SWI – semestrální projekt

Richard Bohunovský, BOH0162

26. dubna 2024

Obsah

Seznam diagramů

1 Zadání projektu

1.1 Problém

Popis problému:
Rozvozci v pizzerii musí počítat kolik objednávek za den rozvezli a kolik hotovosti mají odevzdat účetní. Jenže nemají žádný jiný systém, než počítat denní tržbu z papírových účtenek, které se často ztrácejí a rozvozci musí účtenku opsat ručně. Dále také nevíme, na jaké adrese jaký rozvozce byl a je těžké ohlídat, jestli rozvozci nejezdí jinými auty než mají, nebo nejezdí v jednom autě pospolu.

Další problém je také, že storno objednávky musí udělat manažer pobočky z počítače, takže když zrovna není přítomný na pobočce, storna objednávek se kupí.

1.2 Řešení

Řešením tohoto problému by byla aplikace pro rozvozce, kde by viděli objednávky, které přišly do pizzerie. Sami by si poté vybrali jaké objednávky chtějí odvézt a v aplikaci, která je napojena na navigaci by si mohli naplánovat trasu s více objednávkami. U těchto objednávek by mohli měnit status, který objednávka má (v přípravě, doručuje se, atd.).

V nejlepším případě by se mohlo udělat i rozhraní pro zákazníky, kde by též viděli status objednávky, ale to není součástí zadání.

1.3 Zadání od zákazníka

Nový systém by měl navazovat na starý systém a měl by s ním správně komunikovat. Starý systém již má databázi objednávek a k té by se měl ten nový připojit a rozšířit dokončenou objednávku o informaci, kdo ji rozvezl.

1.4 Rozdělení zadání do úseků

1.4.1 Začátek směny

Na začátku dne, kdy rozvozce přijde do práce se přihlásí do aplikace a klikne na nějaké tlačítko, které mu zobrazí jeho QR code, který lze naskenovat do Frekru(evidence docházky). Jakmile se rozvozce označí ve Frekru, začne se mu odpočítávat čas strávený v práci a je mu přiděleno identifikační číslo ve formátu Rn, kde n $\in \mathbb{Z}$, pod kterým se bude identifikovat pro další rozvozce ten pracovní den.

1.4.2 Výběr objednávky

Po přihlášení do práce, rozvozce klikne na seznam objednávek, který bude seřazen od té, která přišla nejdříve až po tu, která přišla, jako poslední.

Každá objednávka v seznamu bude obsahovat: Čas přijetí objednávky, celkovou cenu, adresu, jméno a telefon zákazníka a způsob platby. Jestliže si nějakou objednávku bude chtít rozvozce vybrat, klikne na tlačítko 'Přidat na trasu' –> tím se u objednávky připíše identifikační číslo rozvozce, aby ostatní rozvozáci viděli, že objednávka je již zabraná.

V aplikaci bude aktuální čas, který se bude neustále aktualizovat, aby rozvozáci věděli, jak moc mají s objednávkou spěchat, popřípadě změnit pořadí (každá objednávka by měla být doručena do hodiny. Možná by v aplikaci mohl být odpočet, kolik zbývá času do uplynutí 1 hodiny (čas by byl zobrazen černě) od přijmutí objednávky. Pokud by uplynul delší čas od příjmutí objednávky, než 1 hodina, čas by byl zobrazen červeně).

1.4.3 Průběh rozvozu

Tím, že rozvozce klikl na přidat adresu na trasu –> adresa z objednávky se přidá na seznam objednávek a Google API vypočítají nejlepší (nejkratší) cestu, jakou lze všechny adresy na seznamu objet. Jakmile bude mít rozvozce všechny objednávky, se kterými chce vyrazit na rozvoz, zaklikne 'Začít rozvoz' po tomto již nebude možné přidávat do trasy další objednávky a náhled aplikace bude mapa (navigace).

Současně se všechny objednávky v seznamu označí 'v přepravě'. Bude ale možné po přijetí na adresu označit objednávku 'Doručena', nebo když si zákazník objednávku nepřevezme 'stornována'.

