

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informatika

Menuroo - Inteligentní systém pro správu objednávek a
rezervaci jídel

Tomáš Kalhous, Michal Vaniš, Matyáš Tázlar

Pardubický kraj

Pardubice, 2025

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informatika

Menuroo - Inteligentní systém pro správu objednávek a rezervaci jídel

Menuroo - Intelligent system for food ordering and reservation

Jméno: Tomáš Kalhous, Michal Vaniš, Matyáš Tázlar

Škola: Delta - Střední škola informatiky a ekonomie, s.r.o.

Kraj: Pardubický kraj

Konzultant: Ing. Jiří Formánek

Pardubice, 2025

Prohlášení

Prohlašuji, že jsem svou práci soč vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce soč jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

v Pardubicích dne 18. února 2025

Tomáš Kalhous, Michal Vaniš, Matyáš Tázlar

Poděkování

Chtěli bychom vyjádřit upřímné poděkování našemu školiteli, Ing. Jiřímu Formánkovi, za jeho cenné rady, podnětné připomínky a odborné vedení, které významně přispělo ke vzniku této práce. Dále děkujeme Diegu Portillovi za vytvoření základních funkcí backendu a cennou pomoc s návrhem infrastruktury. Zvláštní poděkování patří také Metěji Tobiáši Moravcovi za kreativní návrh loga a pečlivé zpracování UI/UX.

Abstrakt

Tato práce se zabývá vývojem a implementací softwarové platformy Menuroo, která je navržena pro optimalizaci správy objednávek, rezervací a zásob surovin v restauracích. Cílem systému je zefektivnit provoz, snížit administrativní zátěž a minimalizovat plýtvání surovinami prostřednictvím automatizace a integrace klíčových procesů. Menuroo propojuje správu objednávek se sledováním zásob a plánováním jídel na základě rezervací, čímž umožňuje restauracím lépe řídit nákup surovin a optimalizovat využití zdrojů. Platforma je navržena jako modulární a škálovatelná, což umožňuje její snadnou adaptaci na různé typy gastronomických provozů. Důraz je kladen na intuitivní uživatelské rozhraní, které usnadňuje obsluhu a zvyšuje efektivitu personálu. Výsledkem implementace Menuroo je nejen zlepšení interních procesů restaurací, ale také zvýšení spokojenosti zákazníků díky plynulejšímu provozu a vyšší kvalitě služeb.

Klíčová slova

restaurace, správa objednávek, automatizace, optimalizace zásob, efektivita

Abstract

This thesis focuses on the development and implementation of the Menuroo software platform, designed to optimize order management, reservations, and ingredient inventory in restaurants. The system aims to enhance operational efficiency, reduce administrative workload, and minimize food waste through automation and integration of key processes. Menuroo connects order management with inventory tracking and reservation-based meal planning, enabling restaurants to better control ingredient procurement and optimize resource utilization. The platform is designed to be modular and scalable, allowing for easy adaptation to various types of food service establishments. Emphasis is placed on an intuitive user interface that simplifies operations and improves staff efficiency. The implementation of Menuroo not only enhances internal restaurant processes but also increases customer satisfaction by ensuring smoother operations and higher-quality service.

Keywords

restaurant, order management, automation, inventory optimization, efficiency

Obsah

1	Úvod	2
1.1	Věcné představení	2
1.2	Jak jsme se k tomu dostali	2
1.3	Logo a jméno	3
2	Teoretická část	4
2.1	Informační systémy	4
2.2	Architektury	5
2.3	Technologie	6
2.4	Infrastruktura	8
2.5	Bezpečnost	9
3	Praktická část	12
3.1	Propojení systému a tok dat	12
3.2	Verze systému	13
3.3	Popis jednotlivých částí	15
3.4	Design	24
3.5	Největší problémy při vývoji	25
3.6	Businessová stránka projektu	26
4	Závěr	27

1 Úvod

1.1 Věcné představení

Předmětem této práce je systém Menuroo, který slouží k rezervaci jídla v restauracích a správě objednávek. Strávníci si mohou prohlédnout nabídku jídel, kterou do systému vkládají zapojené restaurace, a rezervovat si jídlo na konkrétní dny až do výše finančního kreditu, který v aplikaci mají. Kredit si doplňují převodem peněz na účet pod svým variabilním symbolem (aplikace je napojena na API internetového bankovníctví). Restaurace mají k dispozici statistiky, kolik a jakých jídel je na jednotlivé dny objednáno, což přispívá k lepší efektivitě a redukci plýtvání surovinami. Administrátorský panel číšníka umožňuje kontrolu objednaného oběda hostem restaurace.

1.2 Jak jsme se k tomu dostali

Před dvěma lety jsme přišli o možnost stravování ve školní jídelně, kam naše škola chodila 25 let. Pan ředitel řešil náhradní možnosti stravování a jako jedno z nich domluvil možnost obědů v restauraci Zlatá Štika, s níž dohodl pro studenty snížené ceny.

Pro objednávku obědů se používaly Google Forms. Následně pak paní sekretářka tiskla papírové stravenky, na které lepila každý den jiné samolepky. Stravenky se vydávaly strávnickům na celý týden proti platbě v hotovosti. V restauraci číšníci výměnou za stravenku přinesli studentům jejich vybrané jídlo.

Bylo zřejmé, že toto řešení je organizačně náročné a neefektivní, a proto byl hned týden po začátku školního roku vyhlášen hackathon, během něhož měl vzniknout systém pro digitalizaci těchto úkonů. Zde jsou kořeny systému Menuroo. Původně sloužil jen k objednávání obědů v restauraci Zlatá Štika a takto je zde nasazen a provozován dodnes.

Při zavádění systému jsme však objevili mezeru na trhu. Když něco funguje aaše studenty a Zlatou Štiku, proč by stejné řešení nemělo fungovat i obecně? Existují stravovací aplikace jako Foodora nebo Wolt, ale žádná z nich nezaručuje, že v konkrétní den dostanete přesně to jídlo, na které máte chuť. Navíc nabízejí pouze doručení jídla, nikoliv možnost rezervace a konzumace přímo v restauraci.

Náš systém by mohl výrazně pomoci především lidem pracujícím ve velkých institucích, které nemají pro zaměstnance zajištěné obědy, takže musí během obědové pauzy chodit do okolních restaurací. V těchto podmínkách se často stává, že jídlo, na které máte chuť, je už vyprodané. Systém Menuroo by snížil stres tím, že by lidem zaručil rezervaci jídla. Navíc by se stravovali přímo v restauraci, což bývá mnohdy příjemnější než jíst v přeplněných kancelářích. Oproti dovážkovým službám stravování přímo v restauracích produkuje i menší množství odpadu a emisí.

Po úspěchu systému jsme dokonce získali poptávku od starosty městské části Bratislavy na adaptaci systému pro rozvoz jídel seniorům. Systém jsme museli rozšířit i o „papírový“ způsob ovládání, na který jsou důchodci zvyklí. Tato verze nyní běží v testovacím režimu, kde se doladují detaily a provádí finální úpravy.

1.3 Logo a jméno

Jméno "Menuroo" je chytrá slovní hříčka, která v sobě spojuje dvě slova: "menu" a "kangaroo" (klokan). Slovo "menu" je v kontextu restaurací a stravování zcela srozumitelné, odkazuje na nabídku jídel a nápojů. Klokan, jakožto zvíře, je pak zvolen pro svou charakteristickou schopnost skákat. Spojením těchto dvou slov vzniká "Menuroo", které lze interpretovat jako "skočte si na menu". Toto jméno je nejen zapamatovatelné, ale také vystihuje podstatu naší služby – rychlý a snadný přístup k nabídce jídel.



Obrázek 1.1: Logo Menuroo

2 Teoretická část

2.1 Informační systémy

Informační systém lze obecně definovat jako integrovanou soustavu technických, organizačních a lidských komponent, jejichž cílem je získávání, zpracování, ukládání a distribuce informací. [1].

