



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

REQUIREMENTS AND DESIGN DOCUMENT

COS 301

Team

Marcus Bornman *u15024522*
Vernon Francis *u15185703*
Leonardo Ianigro *u16036094*
Myron Ouyang *u16008368*
Keagan Seddon *u14017823*
Michael Yatrakos *u15015531*

By submitting this document we confirm that we have read and are aware of the University of Pretoria's policy on academic dishonesty and plagiarism and we declare that the work submitted in this assignment is our own as delimited by the mentioned policies. We explicitly declare that no parts of this assignment have been copied from current or previous students' work or any other sources (including the internet), whether copyrighted or not. We understand that we will be subjected to disciplinary actions should it be found that the work we submit here does not comply with the said policies.

May 11, 2018

Contents

| | | |
|----------|---|----------|
| 1 | Introduction | 2 |
| 1.1 | Purpose | 2 |
| 1.2 | Scope | 2 |
| 2 | Overall Description | 2 |
| 2.1 | Product Perspective | 2 |
| 2.1.1 | User Interfaces | 2 |
| 2.1.2 | Hardware Interfaces | 2 |
| 2.1.3 | Software Interfaces | 2 |
| 2.1.4 | Memory | 3 |
| 2.2 | Product Functions | 3 |
| 2.3 | User Characteristics | 3 |
| 3 | Specific Requirements | 3 |
| 3.1 | External Interface Requirements | 3 |
| 3.1.1 | User Interface Requirements | 3 |
| 3.2 | Functional Requirements | 4 |
| 3.3 | Performance Requirements | 4 |
| 4 | Architecture | 4 |
| 4.1 | Domain Diagram | 4 |
| 4.2 | Deployment Diagram | 4 |
| 4.3 | Package Diagram | 4 |

1 Introduction

1.1 Purpose

The purpose of this document is to provide a comprehensive guide on the overall requirements of general architecture of the system.

1.2 Scope

- Name: Carpool
- Functions and Limitations:
 - The system will provide a platform for communication between commuters with similar travel paths
 - It will attempt to provide as much security amongst users as possible, however it cannot guarantee complete safety for its users.
- Product uses:
 - Create a community amongst travelers
 - Simplify trip organization
 - Provide a platform for carpooling

2 Overall Description

2.1 Product Perspective

2.1.1 User Interfaces

- Landing Page
A simple page to allow the user to choose between logging in and signing up.
- Login Page
A simple page with two text boxes where the user will enter their email and password to enter the application
- Dashboard Page
This page will be the homepage that will allow the user to navigate to the rest of the app. It will default to the routes tab, which shows the pending routes the user has active.

2.1.2 Hardware Interfaces

As a progressive web application it will have an almost seamless transition between almost any "smart" device (desktop computers, smart phones, tablets, etc.), from the users perspective of the hardware.

However, on the back-end we will have series of servers running, in order to reduce downtime of the servers and also to ensure that service and data is not lost, or at least that the loss is minimized, if a server goes down for some reason. The deployment diagram (Fig:2) shows a better view on how the hardware will interact, it can be found in section 4.2.

2.1.3 Software Interfaces

The application will be able to connect to the servers through most web-browsers, but will have the option to be downloaded as a progressive web application. This will create a shortcut on the users home-screen as well as install a service worker which will then communicate with the server in a much cleaner fashion, and also allow some off-line features with caching.

The server will communicate with the database using mongoose. The package diagram (Fig :3) contains a more detailed view, it can be found in section 4.3.

2.1.4 Memory

This being a progressive web application, the static components of the web application will be stored in device memory. The dynamic data (data which needs to be constantly retrieved and checked against the server) will be cached for a fixed period of time on the device cache storage to allow limited offline usage of the system.