1.4.4 Konec směny

Na konci pracovního dne si stejně, jako na začátku rozvozce zobrazí QR code, který opět naskenuje do Frekru a rozvozci se vypočítá výplata, kterou za daný den dostane podle toho, kolik minut odpracoval od zakliknutí příchodu, až po zakliknutí odchodu. Poté už záleží na domluvě, zda se rozvozce vyplácí ihned, nebo až za odpracovaný měsíc.

Po zadání odchodu se také spočítá denní tržba, kolik má rozvozce odevzdat z karet, hotovosti, stravenek atd. Žádné papírové účtenky již nejsou potřeba.

Na konci bude pěkný výpis:
Rozvozce: R1
Jméno: Richard Bohunovský
SPZ auta: — —-
Pobočka: Ostrava
Celková denní tržba: ——,-
Tržba z karet: —-,-
Tržba ze stravenek: —-,-
Odvod za pohonné hmoty: —,-
Odvod za nákup: —-,-
Výplata za dnešní den: —-,-
Odevzdávaná hotovost: —-,-

1.4.5 Neobvyklé situace

Pokud rozvozce bude mít nehodu, nebo se stane nějaká neočekávaná událost, musí existovat funkce 'vrátit všechny objednávky do fronty', aby bylo možné se v objednávkách zorientovat.

Objednávku bude muset jít stornovat přímo od rozvozce, aby storna nemusel řešit manažer pobočky. U každého storna je ale nutno uvést důvod proč byla stornována, aby rozvozce schválně nestornoval objednávku a peníze si nechal.

2 Systémové požadavky FURPS model

2.1 Functionality – F

FR1 Aplikace musí využívat GPS.

FR2 Aplikace musí být napojena na databázi objednávek.

FR3 Aplikace musí umožnit rozvozci vidět objednávky z databáze.

FR4 Aplikace musí umožnit změnu označení 'v přepravě', 'čeká na doručovatele', 'v přípravě'.

FR5 Aplikace musí umět udělat storno objednávky z databáze (aby na storno nebylo potřeba volat někoho jiného).

2.2 Usability – U

UR1 Aplikace musí mít noční režim.

UR2 Aplikace musí být jednoduchá ve smyslu ovládání, ideálně jedna hlavní pracovní plocha s objednávkami, aby rozvozci při navigaci nevyskakovali pop-up okna atd.

UR3 Aplikace musí používat barvy k označení stavu objednávky.

UR4 Aplikace musí podporovat vložení adresy pomocí hlasu.

UR5 Aplikace musí mít dostupnou nápovědu 'návod na ovládání'.

2.3 Reliability – R

RR1 Aplikace se bude obnovovat jednou za 20 sekund.

RR2 Systém musí být zabezpečený proti ztrátě dat, nebo vložení corrupted data ze staré databáze.

RR3 Systém si musí udržovat naposledy načtená data i po výpadku internetového připojení. Po výpadku internetu se nesmí zobrazit jen 'bílá obrazovka'.

RR4 Systém musí mít implementovaný návrat k poslednímu funkčnímu stavu (restart stav).

2.4 Performance – P

PR1 Systém musí být mobile device friendly –> nevyužívat mnoho baterie při práci na pozadí.

PR2 Systém musí být škálovatelný, např. přidání nové provozovny se stejným systémem.

PR3 Systém musí mít maximální dobu odezvy na server 1s.

PR4 Data ze systému se musí recyklovat, například odchod zaměstnance, zaměstnanec se vymaže.

2.5 Supportability – S

SR1 Aplikace musí mít dokumentaci.

SR2 Error feedback od uživatelů, v případě chyby.

SR3 Pravidelné zálohování dat.

3 Charakteristika postav a prostředí

3.1 Uživaté používající systém

- **Rozvozce**: Zaměstnanec, který bude aplikaci využívat nejvíce. Může přistupovat k datům z databáze a měnit status objednávky. Poté může vytvářet své záznamy o příchodu a odchodu z práce. Využívá aplikaci i jako navigaci a mapy, sestavuje seznam adres, které na trase pojede.
- **Manažer**: Manažer dědí z rozvozce všechny práva, které rozvozce má. Zaměstnanec, který může mazat záznamy jakéhokoli rozvozce o příchodu a odchodu z práce, nebo je upravovat. Může nastavovat zaměstnancovu hodinovou odměnu mazat auta ze systému, která byla zničena, nebo poškozena, přeřazovat auta od jednoho rozvozce k druhému. Dále dostává výpis storno objednávek z každé pobočky pizzerie. Tento report bude obsahovat i kolik pobočka vydělala za jeden den a další informace.