2.1.1 Komponenty informačního systému

Informační systémy se skládají z několika základních vrstev:

1. **Datová vrstva** – tato vrstva zahrnuje databáze a úložiště, ve kterých jsou uchová-
vány důležité informace, např. data o menu, objednávkách či platebních transakcích.
2. **Aplikační (logická) vrstva** – zde je implementována logika, která zajišťuje auto-
matizaci procesů a integraci jednotlivých modulů systému (např. propojení správy
objednávek s tiskem faktur).
3. **Prezentační vrstva** – tato vrstva poskytuje uživatelské rozhraní, přes které mohou
koncoví uživatelé, ať už provozovatelé restaurací nebo jejich zákazníci, interagovat
se systémem.

2.1.2 Technologie dle komponent

Každá z těchto vrstev je implementována technologiemi pro danou vrstvu nejvhodnějšími. Programovací jazyk, který se hodí pro implementaci aplikační vrstvy, byť sám o sobě efektivní a robustní, nemusí být vhodný pro implementaci například prezentační vrstvy.

1. **Datová vrstva** – Nejčastěji používanou technologií pro relační databáze je MySQL, pro úložiště souborů pak Amazon S3. My jsme si vybrali PostgreSQL, protože je robustní a dobře škálovatelný.
2. **Aplikační (logická) vrstva** – Pro programování aplikační vrstvy patří velká škála programovacích jazyků, jako například Rust, C# nebo JavaScript. Zvolili jsme programovací jazyk Golang, který je rychlý a efektivní, existují pro něj mnohé knihovny a má velkou komunitní podporu.

3. **Prezentační vrstva** – Mezi běžně používané technologie pro vývoj uživatelských rozhraní na webu patří React, Svelte, Vue nebo Angular. Pro mobilní aplikace existují technologie jako Flutter nebo React Native, které umožňují vytvářet multiplatformní aplikace, které mohou být spuštěny jak na iOS, tak na Androidu. Pro naše účely jsme zvolili React na programování webového rozhraní a Flutter pro mobilní aplikace.

2.1.3 Architektura a integrace klíčových funkcí

Vyvíjený systém pro restaurace využívá modulární architekturu, která integruje samostatné funkční celky do jednotného informačního prostředí. Konkrétně tento přístup zahrnuje:

- **Modul pro správu dat** – zajišťuje uchovávání a aktualizaci informací týkajících se nabídky, objednávek.
- **Automatizace procesů** – umožňuje automatické generování faktur, správu objednávek a poskytuje analytické nástroje pro podporu rozhodovacích procesů.
- **Integrační nástroje a API** – zajišťují komunikaci mezi jednotlivými moduly a umožňují budoucí integraci externích systémů či nových funkcionalit.

2.2 Architektury

Pojmem architektura se v informatice rozumí způsob, jakým jsou jednotlivé komponenty systému organizovány a jak spolu komunikují. V rámci softwarového inženýrství se používají různé architektonické vzory, které definují základní strukturu a chování systému. V následujících sekcích jsou popsány některé z nejpoužívanějších architektur a jejich využití. [2] Pro tento projekt byla zvolena více vrstvá architektura s oddělením prezentační, aplikační a datové vrstvy. Mohli bychom ji také popsat jako distribuovanou architekturu. Umožňuje nám oddělení různých částí systému a dosáhnout tak vysoké flexibility a škálovatelnosti.

2.2.1 Monolitická architektura

Monolitická architektura je přístupem k návrhu softwarových systémů, kdy je celý systém implementován jako jediná aplikace. Všechny komponenty, včetně uživatelského rozhraní, aplikační logiky a datové vrstvy, jsou integrovány do jednoho celku. Tento přístup je jednoduchý na implementaci a nasazení, ale může být obtížné udržovat a škálovat s rostoucí složitostí systému.

2.2.2 Mikroslužby

Mikroslužby jsou moderním architektonickým vzorem, který rozděluje aplikaci na malé, nezávislé služby, které spolu komunikují pomocí API. Každá služba je navržena tak, aby vykonávala jednu konkrétní úlohu a měla vlastní databázi. Tento přístup umožňuje snadnou škálovatelnost a udržitelnost systému, ale zvyšuje složitost implementace a správy infrastruktury.

2.2.3 Serverless

Serverless architektura je novým trendem v oblasti cloud computingu, který umožňuje vývojářům psát a nasazovat kód bez nutnosti správy serverů. Aplikace jsou spouštěny v reakci na události. Tento přístup snižuje náklady na provoz a umožňuje rychlejší vývoj a nasazení nových funkcionalit.

2.3 Technologie

V rámci vývoje softwarového systému je důležité zvolit vhodné technologie, které umožní efektivní implementaci a provoz aplikace. V následujících sekcích jsou popsány některé z nejpoužívanějších technologií a jejich využití v různých částech systému.

2.3.1 Frontend

Očekávání toho, jak se bude prezentační vrstva chovat se za posledních několik let dramaticky změnilo. V minulosti bylo běžné, že veškerá logika webové aplikace probíhá na serveru a webový prohlížeč je pouze jakýmsi zobrazovacím nástrojem, který dostává hotové HTML stránky ze serveru. Nyní je však běžné, že velká část logiky probíhá i na straně klienta, což umožňuje vytvářet velice interaktivní aplikace. Převedením veškeré logiky na klient avšak dojde ke ztrátě některých výhod, které serverová logika má. Naštěstí jsou zde technologie a frameworky, které umožňují vytvářet interaktivní aplikace, aniž byste museli ztratit výhody serverové logiky.

Moderní frontendové technologie

Firma Vercel významně ovlivňuje vývoj moderního webu, zejména prostřednictvím frameworku Next.js. Tento framework využívá server-side rendering (SSR), což znamená, že se stránky generují na serveru a do prohlížeče se odesílají již vykreslené. Tento přístup přináší výhody, jako je rychlejší načítání stránek a lepší optimalizace pro vyhledávače (SEO). Next.js také umožňuje generování statických webových stránek, což zajišťuje vysoký výkon a škálovatelnost. V našem projektu jsme původně využívali Next.js, ale později jsme přešli na Remix.js, který staví na podobných principech, ale nabízí odlišný přístup k některým aspektům vývoje.

Architektura frontendu a její vývoj

Při vývoji webových aplikací lze využít širokou škálu architektur, od aplikací plně běžících na straně klienta až po ty, kde veškerá logika probíhá na serveru a komunikace s uživatelem spočívá v odesílání HTML stránek. Pro náš projekt jsme zvolili hybridní architekturu, která kombinuje výhody serverové a klientské logiky. Tato architektura umožňuje vytvářet interaktivní aplikace, aniž bychom se vzdali výhod serverového zpracování. Mezi tyto výhody patří například skrytí komunikace s backendem před klientem, který komunikuje pouze s webovým serverem. Dalším přínosem je větší autonomie frontendu, který tak není zcela závislý na backendu.

2.3.2 Mobilní frontend

Mobilní zařízení se staly nedílnou součástí každodenního života. Mezi nejpopulárnější operační systémy, které jsou na této platformě používány, patří certifikovaný UNIX, iOS a UNIXu podobný Android. I když tyto operační systémy sdílí POSIXové standardy, tvorba uživatelských aplikací je pro ně odlišná. iOS umožňuje vývojářům tvořit aplikace s pomocí jazyka Swift a knihovny SwiftUI, zatímco Android nabízí jazyk Kotlin a knihovnu Jetpack Compose. Pro vývoj multiplatformních aplikací existují technologie jako Flutter nebo React Native, které umožňují vytvářet aplikace, které mohou být spuštěny na obou platformách. Pro naše účely jsme použili Flutter, který je díky svému vykreslovacímu enginu velmi rychlý a umožňuje tvorbu krásných uživatelských rozhraní.

