazem npx prisma migrate dev. Vytvoří se service třída s metodami pro CRUD

operace. Implementuje se controller pro zpracování HTTP požadavků. A nakonec se přidají API routes pro nové endpointy.

3.2.2 Přidání nového typu emailové notifikace

EmailService se rozšíří o novou metodu pro odesílání specifického typu emailu. Vytvoří se HTML template pro daný typ emailu s responzivním designem. Nová metoda se zavolá z příslušného service při vzniku události.

3.2.3 Integrace nové platební metody

Pro integraci další platební brány se vytvoří nový service pro konkrétní provider (např. PayPal). Implementují se metody pro vytvoření platby a zpracování webhooků. PaymentService se aktualizuje pro podporu více platebních metod s podmíněným větvením podle zvolené metody.

3.2.4 Přidání nové uživatelské role

V Prisma schema se rozšíří enum UserRole o novou hodnotu. Middleware pro kontrolu rolí se aktualizuje pro podporu nové role. Na příslušných API endpointech se přidá role do povolených rolí pomocí roleMiddleware.

3.3 Testování

Projekt obsahuje tři úrovně testování. Frontend testy jsou implementovány pomocí Vitest a React Testing Library, pokrývají React komponenty, utility funkce a konstanty. Backend testy využívají Jest a Supertest pro integration testing všech API endpointů s testovací PostgreSQL databází. E2E testy jsou napsány v Playwright a testují kompletní user flows od registrace přes rezervaci až po platbu.

3.4 Deployment

Pro lokální vývoj je připraven Docker Compose setup automaticky spouštějící frontend, backend, PostgreSQL databázi a Stripe CLI. Pro production je aplikace připravena pro deployment na platformy jako Heroku, Railway, AWS/Azure nebo kombinaci Vercel (frontend) a Render (backend). Důležité je nastavení všech environment proměnných včetně databázových credentials, JWT tajných klíčů, Stripe API klíčů a SMTP konfigurace pro emaily.

4 Uživatelská příručka

Tato kapitola poskytuje kompletní návod pro koncové uživatele aplikace ve všech rolích.

4.1 První kroky

4.1.1 Registrace

Pro vytvoření nového účtu je třeba otevřít aplikaci v prohlížeči a kliknout na tlačítko "Registrovat se". Ve formuláři se vyplní email, který bude sloužit jako přihlašovací jméno, heslo s minimální délkou 6 znaků, křestní jméno a příjmení. Po kliknutí na "Vytvořit účet"se zobrazí potvrzení a na uvedený email přijde uvítací zpráva.

4.1.2 Přihlášení

Přihlášení probíhá na hlavní stránce kliknutím na "Přihlásit se", zadáním emailu a hesla a potvrzením tlačítkem "Přihlásit". Po úspěšném přihlášení dojde k automatickému přesměrování na domovskou stránku s přehledem akcí.

4.2 Prohlížení a rezervace akcí

4.2.1 Prohlížení akcí

Hlavní stránka zobrazuje seznam všech dostupných akcí. Každá karta obsahuje název akce, datum a čas konání, místo, kategorii, cenu vstupenky a počet dostupných vstupenek.

Akce lze filtrovat podle kategorií: Hudba (koncerty, festivaly), Divadlo (představení), Film (projekce), Sport (sportovní události), Technologie (meetupy, hackathony), Vzdělávání (přednášky, workshopy) a Jiné (ostatní akce). Filtr se aktivuje kliknutím na příslušné tlačítko kategorie.

Pro vyhledávání konkrétní akce slouží vyhledávací pole v horní části stránky. Výsledky se filtrují automaticky při zadávání textu.

4.2.2 Detail a rezervace

Kliknutím na kartu akce se zobrazí detail s úplným popisem, obrázkem, informacemi o organizátorovi a tlačítkem pro rezervaci. Pro vytvoření rezervace se klikne na "Rezervovat vstupenky", vybere počet vstupenek (maximum 10 kusů na rezervaci), zkontroluje celková cena a potvrdí rezervace.

U placených akcí následuje platební proces. Otevře se formulář Stripe Elements pro bezpečné zadání platební karty. Po zadání čísla karty, data expirace a CVC kódu se platba potvrdí tlačítkem "Zaplatit". Pro akce zdarma se rezervace potvrdí okamžitě bez platebního procesu.

Po úspěšné platbě se zobrazí potvrzovací zpráva, na email přijde potvrzení a uživatel je přesměrován na stránku "Moje rezervace".

4.3 Správa rezervací

V sekci "Moje rezervace"je k dispozici přehled všech rezervací s jejich statusy: Čeká na platbu (rezervace vytvořena, čeká se na dokončení platby), Zaplaceno (platba úspěšná, vstupenka aktivní), Zrušeno (rezervace zrušena) a Refundováno (rezervace zrušena a peníze vráceny).

U zaplacených rezervací je k dispozici tlačítko "Stáhnout vstupenku", které stáhne PDF soubor obsahující QR kód pro skenování při vstupu, detaily akce, jméno uživatele, počet vstupenek a rezervační kód.