Jedině manažer, nebo admin může role aktérů měnit.

3.2 Prostředí ve kterém se aplikace využívá

Jelikož je způsob, jakým pizerie dělá denní tržby a přehledy velice zastarlý a rozvozce si musí hlídat, aby účtenku nikde neztratil, je navržena digitalizace. Tento systém velice pomůže automatizovat proces ukončení pracovního dne, zpřehlední proces jak pro rozvozce, tak i pro účetní a manažery.

Toto rozhodnutí s sebou nese komplikace, že každý rozvozce u sebe musí mít moderní mobilní zařízení, nebo mu musí být zapůjčeno. Otázkou je, jestli s mobilním telefon bude umět zacházet, takže se u pohovoru musí uvést nutná podmínka: schopnost umět zacházet s mobilním telefonem.

U této aplikace nemusí existovat PC verze, protože ta již existuje ve starém systému. Systém bude napojen na starý systém, takže je nutné nový se starým provázat.

4 Use case diagram

Přiložený use case diagram, znázorněný v diagramu číslo 1, zobrazuje přístupy Šéfa(manažera) a Rozvozce v aplikaci, nezabývá se věcmi ve staré systému, jako jsou: přijetí objednávky a akce týkající se zákazníků. Diagram zobrazuje především akce, které může udělat rozvozce.

