

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 Zácpal, Ph.D.
Počet stran: 16
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, Palacky University Olomouc
Year of defense: 2025
Study program: Information Technologies, combined form
Supervisor: Mgr. Jiří Zácpal, Ph.D.
Page count: 16
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, děkuji, děkuji.

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	7
1.1 Popis jednotlivých řešení	7
1.1.1 Eventbrite	7
1.1.2 Ticketmaster	7
1.1.3 Meetup	7
1.1.4 Eventzilla	7
1.2 Hodnocení kladů a záporů	8
1.3 Vyhodnocení přínosů	8
1.4 Přínosy vlastní práce	8
2 Specifikace řešení	9
2.1 Uživatelské role	9
2.2 Správa událostí	9
2.3 Rezervace	9
2.4 Reklamace a vrácení vstupenek	10
2.5 Návrh vhodných technologií	10
2.5.1 Backend	10
2.5.2 Databáze	11
2.5.3 Frontend	11
2.5.4 Autentizace a autorizace	11
2.5.5 Platby	12
2.5.6 Hosting a nasazení	12
2.6 Implementované pokročilé funkce	12
2.6.1 Email notifikace	12
2.6.2 PDF vstupenky s QR kódy	13
2.6.3 Admin dashboard	13
2.6.4 Testování	13
2.7 Relační model databáze	14
2.8 Diagram tříd	14
2.9 Diagram případu použití	14

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 dalsím přínosem této práce. Tím se obohatí univerzitní komunita a nabídnu 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 cen.

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 (`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.

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 stavu a reagovat na ně. Při vyřízení reklamace admin vyplní odpověď a volitelně zaškrtnut 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 Návrh vhodných 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.

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 vysoký a flexibilní frontendový framework. React umožňuje vytváření dynamických uživatelských rozhraní a díky komponentovému přístupu usnadňuje správu a opakování použití kódů.

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.

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 detailem 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

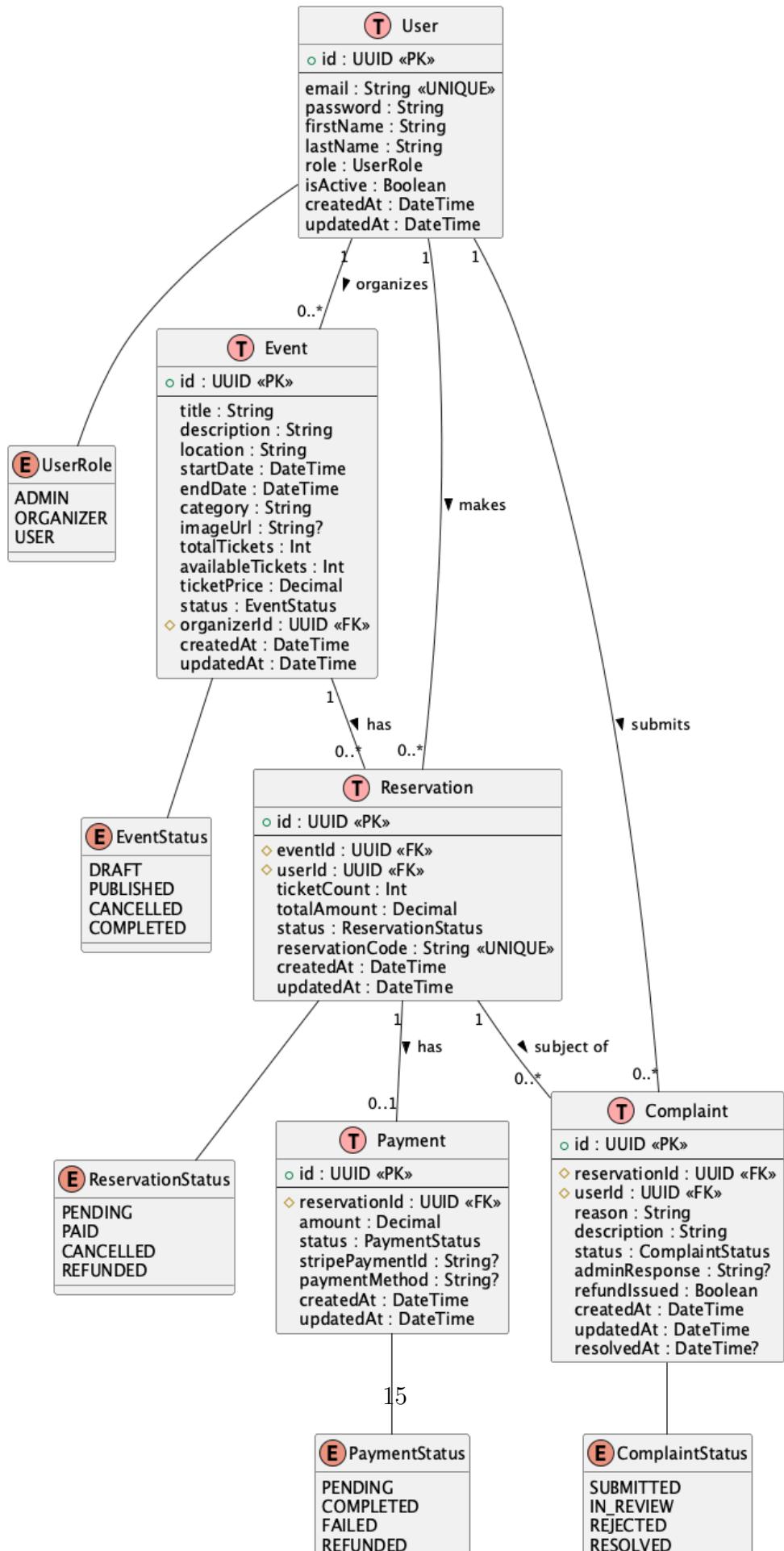
Databázový model obsahuje 5 hlavních entit (User, Event, Reservation, Payment, Complaint) a 5 enumeračních typů pro stavky. Všechny tabulky využívají UUID primární klíče a obsahují časové razítko pro sledování vytvoření a aktualizace záznamů. Vztahy mezi tabulkami zajišťují referenční integritu a podporují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