Architektura mobilního frontendu a jeho vývoj

Mobilní aplikace představují složitý soubor vrstev, který musí efektivně spolupracovat, aby byla zajištěna plynulost, stabilita a maximální využití systémových zdrojů. V moderním vývoji mobilních aplikací, zejména u multiplatformních řešení jako je Flutter, se architektura skládá z několika klíčových komponent, jejichž vzájemná integrace je nezbytná pro dosažení optimálního výkonu a uživatelského zážitku. Na nejvyšší úrovni se nachází aplikační logika a uživatelské rozhraní, kde Flutter využívá deklarativní přístup k tvorbě widgetů a dynamickému vykreslování. Díky vlastním renderovacímu enginu napsanému v C/C++ je možné dosáhnout vysokého výkonu a plynulosti animací, i když jsou operace prováděny v jazyce Dart. Jako vývojáři tak máme možnost vytvářet interaktivní rozhraní, která zároveň těží z nativní optimalizace pro danou platformu.

2.3.3 Backend

Backend je částí softwarového systému, která zajišťuje zpracování dat a logiku aplikace. Backend je obvykle implementován jako webový server, který komunikuje s frontendem pomocí API. Pro vývoj backendu lze využít různé technologie a frameworky, které umožňují efektivní implementaci a provoz aplikace. V moderním vývoji backendových systémů se často využívají frameworky jako Node.js, Django, Flask, nebo Ruby on Rails, které poskytují robustní nástroje pro rychlý vývoj a správu aplikací. Tyto frameworky umožňují snadnou integraci s databázemi, jako jsou MySQL, PostgreSQL, MongoDB nebo Redis, a podporují různé architektonické styly, například REST nebo GraphQL. Backend může být navržen jako monolitická aplikace nebo jako sada mikroslužeb. Monolitická architektura je jednodušší na vývoj a nasazení, ale může být obtížnější škálovat. Mikroslužby naopak umožňují rozdělení aplikace na menší, nezávislé komponenty, které lze snadno škálovat a spravovat. Tento přístup však vyžaduje složitější orchestraci a správu komunikace mezi službami, například pomocí nástrojů jako Kubernetes nebo Docker Swarm. Pro zajištění bezpečnosti backendu je důležité implementovat šifrování dat, autentizaci a autorizaci uživatelů, a také ochranu proti běžným útokům, jako jsou SQL injection nebo cross-site scripting (XSS). Dále je vhodné využívat logování a monitorování, například pomocí nástrojů jako Prometheus nebo ELK stack, pro detekci a řešení problémů v reálném čase. V našem projektu jsme zvolili architekturu mikroslužeb, která nám umožňuje snadno škálovat jednotlivé části systému podle potřeby. Pro implementaci backendu jsme použili jazyk Go bez žádných frameworků.

2.4 Infrastruktura

Infrastruktura je základním stavebním kamenem každého informačního systému. Správná infrastruktura zajišťuje spolehlivost, výkon a bezpečnost systému. V následujících sekcích jsou popsány některé z nejpoužívanějších technologií a postupů pro správu infrastruktury.

2.4.1 Cloud computing

Cloud computing je model poskytování služeb třetí stranou, který umožňuje přístup k sdíleným zdrojům, jako jsou výpočetní kapacity, úložiště a aplikace. Cloud computing poskytuje řadu výhod, jako jsou nižší náklady na provoz, vyšší flexibilita a škálovatelnost, a zvyšuje efektivitu a bezpečnost informačních systémů.

2.4.2 On premise

On premise je model, kdy jsou všechny zdroje a aplikace provozovány a spravovány v rámci organizace. Tento model poskytuje větší kontrolu nad infrastrukturou a daty, ale vyžaduje větší investice do hardwaru a personálu.

2.4.3 Konternerizace

Konternerizace je technika, která umožňuje zapouzdření aplikací a všech jejich závislostí do kontejnerů, které mohou být snadno nasazený a spuštěny na různých platformách. Konternerizace zvyšuje flexibilitu a škálovatelnost aplikací a zjednodušuje správu infrastruktury. Pro naše potřeby jsme zvolili cloudové řešení, konkrétně platformu DigitalOcean, která poskytuje širokou škálu služeb pro správu infrastruktury, jako jsou virtuální servery, úložiště a síťové služby. DigitalOcean umožňuje snadné nasazení a správu aplikací v cloudu a poskytuje vysoký výkon a spolehlivost. Zároveň jsme se rozhodli využít technologii Kubernetes, která umožňuje efektivní správu a škálování kontejnerizovaných aplikací. Kubernetes poskytuje řadu funkcí pro správu kontejnerů, jako jsou automatické škálování, vyvažování zátěže a obnovu po havárii.

2.5 Bezpečnost

Se stále větší digitalizací roste objem dat a informací, které jsou ukládány a zpracovány v informačních systémech. Zabezpečení těchto systémů je proto klíčovým aspektem, který je třeba brát v potaz již při návrhu a implementaci. V následujících sekcích jsou popsány některé z nejpoužívanějších technik a postupů pro zajištění bezpečnosti informačních systémů.

2.5.1 Autentizace a autorizace

Autentizace a autorizace jsou základními mechanismy pro zajištění bezpečnosti informačních systémů. Autentizace zajišťuje ověření identity uživatele, zatímco autorizace určuje, jaké akce a zdroje má uživatel oprávnění provádět. Pro autentizaci se nejčastěji používají uživatelská jména a hesla, ale existují i další metody jako dvoufaktorová autentizace. Autorizace se obvykle provádí pomocí rolí a oprávnění, které jsou přiřazeny uživatelům na základě jejich role v organizaci. V našem případě se uživatelé rozdělují do dvou rolí, které mají jasně dané oprávnění a přístup k funkcím systému. Administrátoři mají plný přístup ke všem funkcím administračního panelu, zatímco běžní uživatelé pouze k části pro nich určené.

2.5.2 Šifrování

Šifrování je technika, která zajišťuje, že data jsou přenášena a ukládána v bezpečné formě, která je nečitelná pro neautorizované osoby. Existují různé metody šifrování, jako například symetrické a asymetrické šifrování. Symetrické šifrování používá stejný klíč pro šifrování a dešifrování dat, zatímco asymetrické šifrování používá dva klíče, veřejný a soukromý. Veřejný klíč je použit pro šifrování dat, zatímco soukromý klíč je použit pro dešifrování dat. Pro běh našeho systému vyžadujeme TLS, který zajišťuje šifrování dat přenášovaných mezi klientem a serverem. Tímto způsobem zabraňujeme odposlechu a zneužití dat třetími stranami.

2.5.3 Zálohování a obnova dat

V případě havárie nebo ztráty dat je důležité mít zálohovací mechanismy, které umožní obnovu dat a minimalizují výpadek systému. Existují různé metody zálohování, jako jsou pravidelné zálohy na externí úložiště, replikace dat a zrcadlení dat. Důležité je také pravidelně testovat zálohovací mechanismy a zajišťovat, že jsou správně nastaveny a fungují správně. Pro náš systém jsme zvolili pravidelné zálohy dat na externí úložiště, které jsou prováděny automaticky a zajišťují, že v případě havárie nebo ztráty dat můžeme rychle obnovit systém a minimalizovat výpadek.