2.2 Product Functions

A high level summary of the product functionality

- Allow users to sign up
- Allow users to log out
- Authenticate users, based on supporting documentation
- Rate user(s) after trips
- View other users profiles
- Each user should have a "security level" based on multiple factors including: overall rating and security checks completed
- Set up a route
- Match similar routes
- Make use of artificial intelligence to recommend commuters with similar routes
- Allow commuters to communicate with each other
- Provide limited offline functionality

2.3 User Characteristics

This application is designed for general users with either no means of transport between destinations or users with means of transportation and is willing to assist others. The minimum requirements for the users is:

- They must be able to operate a smart phone
- They must be a South African citizen

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interface Requirements

- Adjust for different screen sizes
- Store the static components of the webpage
- Cache the dynamic components of the webpage for a limited period of time
- Easy to navigate
- Responsive

3.2 Functional Requirements

- Allow users to sign up
- Allow users to log out
- Allow users to upload identifying documentation
- Rate user(s) after trips
- View other users profiles
- Set up a route
- Match similar routes
- Make use of artificial intelligence to recommend commuters with similar routes
- Allow commuters to communicate with each other
- Provide metadata about trips
- Provide limited offline functionality

3.3 Performance Requirements

- Provide a sense of trust and security amongst users
- System must be responsive during navigation between components
- System must be easily scalable
- System must be maintainable
- System must be portable

4 Architecture

4.1 Domain Diagram

Fig:[1] The system will be heavily focused on the user as most of the functionality will need the user in some way. The background and review system will be how we establish trust between our users. The route and trip subsystem will provide the main functionality of the system as it'll contain all details about the trip including metadata about the trip such as ETAs. There will also be a simple messaging subsystem in order for users to communicate about their upcoming trip.

4.2 Deployment Diagram

Fig:[2] We are mostly going to use a client server architecture. This will make it possible to see other users data and routes even if their device is offline as it will be centralised. This allows most of the processing to be done server side making the speed consistent across all devices.

The server and database will consist of a trio of each to help minimize downtime and data loss in the case of a server failure.

4.3 Package Diagram

Fig:[3] This is just a general overview of how the systems of our project will work together. The interesting part of this is that we're using a progressive web app.

Being a PWA this: Will allow the user to download an extreme lightweight application shell onto their device Will also allow the app to function as if it has been installed on the device with features such as push notifications and background sync.

Very powerful because it allows the app to be used across a large range of devices very easily with little overhead on the devices themselves

