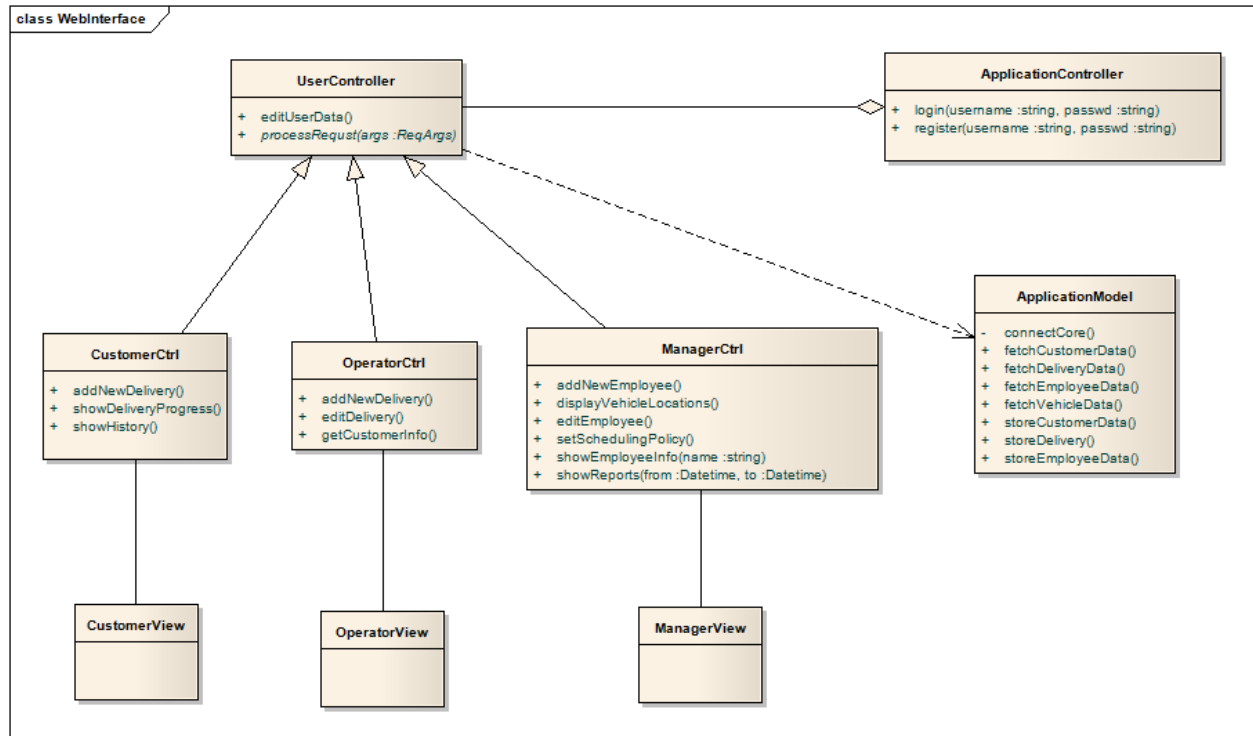


Cargo Wizard - minispecifikace komponent, návrh rozhraní

Web Application



Webová aplikace umožňuje uživatelům komunikaci se systémem. Je postavena na návrhovém vzoru model-view-controller. Skrze jednotlivé pohledy poskytuje uživatelům data podle jejich rolí:

Nepřihlášený uživatel:

- přihlášení
- registrace zákazníka

Zákazník:

- zadání nové objednávky
- zobrazení průběhu doručení
- zobrazení historie objednávek

Operátor:

- zadání nové objednávky pro zákazníka
- zobrazení a editace objednávky zákazníka
- zobrazení informací o zákazníkovi

Manažer:

- zobrazení reportů o provozu společnosti
- zobrazení informací o zaměstnanci
- přidání a editace informací o zaměstnanci
- zobrazení aktuální polohy vozidel

Webová aplikace přistupuje k informacím uloženým v databázi nepřímo přes komponentu Core.