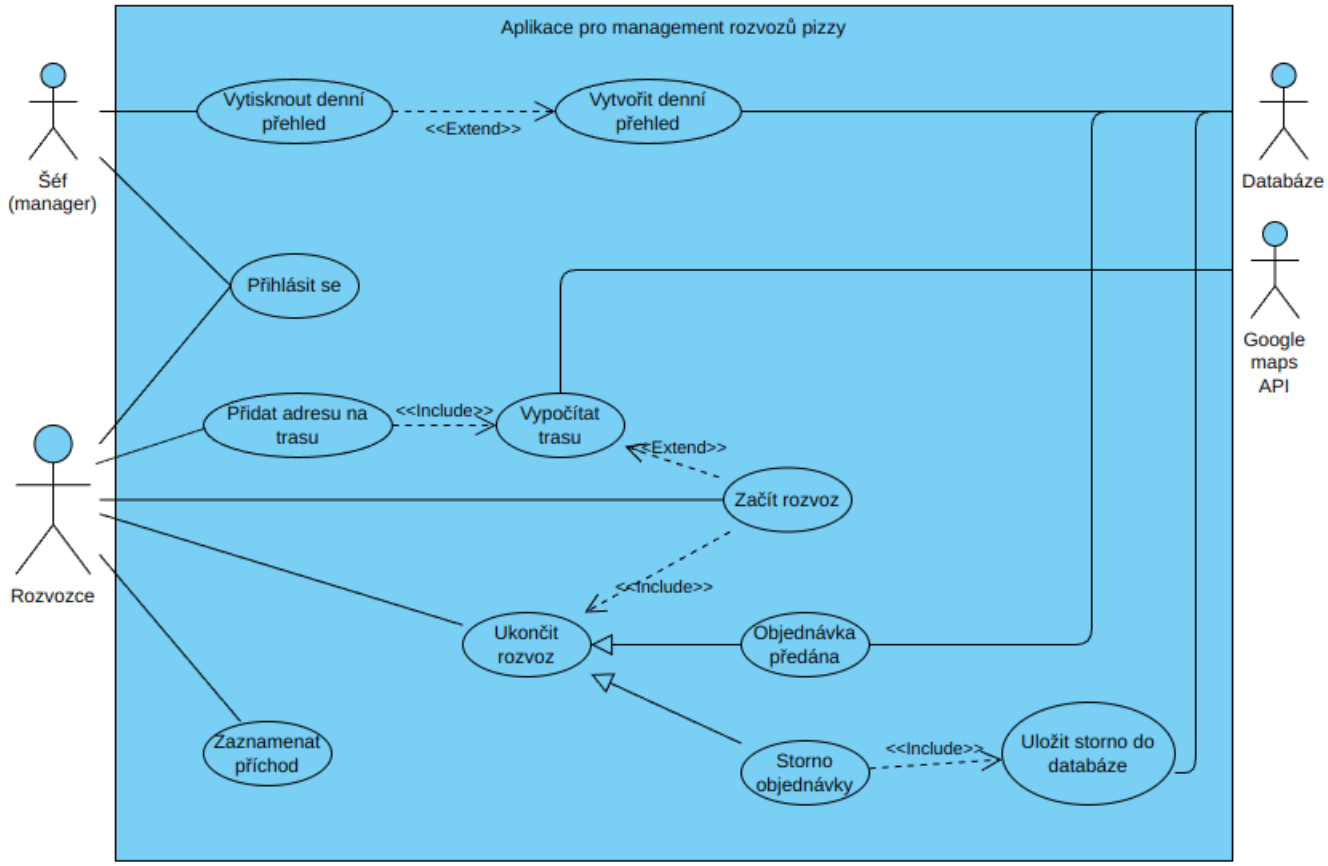

Diagram 1: Use case diagram

4.1 Use case scenarios

4.1.1 Shorter use case

Title: Zaznamenání odchodu
Actors: Rozvozce
Trigger: Rozvozce klikne na tlačítko v aplikaci.
Pre-condition: Rozvozce musí být přihlášen a musí mít zaznamenaný příchod.
Post-condition: Rozvozci musí být vypočítána mzda.
Minimal guaranties: V aplikaci vyskočí pop-up okno s vypočítanou mzdou.
Main success:

1. Uživatel zaznamená odchod.
2. Čas se uloží do aplikace.
3. Zaměstnancovi je vypočítána mzda a vyskočí pop-up okno.

Extension scenarios:

1a.: Čas příchodu je stejný, jako čas odchodu.

1.a1 - Uživateli je sděleno, že čas, který zadal není v souladu s jeho pracovní dobou.

1.a2 - Čas se neuloží do databáze.

1.a3 - Zaměstnancovi není vypočítána mzda a pop-up okno nevyskočí.

Alternative scenarios:

3a.: Zaměstnancova hodinová sazba je 0,-

3.a1 - Účetní, nebo managerovi se odešle výzva k nastavení mzdy.

3.a2 - Use case pokračuje dál, s výslednou mzdou 0,-

4.1.2 Complex use case

Title: Přidání adresy na trasu
Actors: Rozvozce
Trigger: Rozvozce zadá adresu rozvozu.
Pre-condition: Rozvozce musí být přihlášen.
Minimal guaranties: V aplikaci se adresa přidá do seznamu adres.
Main success:

1. Rozvozce zadá adresu rozvozu.
2. Vypočítá se nejlepší možná trasa.
3. Seznam adres je přeuspořádán vzhledem k nejrychlejšímu času doručení.
4. Use-case se opakuje do té doby, než uživatel klikne na začít rozvoz.

Extension scenarios:

1a.: Rozvozce zadal stejnou adresu, kterou již v seznamu má.

1.a1 - Uživateli vyskočí info a zeptá se, jestli adresu chce přidat podruhé.
1.a1.YES - Pokračuje se v use case 2. krokem.
1.a1.NO - Pokračuje se v use case 1. krokem.

Alternative scenarios:

1b.: Rozvozce zadal neplatnou adresu, která neexistuje.

- 1.b1 - Uživateli vyskočí upozornění, že adresa není platná.
- 1.b2 - Uživateli se nabídne korekce adresy, pokud existuje.
- 1.b3 - Use case pokračuje 1. krokem.

1c.: Rozvozce zadal neúplnou adresu. (chybí číslo popisné)

- 1.c1 - Uživateli vyskočí upozornění, že adresa není úplná.
- 1.c2 - Uživateli se nabídne korekce adresy, pokud existuje.
- 1.c3 - Use case pokračuje 1. krokem.

2a.: Není možné vypočítat trasu. (lesy, polní cesty atd..)

- 2.a1 - Uživateli vyskočí upozornění, že není možné vypočítat trasu, ale je mu ukázána mapa s možností "drop a pin to closest road"a rozvozce tak místo adresy zadá souřadnice.
- 2.a2 - Use case pokračuje 2. krokem.