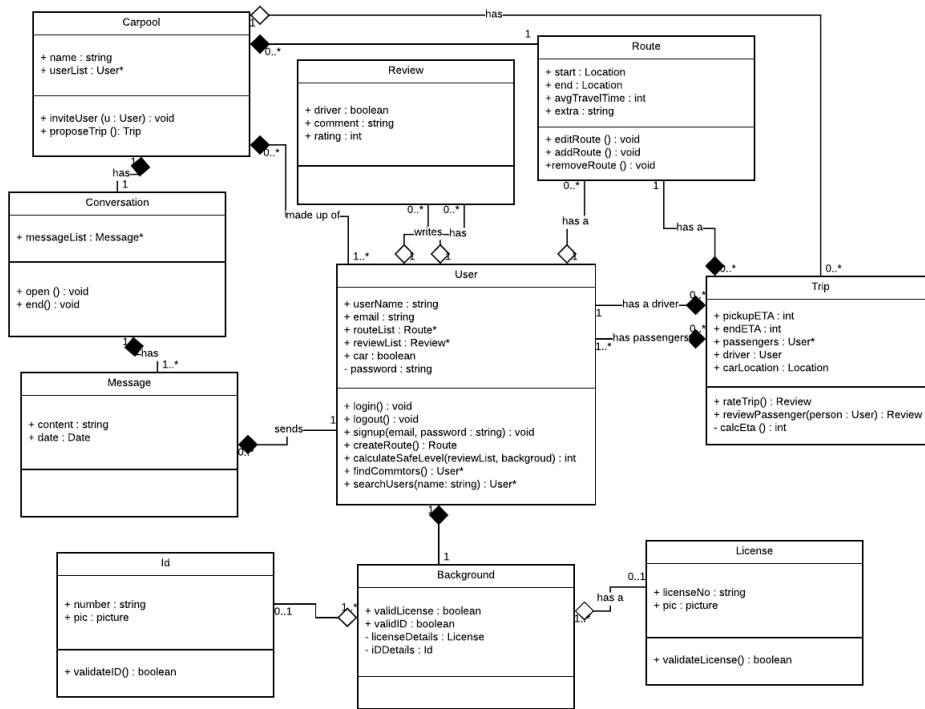


Figure 1: UML Domain Diagram for the Carpool System

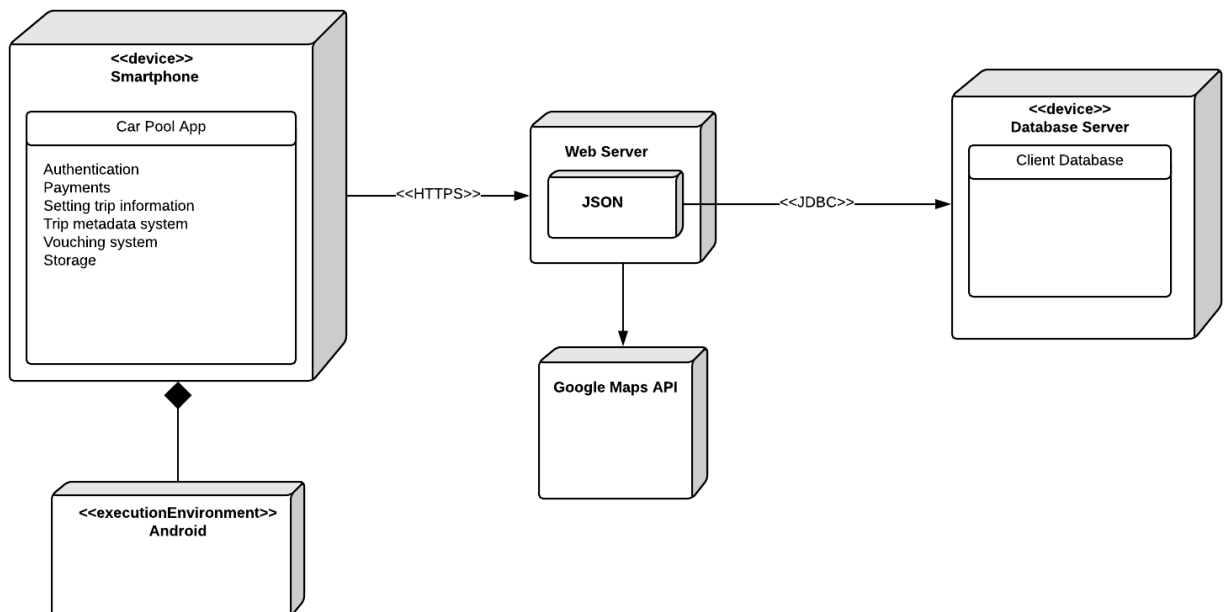


Figure 2: UML Deployment Diagram for the Carpool System

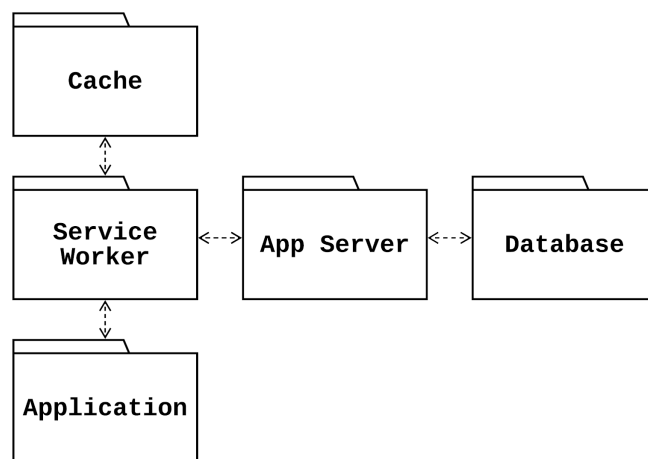


Figure 3: UML Deployment Diagram for the Carpool System