On-board Application

On-board aplikace komunikuje na jedné straně s koncovým uživatelem (řidičem) prostřednictvím uživatelského rozhraní a na druhé straně s Core systémem. Platforma na níž poběží On-board aplikace je Android verze alespoň 4.0.

Controller

Hlavní komponentou aplikace je controller. Jeho prováděné úkoly jsou:

- Komunikace s Core aplikací
- Příjem notifikací od Core aplikace
- Příjem dat z GPS modulu
- Zobrazování dostupných dat v GUI

Výměna dat mezi Core a On-board aplikací

Komunikaci je potřeba provádět oběma směry. Vzhledem k nestálému charakteru připojení k internetu On-board aplikace je nutné zohlednit, že může docházet k častým přerušením spojení.

Core aplikace by zde měla vystupovat v roli serveru, pokud tedy bude chtít On-board aplikace odeslat nějaká data, bude to realizovat voláním veřejného API Core aplikace. Přenos dat zde bude využívat protokol HTTP a datovým formátem bude JSON. Pro autentifikaci se bude využívat protokol OAuth.

Typickým voláním tohoto API může být například odeslání informace o předání zásilky zákazníkovi nebo požadavek na stažení všech dat pro dané zařízení (např. po připojení do sítě).

Komunikace druhým směrem bude probíhat formou push notifikací ze strany serveru (Core aplikace) ke klientovi (On-board aplikace), narhovaná konkrétní implementace je Google Cloud Messaging for Android (<https://developer.android.com/google/gcm/index.html>).

Typickým případem komunikace tímto směrem je např. notifikace o přeplánování současné trasy.

GUI

GUI část bude používat standardní ovládací prvky prostředí Android. Pro přenos dat mezi GUI a controllerem by mohl být využit návrhoý vzor MVVM, konkrétní volba je přenechána SW architektovi a programátorům On-board aplikace, kteří by měli zohlednit vyspělost jednotlivých technik a zvolit tu, se kterou mají největší zkušenosti vzhledem k tomu, že rozsah prací v GUI je poměrně malý a nevyplatí se tak studovat nové technologie.

Core System

Úkolem Core je fungovat jako backend pro celý systém. Core propojuje ostatní komponenty a poskytuje jim přístup k databázi. Skládá se sice z řady modulů, ovšem jako runtime komponenty se v core systému vyskytují pouze `core::Controller` a `core::Scheduler`.

Controller

Controller řídí logiku core systému, přijímá požadavky z webového rozhraní, deleguje je na plánovač. Hotové plány předkládá modulu RouteFinder pro nalazení skutečné trasy. Tuto trasu poté prostřednictvím OnboardControlleru posílá do zařízení ve vozidlech. Zároveň ukládá do databáze všechna data, která během procesu vznikla.