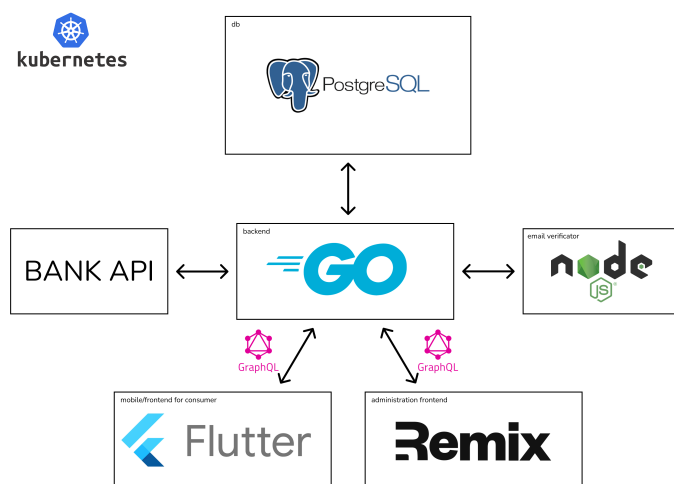
2.5.4 Ochrana proti útokům

Informační systémy jsou často cílem různých typů útoků, jako jsou DDoS útoky, SQL injection, cross-site scripting a další. Pro zajištění bezpečnosti systému je důležité mít mechanismy, které chrání systém před těmito útoky. Existují různé techniky a nástroje, které mohou pomoci chránit systém před útoky, jako jsou firewall, antivirový software, firewall webové aplikace (WAF) a další. Všem typům útoků se snažíme předcházet pomocí různých bezpečnostních opatření, jako je například filtrování vstupních dat, omezení přístupu k citlivým informacím a pravidelné aktualizace nástrojů a knihoven.

3 Praktická část

3.1 Propojení systému a tok dat

Systém Menuroo je navržen s modulární architekturou, která umožňuje efektivní propojení jednotlivých komponent a zajišťuje plynulý tok dat mezi nimi. Následující popis detailněji objasňuje, jak jsou tyto části propojeny a jak data proudí systémem:



Obrázek 3.1: Schéma propojení systému

3.1.1 Komunikace mezi frontendem a backendem

Uživatelská rozhraní, jak pro mobilní aplikace (Flutter), tak pro webové administrátorské rozhraní (Remix), komunikují s backendem prostřednictvím GraphQL API. GraphQL slouží jako jednotný endpoint pro všechny požadavky na data, což umožňuje efektivní a flexibilní získávání informací.

3.1.2 Backend jako centrální uzel

Backend, napsaný v jazyce Go, funguje jako centrální uzel systému. Zpracovává všechny požadavky z frontendů, komunikuje s databází PostgreSQL pro ukládání a načítání dat a integruje se s externími službami, jako je bankovní API a e-mailový verifikátor (Node.js).

3.1.3 Tok dat při objednávce

Při objednávce jídla uživatel provede akci v mobilní aplikaci. Tato akce je odeslána jako GraphQL požadavek na backend. Backend zpracuje požadavek, uloží objednávku do databáze PostgreSQL. Následně backend odešle potvrzení o objednávce zpět na frontend.

3.1.4 Správa dat v databázi

Databáze PostgreSQL slouží jako trvalé úložiště pro všechna data v systému, včetně informací o uživateli, objednávkách, jídelním lístku a platbách. Backend zajišťuje konzistenci a integritu dat pomocí transakcí a validací.

3.1.5 Integrace s externími službami

Systém Menuroo se integruje s externími službami pro rozšíření funkcionality. Komunikace s bankovním API umožňuje automatické zpracování plateb a ověření transakcí. E-mailový verifikátor (Node.js) zajišťuje ověření e-mailových adres uživatelů při registraci a obnově hesla.

3.1.6 Nasazení pomocí Kubernetes

Pro zajištění škálovatelnosti a spolehlivosti je systém nasazen pomocí Kubernetes. Kubernetes umožňuje automatické škálování backendu a dalších komponent systému podle aktuální zátěže.

3.2 Verze systému

Systém Menuroo v současnosti existuje v několika verzích, které se liší svým účelem a funkcionalitou. Každá z nich vznikla na základě specifických požadavků uživatelů a postupně se rozšiřovala o nové možnosti podle reálných potřeb provozu.

3.2.1 Verze Zlatá Štika 1.0

První verze systému Menuroo byla vyvinuta na základě požadavků školy a v průběhu vývoje prošla několika úpravami, jak z hlediska grafického rozhraní, tak i samotné funkcionality. Jejím hlavním cílem je rezervace jídel v restauraci Zlatá Štika, kde je nasazena a stále aktivně využívána.

Systém umožňuje strážníkům dobíjet kredit, který následně využívají k rezervaci jídel. Kredit lze dobít zasláním platby na specifikovaný účet, přičemž přiřazení částky konkrétnímu uživateli probíhá automaticky prostřednictvím bankovního API na základě variabilního symbolu. Tento mechanismus eliminuje nutnost manuálního zpracování plateb a zjednodušuje celý proces správy rezervací.

3.2.2 Testovací verze

Testovací verze je odvozena od verze Zlatá Štika, avšak není nasazena do ostrého provozu. Slouží primárně k ověřování nových funkcí a jejich ladění před případným nasazením do produkčního prostředí. Mezi testované funkcionality patří například možnost objednávat jen určité části menu, implementace deep-linků do strážnické aplikace nebo podpora více restaurací v rámci jednoho systému.

3.2.3 Verze Podunajské Biskupice

Tato verze systému byla přizpůsobena potřebám městské části Bratislavy, Podunajských Biskupic, a představuje zatím nejrozsáhlejší variantu Menuroo. Původní systém rezervací jídel byl rozšířen o funkcionality určené pro administraci rozvozu jídel seniorům. Nově byla implementována funkce generování tiskových sestav, které usnadňují organizaci rozvozu. Patří mezi ně například sestava pro kurýry, která zobrazuje seznam objednaných jídel spolu s adresami v optimálním pořadí doručení, a sestava jídelního lístku, která umožňuje seniorům vybírat jídla v tištěné podobě. Objednávky jsou následně zadávány do systému pracovníky města prostřednictvím administrátorské funkce umožňující přihlášení pod konkrétního uživatele. Platební systém byl v této verzi přizpůsoben specifickým potřebám seniorů – oproti předplacenému kreditu umožňuje uživatelům jít do negativního zůstatku, přičemž vyúčtování probíhá měsíčně formou faktur. Tento přístup je pro starší strážníky pohodlnější a snižuje administrativní zátěž při dobíjení kreditu.

3.2.4 Demo verze

Demo verze slouží výhradně k prezentaci systému Menuroo v rámci soutěží, školních prezentací a dalších příležitostí, kde je potřeba ukázat funkcionality systému bez propojení s reálnými daty. Obsahuje simulovaná data a umožňuje demonstraci klíčových funkcí v kontrolovaném prostředí.

3.3 Popis jednotlivých částí

3.3.1 Aplikační programové rozhraní

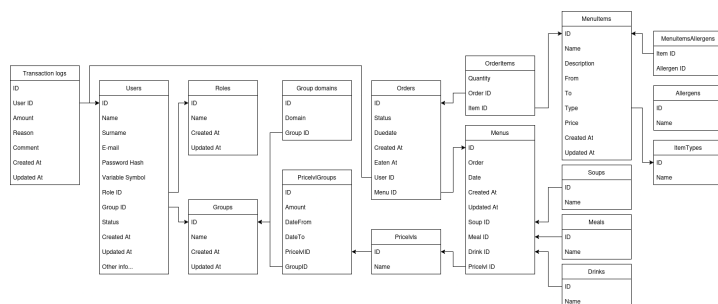
Pro přístup k datům ze serveru se používá aplikační programové rozhraní. API je jediná část systému, která má přístup k databázi. Pokud jiná část systému potřebuje získat data z databáze, API slouží jako prostředník a auditor této komunikace. Všechny požadavky jsou zapisovány do logu. Obsahuje jeden HTTP endpoint, který je pomocí GraphQL schopný plnit nespočet požadovaných funkcí.

Jakmile přijde na nějaký endpoint požadavek, API zkontroluje zda je uživatel autorizován k provedení požadavku. Pokud ano, požadavek se předá do další části systému, která se stará o zpracování požadavku. V další části systému se zavolají všechny potřebné moduly k zpracování požadavku. Moduly jsou navrženy tak, aby na sobě byly co nejvíce nezávislé a mohly být snadno vyměněny nebo rozšířeny. Nejdůležitějším modulem je modul pro správu dat, který zajišťuje ukládání a aktualizaci dat v databázi. Další moduly se starají o různé aspekty systému, jako je napojení na e-mail, banku nebo jiné externí systémy.