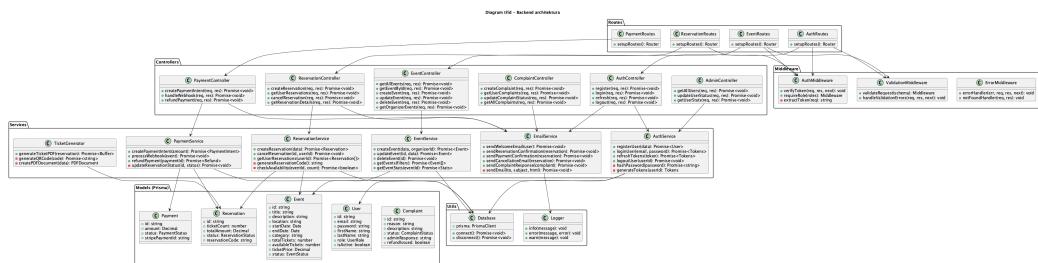
2.8 Diagram tříd

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.

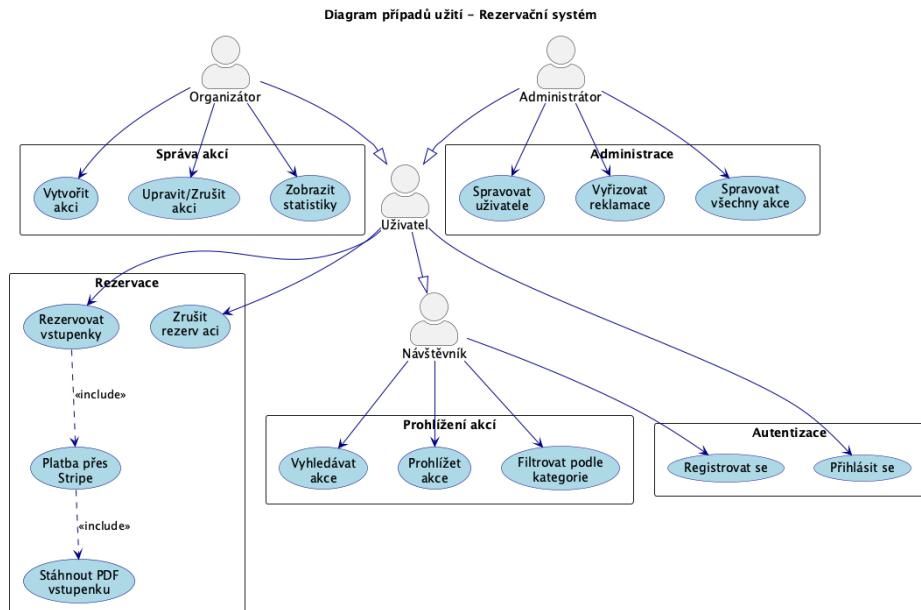
2.9 Diagram případu použití

Diagram relačního modelu databáze





Obrázek 2: Diagram tříd - Backend struktura (MVC pattern)



Obrázek 3: Diagram případu použití - Rezervační systém