3a.: Rozvozce chce pořadí upravit podle sebe

- 3.a1 - Rozvozci je umožněno přeuspořádat si adresy podle sebe.
- 3.a2 - Use case pokračuje 4. krokem, ale již bez 3. kroku.

5 Activity diagram

Tento diagram aktivit, diagram s číslem 2, popisuje proces zadávání adres do seznamu, podle kterého rozvozce bude objednávky rozvážet.

Na začátku se bude čekat na signál, což v tomto případě je kliknutí do vyhledávače na mapě a psaní adresy, nebo hlasové zadání adresy.

Po začátku vyhledávání se vytvoří objekt 'adresa', která je následně analyzována, jestli je korektní, nechybí číslo popisné, nebo zda adresa existuje. Jestliže je adresa nevalidní, vyhledávač navrhne možnosti opravy.

Až bude adresa validní, přidá se na seznam adres na trase. V tomto bodě si uživatel může vybrat, zda bude zadávat další adresu, nebo začne rozvážet.

Pokud klikne na možnost začít rozvážet, výběr se ukončí a navigace začne. Odešle se signál, že rozvoz již začal pro další aktivitní diagramy.

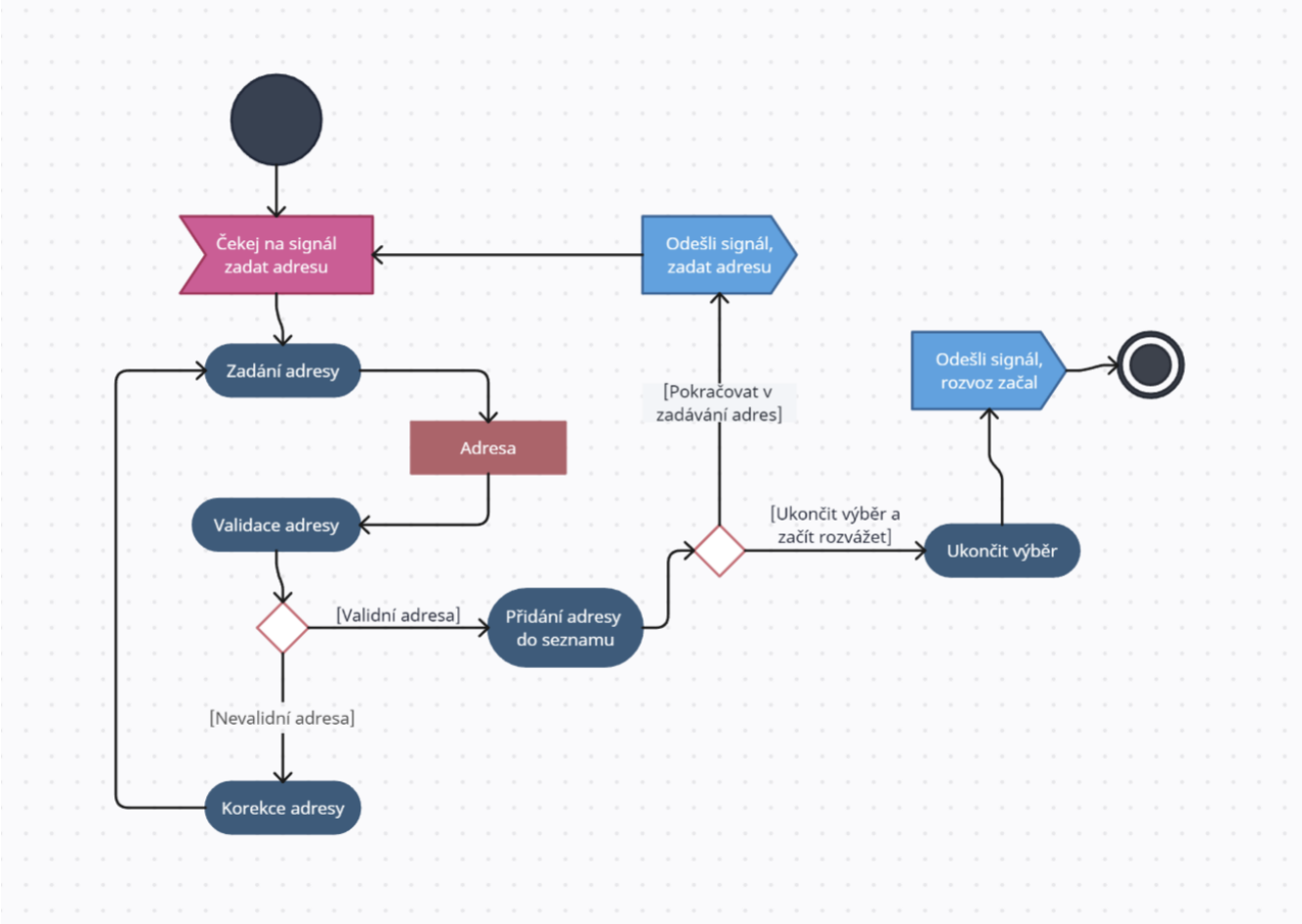

Diagram 2: Activity diagram

6 Class diagram

Následující třídní diagram, diagram číslo 3, znázorňuje strukturu tříd, které na sebe v mém systému navazují. Pro lepší představu jsou zde doplněny i třídy, které se přímo netýkají nového systémů. Například třída Customer a třída Order.

6.1 Vysvětlení tříd v diagramu a zmínění některých důležitých informací o třídách.

Car
Třída, která zastupuje rozvozácké auto, tato třída obsahuje spz a datum konce dálniční známky.

DeliveryMan