Pokud v jakékoli části dojde k chybě, API vrátí chybovou hlášku s odpovídajícím HTTP kódem. Všechny chybové hlášky jsou logovány pro pozdější analýzu.

3.3.2 Databáze

Databáze je nejdůležitější částí systému. Obsahuje všechna data, která jsou potřebná pro správný chod aplikace. V našem případě je databáze PostgreSQL, což je relační databáze, která je vhodná pro ukládání strukturovaných dat. Databáze obsahuje několik tabulek, které obsahují data o uživateli, objednávkách, jídelním lístku a dalších entitách. Všechna data jsou ukládána v normalizované podobě, což zajišťuje konzistenci a integritu dat.



Obrázek 3.2: Databázové schéma

3.3.3 Administrátorské webové rozhraní pro restaurace

Tato část je určena pro použití restauracemi. Umožňuje zadávat jídelní položky na jednotlivé dny, které se propíší do databáze a následně zobrazí strážníkům v jejich aplikaci.

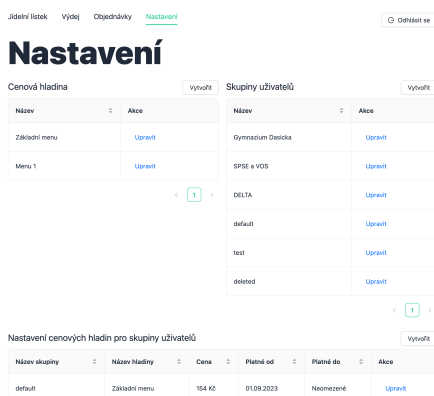
Tuto možnost poskytuje skrz jednoduchý formulář pro zadávání položek s podporou doplňování již dříve zadanych položek pro zvýšení uživatelského komfortu a úsporu času.

Obrázek 3.3: Správa jídelního lístku

Dále nabízí rozhraní přehled všech objednávek, kde restaurace můžou vidět kolik lidí kdy přijde a množství objednaní všech položek. Restauraci díky přehledu mohou zefektivnit jejich nakupování potřebných surovin a snížit jejich plýtvání.

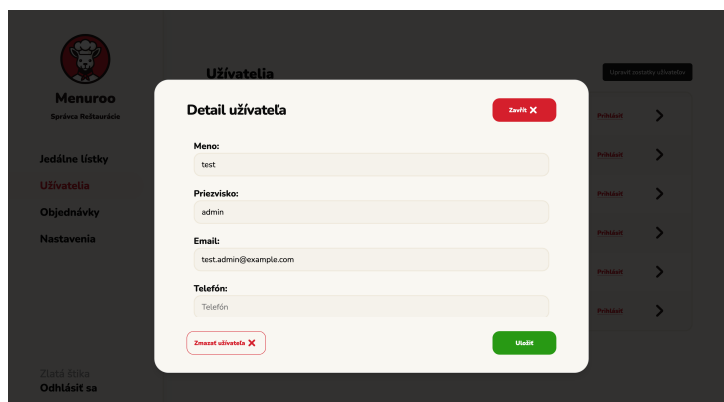
Obrázek 3.4: Přehled objednávek

Administrátorské rozhraní dále umožňuje zakládat skupiny pro uživatele do kterých je následně může restaurace zařadit. Díky této funkcionalitě je efektivně restauracím umožněno vytvářet skupiny pro lidi a zadávat různou cenu pro různé skupiny.



Obrázek 3.5: Přehled nastavení skupin a cenových hladin

V neposlední řadě administrátorské rozhraní poskytuje restauracím možnost správy uživatelů. Restaurace má možnost změnit uživatelská data jako je například jméno uživatele. Toto je v rozhraní zahrnuto z důvodu, že zatím obsluhujeme pouze jednu restauraci, a pokud bychom v budoucnosti expandovali do více restaurací, tato funkcionalita bude smazána z bezpečnostních důvodů.



Obrázek 3.6: Správa uživatele.

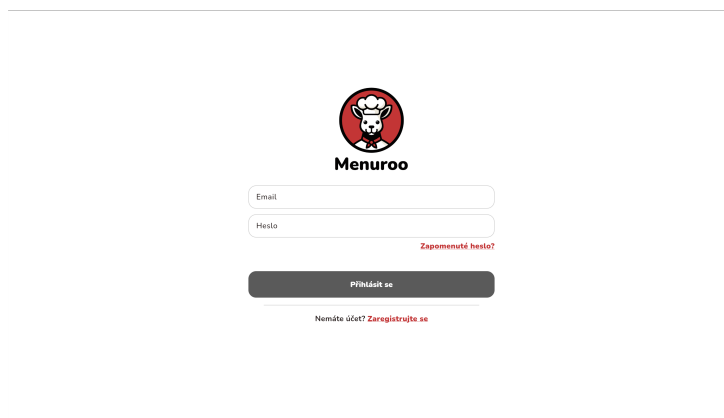
Pro verzi využívanou v Podunajských Biskupicích zde vzniklo několik nových prvků v podobě tiskových sestav. Tyto tiskové sestavy byly navrženy s důrazem na šetření papírem a snahou dostat co nejvíce informací na co nejméně místa a zároveň aby informace byly čitelné.

3.3.4 Mobilní aplikace a webové rozhraní pro strážníky

Tuto část využívají strážníci.

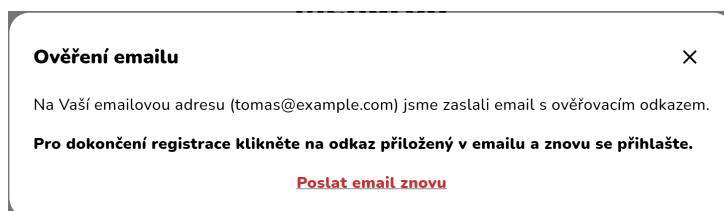
Autentizace uživatele

Pro přístup ke stránce pro objednání jídelních položek je nutná autentizace uživatele. Je zde k dispozici jednoduchý formulář pro přihlášení uživatele či založení nového uživatele.



Obrázek 3.7: Stránka pro přihlášení

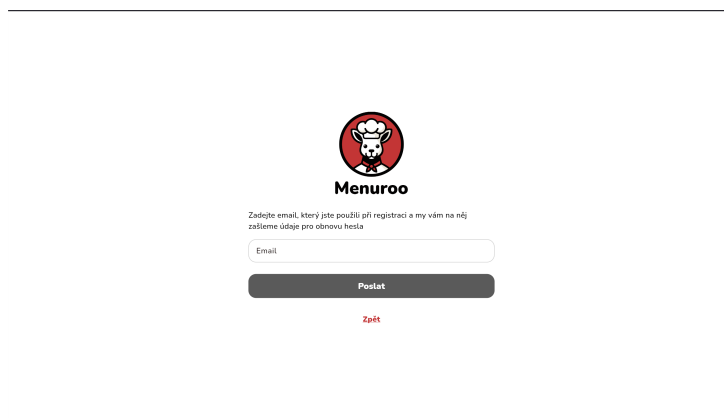
Jakmile si uživatel založí účet, je vyzván k ověření své emailové adresy, kterou do formuláře zadal. Pokud tak neučiní, bude mu odepřen přístup do aplikace. Dialog vyzývající k ověření poskytuje možnost zaslat email znovu. Uživatel obdrží email s URL adresou, která ho přesměruje do služby pro ověření emailu.



Obrázek 3.8: Dialog vyzývající k ověření emailové adresy

Po ověření emailové adresy se uživatel již může do aplikace přihlásit prostřednictvím své přihlašovací emailové adresy a zvoleného hesla.