Scheduler

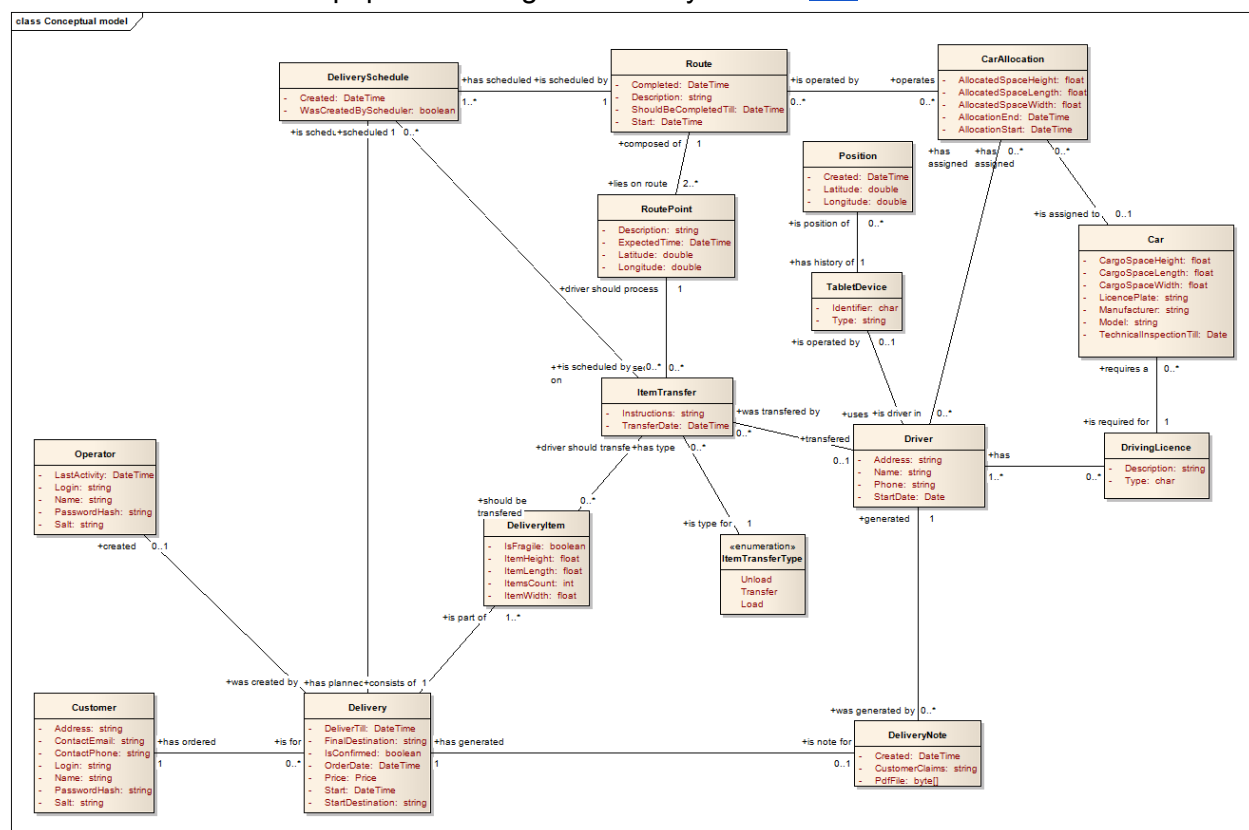
Scheduler je samostatná komponenta, která se stará o plánování zakázek a alokaci vozidel a řidičů. Scheduler je parametrizovaný plánovací politikou (scheduling policy), která ovlivňuje rozhodování plánovače při sestavování plánu. Možné politiky jsou:

- CustomerFirstPolicy - upřednostňuje včasné vyřešení zakázky před efektivitou
- EfficientPolicy - snaží se seskupovat zakázky, pokud je to možné
- OptimalPolicy - snaží se o vyvážení obou předchozích přístupů

Rozhraní zmíněných komponent je popsáno v dokumentu o Architektuře systému:

<http://cargowizard.github.io/CargoWizard/architecture.html>

- základní návrh tabulek popsán na diagramu "Entity model" [zde](#)



- přístup k databázi využívá webové rozhraní (přímo zákazník nebo operátor na telefonu),
onBoard App

- základní součásti databáze

1. Data
2. Triggery
 - Vytvoření zákazníka
 - Zaregistrování objednávky
 - Naplánování přepravy
 - Přijmutí nového zaměstnance
 - Dokončení objednávky
3. Základní pohledy
 - Výpis aktivních a nevyřízených objednávek
 - Výpis naplánovaných cest včetně zobrazení časové rezervy pro možné přidání průjezdního bodu
 - Měsíční výpis aktivity práce zaměstnanců
 - Informace o klientovi

Více informací v částech [Development View](#) a [Data model](#)