Třída, která zastupuje samotného rozvozce, jako člověka. Jako číslo Rn, jak bylo zmíněno v požadavcích 1.4.1 výše na straně 3 je uloženo v dictionary ve třídě Branch, která není součástí tohoto diagramu. Třída obsahuje *jméno* rozvozce, jeho *auto*, jestli mu není přiděleno, bude obsahovat nullptr, *trasu*, kterou bude mít vždy právě jednu, jestliže nebude mít žádné objednávky, trasu bude mít stále, ale bude prázdná, *hodinovou sazbu* a *seznam příchodů a odchodů do práce*.

ShiftBegginingEnd
Třída, která obsahuje datum a čas příchodu a odchodu do práce. Funkce IsLeavingTimeMarked() a IsArrivalTimeMarked() vrací bool hodnotu, na základě toho, jestli je vyplněno datum i čas. Při přidání nového příchodu se musí zkontrolovat, jestli rozvozce nemá dva záznamy stejného typu (Arrival, nebo Leaving).

Route
Třída, která slouží, jako spojovník rozvozce a objednávky. Do této třídy rozvozce uloží objednávky, které chce rozvézt, Google API podle této třídy vypočítají vzdálenost a nejlepší cestu pro rozvoz.

Order
Třída, která je již navržena ve starém systému a pro náš systém není důležitá z hlediska návrhu, takže v diagramu jsou popsány jen klíčové funkce.

Customer
Třída, která je též navržena ve starém systému a zde jsou jen nastíněny některé její funkce.

Status
Třída, která je přidána ke starému systému za účelem rozlišování různých stavů objednávky. Každý stav má přiřazeno jedinečné id, podle toho, v jakém stavu se zrovna nachází. 1:'Nepřijata', 2:'Přijata do systému', 3:'V přípravě', 4:'Čeká na rozvozce', 5:'V přepravě', 6:'Doručena', -1:'Nepřevzata', 10:'Převzata na pobočce', 0:'Stornována'. Tři neobvyklé koncové stavy jsou označeny speciálními čísly 0, -1, 10, tyto stavy společně s číslem 6 mají instanci delivered nastavenou na true. Jakmile je instance delivered nastavena na true, objednávka je definitivně zpracována.