Aktivní rezervaci lze zrušit tlačítkem "Zrušit rezervaci". Systém automaticky vrátí vstupenky do eventu, spustí refundaci přes Stripe u placených akcí, odešle email s potvrzením a změní stav na "Refundováno". Refundace může trvat 5-10 pracovních dní podle banky.

4.4 Reklamace

Pro podání reklamace se v seznamu "Moje rezervace"najde příslušná rezervace a klikne na "Podat reklamaci". Ve formuláři se vybere důvod reklamace z nabízených možností a vyplní podrobný popis problému.

Reklamace prochází těmito stavy: Podáno (nově vytvořená reklamace), V řešení (administrátor reklamaci řeší), Vyřízeno (schváleno, proběhla refundace) a Zamítnuto (reklamace zamítnuta).

Po vyřízení uživatel obdrží email s odpovědí administrátora a v případě schválení automatickou refundaci.

4.5 Funkce pro organizátory

Uživatelé s rolí Organizátor mají přístup k dalším funkcím v sekci "Moje akce".

4.5.1 Vytvoření akce

Pro vytvoření nové akce se vyplní formulář obsahující název, podrobný popis s možností formátování, místo konání, datum a čas začátku, datum a čas konce, kategorii z nabídky, volitelný odkaz na obrázek, celkový počet vstupenek (kapacitu) a cenu vstupenky v korunách (0 pro akce zdarma).

Akce je vytvořena ve stavu "Koncept"a není veřejně viditelná až do publikování.

4.5.2 Správa akcí

U vytvořených akcí jsou k dispozici tyto operace: Upravit umožňuje změnit údaje akce, přičemž některé údaje nelze měnit po vytvoření rezervací. Publikovat změní

stav z "Koncept"na "Publikováno"a učiní akci viditelnou všem uživatelům. Zrušit akci změní stav na "Zrušeno", automaticky zruší všechny rezervace, provede refundace a odešle emaily účastníkům.

Zobrazit statistiky ukazuje celkový počet rezervací, počet prodaných vstupenek, celkovou výši tržeb a obsazenost v procentech.

Zobrazit rezervace poskytuje seznam všech rezervací s jmény zákazníků, emaily, počty vstupenek, statusy a rezervačními kódy, což slouží k ověření při vstupu na akci.

4.6 Funkce pro administrátory

Administrátoři mají plný přístup k systému včetně dashboardu zobrazujícího celkové statistiky: počet uživatelů podle rolí, počet akcí podle stavů, počet rezervací a počet nevyřízených reklamací.

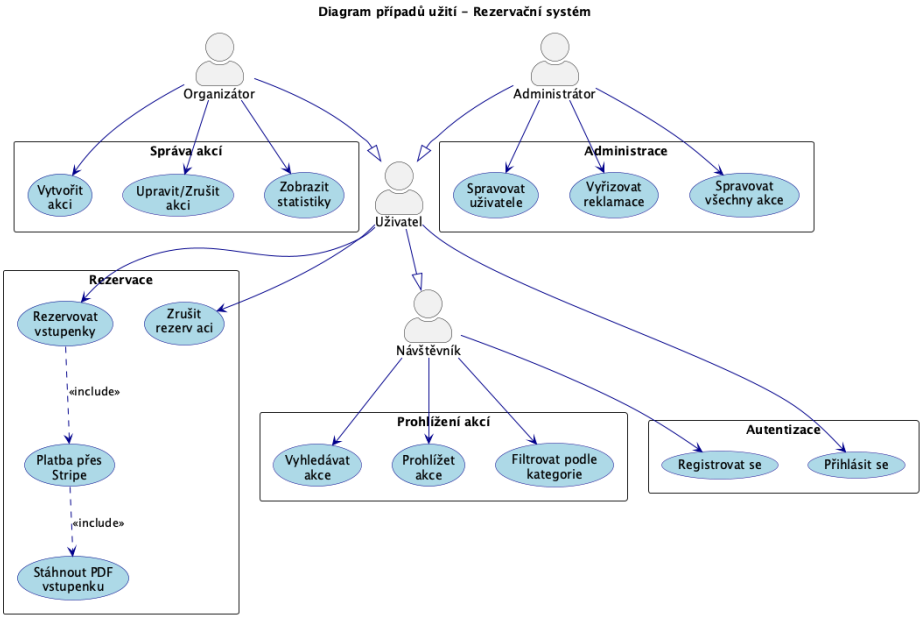
4.6.1 Správa uživatelů

V sekci "Uživatelé"je k dispozici seznam všech uživatelů s možností filtrování podle role, aktivace nebo deaktivace účtu a zobrazení detailních informací. Administrátorské účty jsou chráněny proti deaktivaci.

4.6.2 Správa akcí a reklamací

Administrátoři vidí akce od všech organizátorů a mohou je upravovat nebo mazat. V sekci "Reklamace"je centralizovaný přehled všech reklamací s možností změny stavu na "V řešení", napsání odpovědi pro zákazníka a finálního rozhodnutí: "Vyřízeno"provede automatickou refundaci, "Zamítnuto"ukončí reklamaci bez vrácení peněz. Zákazník obdrží email s odpovědí v obou případech.

4.7 Diagram případu použití



Obrázek 7: Diagram případu použití - Rezervační systém