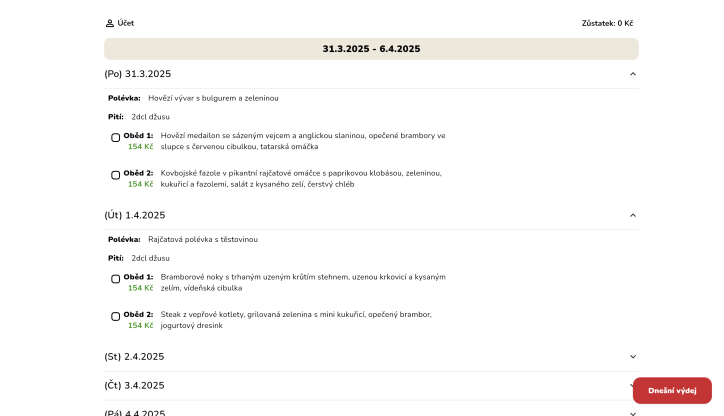
Jestliže uživatel zapomněl své heslo, má možnost požádat o obnovu prostřednictvím funkce v aplikaci. Uživatel je vyzván o zadání přihlašovací emailové adresy, na kterou mu po odeslání žádosti pošleme odkaz, pomocí kterého si může aktualizovat heslo.



Obrázek 3.9: Stránka pro resetování zapomenutého hesla

Zobrazení nabídky restaurace

Po úspěšné autentizaci je uživatel přesměrován na stránku s nabídkou restaurace. Zde vidí nabízené jídelní položky seskupené podle jednotlivých dnů. Všechny informace o denních nabídkách jsou schovány za rozbalovacím tlačítkem, které se nachází u hlaviček každého dne, a načítají se „Lazy“ metodou, tedy pouze tehdy, kdy uživatel kliknutím na toto tlačítko den „rozbalí“. Běžně se uživateli zobrazují dny pouze do příští neděle, nicméně pod posledním zobrazeným dnem se nachází tlačítko, pomocí kterého se zobrazí další týden.



Obrázek 3.10: Hlavní stránka zobrazující nabídku restaurace

Tvorba objednávek

Uživatel si může zvolit položku z nabídky pomocí zaškrtačacího políčka vedle příslušné položky, uživatel může zvolit pouze jednu položku (meníčko) na jeden den na základě domluvy s restaurací. (pozn. ve verzi pro Podunajské Biskupice je možné objednat různé množství jedné položky a více různých položek na jeden den). Jakmile aplikace detekuje změnu ve výběru jídel, zobrazí se pod aplikační lištou výzva k uložení výběru. Po kliknutí na tlačítko uložit se vytvoří, změní nebo smaže objednávka podle změny ve výběru. Pro restaurace poskytujeme předem sjednanou ochrannou lhůtu (současně 2 pracovní dny), během které si již uživatelé nemohou měnit objednávku na den (tzn. Když je pondělí tak si nemůžu změnit objednávku, co mám v úterý, ale středu si změnit můžu)

Aby si uživatel mohl objednat položku na libovolný dostupný den, tj. pracovní den se zadanou nabídkou ze strany restaurace, potřebuje patřičné množství kreditu (pozn. ve verzi pro Podunajské Biskupice je tato funkcionality předělána tak, že uživatel si objednáváním jídel vytváří dluh, který je následně hrazen skrz fakturu vydanou městem). Současný kredit je uživateli zobrazen ve pravé části aplikační lišty jako zůstatek. Tento kredit si může dobít zasláním peněz na námi vlastněný bankovní účet se svým přiřazeným variabilním symbolem. Informace o platbě uživatel zjistí skrz dialog o uživatelských možnostech, který lze otevřít ze stránky s nabídkou restaurace pomocí textového tlačítka v levé části aplikační lišty s nápisem „Účet“ a ikonkou osoby. Tento dialog obsahuje nejenom informace o platbě (číslo bankovního účtu a variabilní symbol), ale také obsahuje jméno přihlášeného profilu v hlavičce dialogu, kontaktní email v případě dotazů nebo připomínek, tlačítko pro odhlášení a tlačítko pro smazání účtu.



Obrázek 3.11: Dialog uživatelských možností

Smazání uživatelského účtu

Pro smazání účtu slouží samostatný dialog, který se vyvolá po stisknutí tlačítka pro smazání účtu v již zmíněném dialogu o uživatelských možnostech. Nachází se zde upozornění o důsledcích vymazání účtu a uživatel je vyzván, aby ověřil svou identitu opětovnou autentizací. Po zadání přihlašovacích údajů je uživateli poskytnuta poslední možnost stornovat výmaz prostřednictvím potvrzujícího dialogu. Pokud uživatel odsouhlasil výmaz, jeho účet je nenávratně smazán.



Smazání účtu ×

Pro smazání účtu se prosím znovu přihlašte.
Zbýlý kredit vám nebude navrácen!

Email

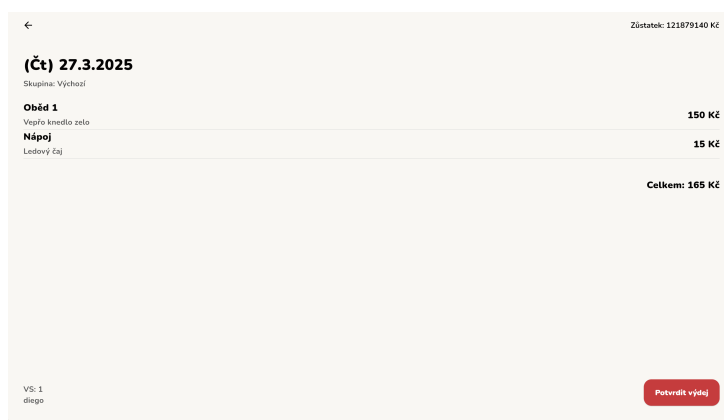
Heslo

Smazat účet

Obrázek 3.12: Dialog pro smazání účtu

Aktivace objednávky

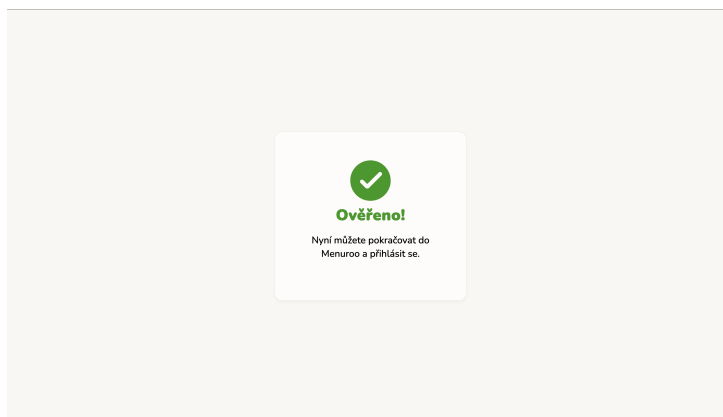
Když uživatel přijde do zapojené restaurace v den, kdy má zadanou objednávku, může prostřednictvím stisknutí plovoucího akčního tlačítka s textem „Dnešní výdej“ být přesunut na stránku pro uplatnění objednávky. Na této stránce se uživateli zobrazí datum, jeho uživatelská skupina a rekapitulace jeho objednávky na příslušný den včetně všech objednaných položek a celkové ceny. V levém spodním rohu stránky se nachází uživatelské informace, jméno a variabilní symbol. Pro aktivaci objednávky slouží plovoucí akční tlačítko obsahující text „Potvrdit výdej“. Po stisknutí tlačítka se zobrazí potvrzující dialog, zda uživatel opravdu chce uplatnit výdej. Po potvrzení dialogu je do systému propsána objednávka jako aktivovaná a zobrazí se tak i v administrátorském rozhraní, pro účely ověření pravosti.