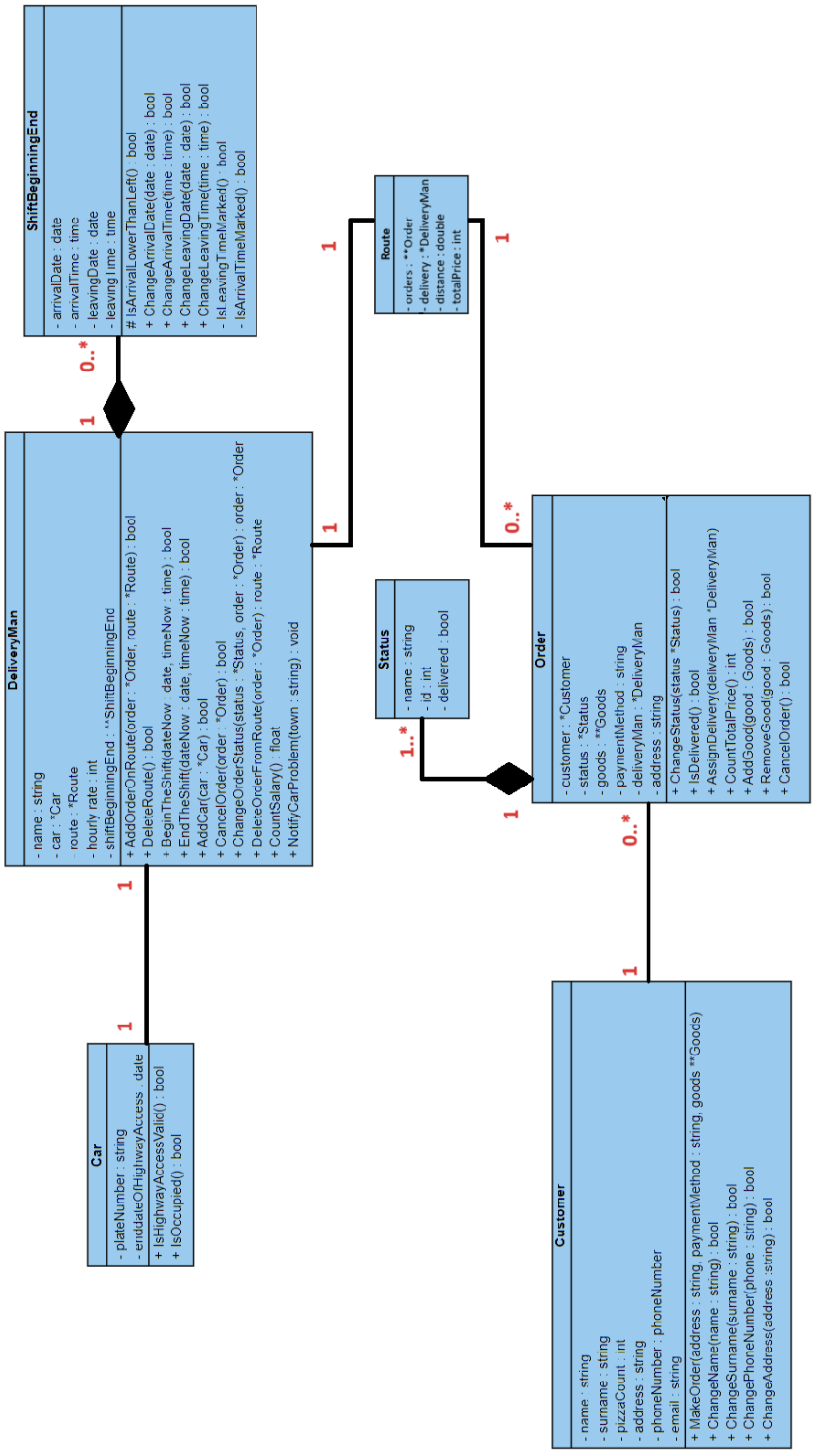

Diagram 3: Class diagram

7 Sequence diagram

Sequenční diagram, který vidíme níže 4 popisuje proces na straně zákazníka a vyobrazuje, jak se zachází s naší nově implementovanou třídou 'Status'.
První krok v našem diagramu je vytvoření objednávky, ta se vytvoří pokud je uživatel přihlášen, ale i pokud není.

Pokud uživatel přihlášen není, objednávce je přiřazen status číslo 1 'nepřijata'.

Pokud uživatel je přihlášen, objednávce je přiřazen status 2 'přijata'. Po tomto přiřazení se na objednávce začíná pracovat na provozovně. Jakmile je objednávka hotova, hledá se rozvozce, který by ji rozvezl.

Pokud je nějaký rozvozce volný a vybere si objednávku, objednávce je přiřazen status číslo 5 'v přepravě'. Pokud žádný rozvozce volný není, objednávka nabývá statusu číslo 4 'čeká se na rozvozce'

A nakonec se status, kterého objednávka nabyla odešle zpět zákazníkovi.

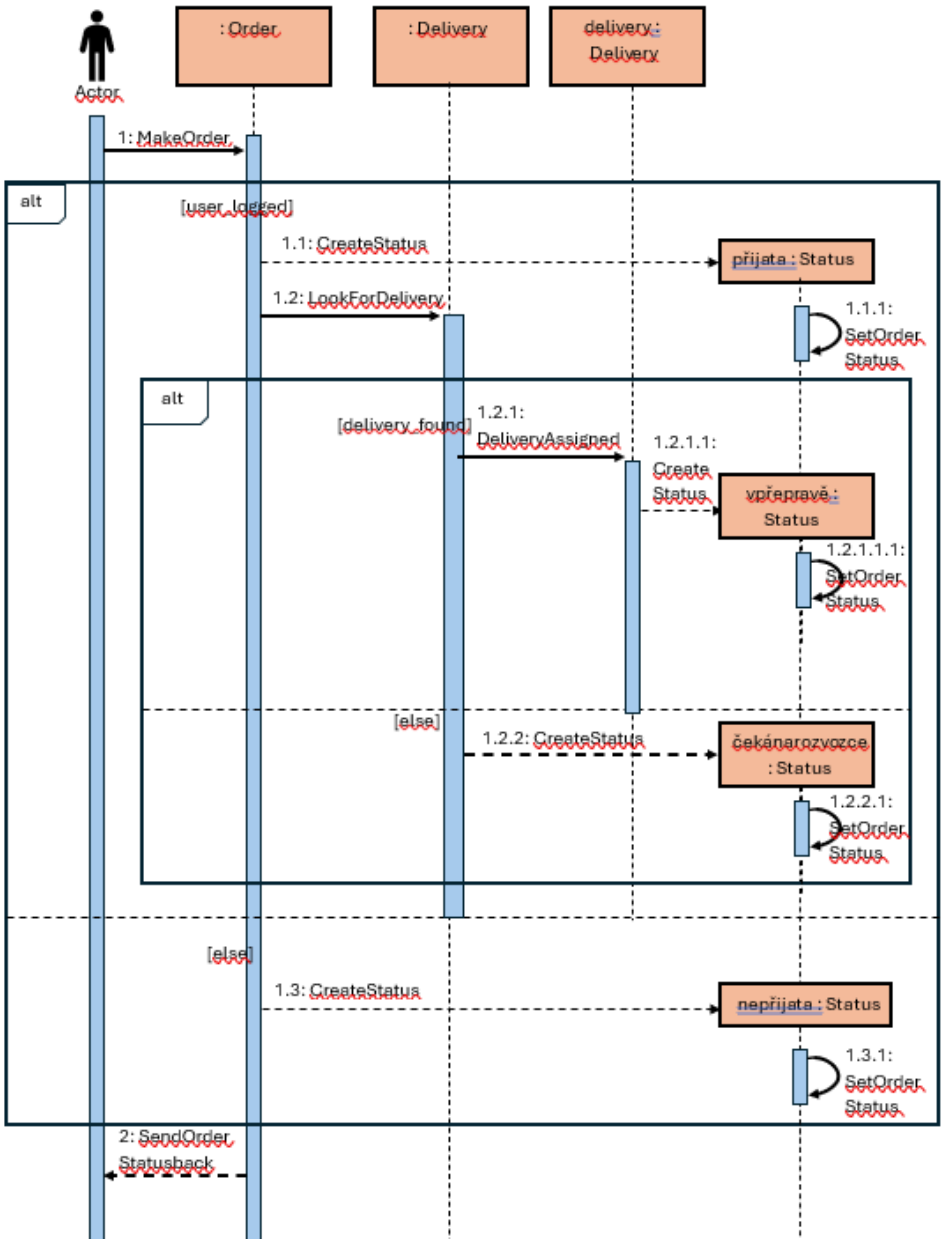

Diagram 4: Sequence diagram