Obrázek 3.13: Stránka pro uplatnění objednávky na den

3.3.5 Služba pro ověření emailu

Jednoduchá webová stránka sloužící pro ověření uživatelského emailu. Po kliknutí na odkázanou URL adresu v emailu, který uživatel obdržel je přesměrován do této služby, která odešle požadavek do aplikačního programového rozhraní s příslušným tokenem z parametru URL. Tento token je uživatelsky specifický. Pokud ověření proběhlo úspěšně, zobrazí se uživateli potvrzení a uživatel je informován, že se může vrátit zpět do aplikace a znovu se přihlásit.



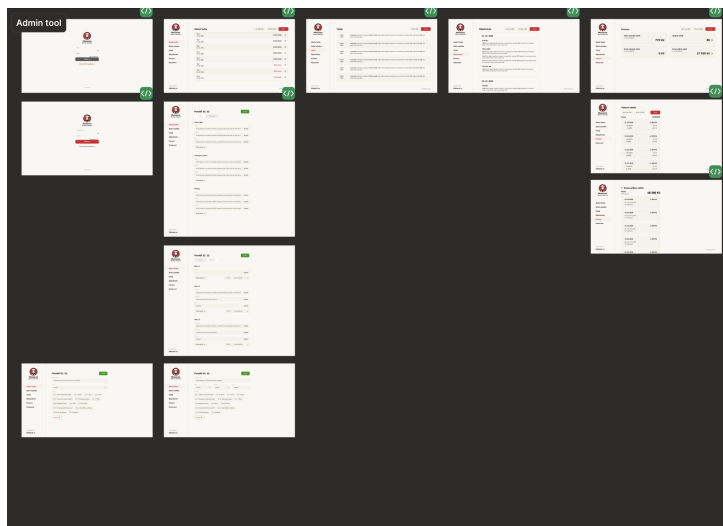
Obrázek 3.14: Stránka potvrzující úspěšné ověření emailu

3.3.6 Jak funguje připisování kreditu na účet

Bankovní účet vedeme u Fio banky, která poskytuje aplikační programové rozhraní ke svému bankovníctví. Prostřednictvím něj jsme schopni získat informace o pohybech na účtu a automaticky připisovat uživatelům kredit na základě variabilního symbolu u příchozích plateb na účet.

3.4 Design

Design systému Menuroo byl klíčovým prvkem pro zajištění intuitivního a uživatelsky přívětivého prostředí. Během celého vývoje jsme kladli důraz na moderní a čistý design, který usnadňuje orientaci a používání aplikace jak pro restaurace, tak pro strávníky.



Obrázek 3.15: UI/UX design administrátorského rozhraní

3.4.1 Nástroj Figma

Pro návrh uživatelského rozhraní jsme zvolili nástroj Figma, který nám umožnil efektivní spolupráci a iterativní design. Figma je výkonný nástroj pro návrh uživatelských rozhraní, který umožňuje vytvářet interaktivní prototypy, sdílet návrhy a spolupracovat v reálném čase.

3.4.2 Designový systém

Při návrhu jsme vytvořili konzistentní designový systém, který zahrnuje definice barev, typografie, ikon a komponent. Tento systém nám umožnil udržet jednotný vizuální styl napříč celou aplikací a zjednodušil vývoj.

3.4.3 Uživatelské rozhraní pro restaurace

Design administrátorského rozhraní pro restaurace byl navržen s ohledem na efektivitu a přehlednost. Klíčové funkce, jako je správa jídelního lístku, přehled objednávek a správa uživatelů, jsou snadno dostupné a intuitivní.

3.4.4 Mobilní aplikace pro strážníky

Design mobilní aplikace pro strážníky byl navržen s důrazem na jednoduchost a rychlost. Strážníci mohou snadno prohlížet nabídku jídel, provádět rezervace a spravovat své objednávky.

3.5 Největší problémy při vývoji

3.5.1 Práce v týmu

Asi největším problémem pro nás byla samotná práce v týmu. Nikdo z nás neměl moc zkušeností s tvorbou informačních systémů, jen jsme měli ponětí o postupu. Ze začátku se děli například věci, kdy jeden člověk pracoval na jedné funkcionalitě a druhý pracoval na úplně odlišné funkcionalitě, ale oba se navzájem domnívali, že pracují na stejné věci. Tento problém se nám podařilo odbourat zvýšením komunikace mezi námi a zavedení vyšší intenzity schůzek s vedoucím projektu.

3.5.2 UI/UX bez návrhu

Když si člověk programuje nějakou aplikaci, má přesnou představu, jak bude vypadat a jak se bude používat, nicméně zapomíná, že tuto aplikaci bude velmi často spíše používat uživatel, který o vývoji nic neví. Tyto situace se nám stávali zejména na začátku, kdy jsme ještě neměli návrh designu, podle kterého se řídit. Ale stávají se dodnes, když se najde nějaká funkcionalita, která by byla skvělá v aplikaci, už si návrh stránky nebo zobrazení funkcionality děláme opět sami. Tento problém řešíme prostřednictvím vzájemných kontrol kódu a funkcionality.

3.5.3 App Store

Původně aplikace pro strážníky měla být opravdu jen aplikace. Nicméně aby mohla být použita i na mobilních telefonech s operačním systémem iOS, musela by být vysazena na obchodě App Store. Toto začal být problém v raných fázích vývoje pro školu, vzhledem k počtu uživatelů, co mají telefony od Apple. U Android telefonů toto překážku nevytváří, vzhledem k otevřenější povaze operačního systému. Takže tedy toto způsobilo to, že uživatelé s operačním systémem Android normálně používali systém Menuroo, zatímco uživatelé s operačním systémem iOS používali stále metodu samolepek na lístcích. Tento problém by vyřešilo dostání aplikace na App Store. Bohužel tento úkol není tak jednoduchý, jak se může zdát. Apple má velice přísné podmínky pro aplikace, které vpouští do svého proprietárního obchodu App Store a kontrolní proces aplikace trvá několik dní po odeslání na kontrolu. Aplikaci kontrolují do té chvíle, než narazí na nějakou chybu, která zamezuje ve vstupu aplikace do obchodu App Store. Proces tedy může vypadat následovně: odešlete aplikaci na kontrolu, získáte zpětnou vazbu, že informace x je špatně, opravíte informaci x, odešlete aplikaci znovu na kontrolu, získáte zpětnou vazbu, že informace x už je sice správně ale informace y je špatně. Tento problém byl vyřešen vysazením aplikace na webovou stránku, takže Android uživatelé mají staženou aplikaci, zatímco uživatelé s iPhonem používají systém skrz webové rozhraní.

3.5.4 UTC časy

Poměrně zásadním problémem se pro nás staly UTC časy, na kterých běží naše databáze. Tohle ovšem přetrvávalo pouze 3 měsíce po začátku problému, jelikož jsme si s tím zvládli v kódu poradit.

3.6 Businessová stránka projektu

S projektem jsme se účastnili několika akcí a konzultací. Navštěvovali jsme pro konzultace v Pardubickém podnikatelském inkubátoru, kde jsme získali cenné byznysové rady. Zúčastnili jsme se akce pořádané Elektrárnami Opatovice, kde jsme od elektráren získali finanční podporu na částečné pokrytí nákladů běhu serverů. Na zahraniční stáži na Slovensku jsme pracovali na vývoji verze 2.0 pro restauraci Zlatá Štika, a dokonce jsme získali poptávku od starosty městské části Podunajské Biskupice pro vytvoření verze k rozvozu pro seniory. S projektem jsme se zúčastnili Festivalu vědy a techniky v Pardubickém kraji a umístili se v krajském kole na 3. místě v oboru „Technické a inženýrské vědy“. Za celý projekt jsme v týmu získali ocenění Mladý Talent Pardubického kraje 2024 v technickém IT oboru.

4 Závěr

V tomto projektu jsme představili Menuroo, komplexní systém pro správu objednávek a rezervací jídel v restauracích. Naším cílem bylo vytvořit nástroj, který zefektivní provoz restaurací a zároveň zlepší zkušenost strávníků. Přistoupili jsme k vývoji systému s důrazem na komplexní přístup, od analýzy potřeb uživatelů, přes návrh a implementaci jednotlivých modulů, až po finální nasazení v reálném prostředí.

Analýza současných procesů v gastronomii odhalila potřebu efektivnějšího řešení pro správu objednávek a rezervací. Naše řešení, Menuroo, integruje klíčové funkce jako správu jídelního lístku, online rezervace, správu plateb a generování statistik do jednoho intuitivního systému. Zavedení automatizace a digitalizace těchto procesů umožňuje restauracím snížit administrativní zátěž a minimalizovat plýtvání surovinami.

Vývoj webového a mobilního rozhraní byl klíčovým krokem k zpřístupnění našeho nástroje širokému publiku. Integrací všech funkcí do uživatelsky přívětivého prostředí jsme umožnili restauracím i strávníkům snadno využívat všechny výhody systému Menuroo.

Závěrem lze říci, že Menuroo představuje pokus o modernizaci a digitalizaci gastronomického průmyslu. Zjednodušením a automatizací klíčových procesů si Menuroo klade za cíl být podpůrným nástrojem, který obohacuje provoz restaurací a zlepšuje zkušenost strávníků. Je to krok směrem k efektivnějšímu a udržitelnějšímu gastronomickému prostředí, kde technologie a inovace otevírají nové možnosti pro restaurace i jejich zákazníky.

Literatura

- [1] Kenneth C. Laudon and Jane P. Laudon. *Management Information Systems: Managing the Digital Firm*. Pearson, 17th edition, 2021.
- [2] Wikipedie. Softwarová architektura. *Wikipedie, otevřená encyklopedie*.

Seznam obrázků

1.1	Logo Menuroo	3
3.1	Schéma propojení systému	12
3.2	Databázové schéma	15
3.3	Správa jídelního lístku	16
3.4	Přehled objednávek	16
3.5	Přehled nastavení skupin a cenových hladin	17
3.6	Správa uživatele.	17
3.7	Stránka pro přihlášení	18
3.8	Dialog vyzívající k ověření emailové adresy	18
3.9	Stránka pro resetování zapomenutého hesla	19
3.10	Hlavní stránka zobrazující nabídku restaurace	19
3.11	Dialog uživatelských možností	20
3.12	Dialog pro smazání účtu	21
3.13	Stránka pro uplatnění objednávky na den	22
3.14	Stránka potvrzující úspěšné ověření emailu	23
3.15	UI/UX design administrátorského rozhraní	24

Zadání maturitního projektu z informatických předmětů

Jméno a příjmení: *Michal Vaniš, Tomáš Kalhous*

Pro školní rok: *2024/2025*

Třída: *4. B*

Obor: *Informační technologie 18-20-M/01*

Téma práce: *Menuroo – systém pro rezervaci obědů (administrační část)*

Vedoucí práce: *Ing. Jiří Formánek*

Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:

Cíl projektu:

Vytvořit Menuroo - komplexní systém pro nabídku, objednávání a placení jídel mezi libovolným počtem restaurací a strávníků. V rámci maturitního projektu bude pouze v českém jazyce a jako měna bude využívána pouze CZK. Skládá se z několika částí:

- Propagační webová stránka se základními informacemi o systému pro strávníky i restaurace, která umožňuje založení účtu restauraci i strávníkovi.
- Část pro restaurace, která zahrnuje:
 - o Administraci uživatelů v rámci restaurace: číšník, provozní, majitel,
 - o Administraci jídelního lístku (po položkách i možnost komplexního menu),
 - o Přehled objednávek strávníků,
 - o Administraci skupin strávníků a tvorbu cenových hladin pro různé skupiny strávníků,
 - o Vyúčtování s Menuroo a dále:
 - o „číšnickou část“, která bude realizována mobilní aplikací nebo responzivní webovou stránkou, ve které bude moci číšník zkontrolovat potvrzení „odebrání stravy“ strávníkem,
- Část pro strávníky bude realizována mobilní aplikací nebo responzivním webem a bude umožňovat:
 - o objednání jednotlivých položek jídla nebo celého menu ze zvolených restaurací,
 - o zrušení objednávky před uplynutím „ochranné lhůty“,
 - o zobrazení odebraných obědů za zvolené období,
 - o zobrazení provedených plateb za zvolené období,
 - o dobítí kreditu na účtu strávníka (bude navázáno na bankovníctví),

- zrušení účtu s převedením zbylého kreditu zpět na bankovní účet, ze kterého uživatel poslal poslední platbu.

Systém poběží na veřejné webové adrese. Provozně bude vyřešena možnost nasazování dílčích nových verzí jednotlivých komponent.

Rozdělení zodpovědnosti v týmu:

- Michal Vaniš je zodpovědný za frontend administrátorské části aplikace a správné volání end-pointů a dále dev-ops,
- Tomáš Kalhous je zodpovědný za frontend mobilní aplikace/responzivního webu strážníka a mobilní aplikace/responzivního webu číšníka včetně správného volání end-pointů,

Dále v tým pracují:

- Matyáš Tázlar (nemá to jako maturitní projekt tento rok) je zodpovědný za správné fungování backend aplikace – všech end-pointů a databáze,
- Marek Hlava (nemá to jako maturitní projekt) je zodpovědný za vytvoření propagační webové stránky projektu (a případně obchod) a
- Matěj Tobiáš Moravec je zodpovědný za grafický návrh front-endu

Specifikace projektu:

1. Analýza a návrh:

- Bude provedena analýza existujících firem v oblasti gastrotech a jejich produktů a analýza nabídky školních stravovacích systémů,
- Bude provedena analýza existujících technologií na řešení dílčích úloh (mobilní aplikace, webový informační systém, správa verzí a jejich nasazování, automatizované testování, rozesílání e-mailů, generování .pdf, komunikace s bankovním účtem,...)
- Budou definovány všechny požadavky na systém, zpracovány pomocí UseCase diagramů a vytvořen návrh uživatelského rozhraní.
- Bude navržena odpovídající databázová struktura.

2. Vývoj uživatelského rozhraní:

- Bude vytvořeno přehledné a intuitivní webové rozhraní, které uživatelům umožní snadný přístup ke všem funkcím aplikace.

3. Technické požadavky:

- Systém bude navržen v takové architektuře, která umožní jeho snadné rozšíření a změny.
- Systém bude umožňovat nasazování dílčích nových verzí bez nutnosti dlouhého maintanance módu.
- Budou implementována bezpečnostní opatření pro ochranu uživatelských dat.

Požadované výstupy:

- Funkční propagační webová stránka včetně uživatelské dokumentace z hlediska restaurace a strávnicka.
- Funkční část pro administraci restaurace.
- Funkční část pro číšníka.
- Funkční část pro strávnicka.
- Vývojářská dokumentace.
- Prezentace a případně demonstrační video.

Hodnocení:

Projekt bude hodnocen na základě následujících kritérií:

- Kvalita a funkčnost uživatelského rozhraní.
- Úroveň dokumentace a prezentace projektu.
- Kreativita a efektivnost řešení.

Stručný časový harmonogram (s daty a konkretizovanými úkoly):

- **Září:** Analýza problému, tvorba UML usecase diagramů, návrh vzhledu aplikace,
- **Říjen:** Vývoj front-endu stravnické aplikace (možnost objednání po dílčích částech jídla a z více restaurací, přehled plateb a odebraných obědů), vývoj front-endu správcovské aplikace (možnost zadání dílčích jídel i celých menu, administrace uživatelů, skupin strávnicků, cenových hladin,...)
- **Listopad:** Napojování front-endu na endpointy back-endu
- **Prosinec:** Finalizace projektu, debugging
- **Leden - březen:** Práce na dokumentaci