Bilkent University
Department of Computer Engineering

Senior Design Project

CarBuds

HIGH LEVEL DESIGN REPORT

Ahmet Emre Nas	21402357
Doğukan Altay	21400627
Ali Osman Çetin	21302483
Aras Heper	21302248

Supervisor: Uğur Doğrusöz

Jury Members: Ercüment Çicek, H. Altay Güvenir

Innovation Expert: Doğukan Şengül

March 19, 2018

Table of Contents

1. INTRODUCTION

In Universities, students create networks in order to solve their problems collectively. The problems can vary in terms of buying used/old books from other students or exchange stuff and most of the time sharing a ride. Most of these networks established in internet environment which not mean to use for a specific problem of students. Thus, Student can have a hard time to find help or another student fast and precise. Finding a ride or sharing your car with another person rises several issues, the most important one is the security. Inside the university campus most of the drivers can be sure that the hitchhiker most probably another student of the same school. However, outside the campus no one can assure that the hitchhiker even a student. To solve this problem students reach each other through un-specified networks and this arises another issue, time.

CarBuds is a mobile application that any university student can register to reach hitchhikers or can be hitchhiker to find a ride to a specific location. The application offers to find a car or a hitchhiker very quickly while assuring the validity of the other person's whether he or she a student of the proposed university. Therefore, with the help of the Carbuds students can reach each other for sharing a ride in a fast and secure way.

1.1. PURPOSE OF THE SYSTEM

Carbuds is a mobile application that allows users to hitchhike to their desired destination both as a driver or as a hitchhiker. The target audience of the application is university students. Significant amount of university students are not residents of their university's dormitories, therefore most of the students that lives outside the campus, they require proper transportation services. Although these services are provided by most of the universities, their range and diversity to the city suburbs are limited. So, most of the students travel with the hitchhiking method.

Carbuds provide the service for matching the hitchhikers and the drivers before they go on the road. Each user can specify their preferences for the trips and choses their candidate hitchhikers or drivers. Carbuds offer a matchmaking services that shows to users some candidate options for their trip, an instant private messaging environment in order to communicate with the matched users and a route offering system that shows possible beneficial stops in their preferred route.

1.2. Design Goals

- The system should hold the users data (schedule, personal details, ratings) .
- The system should be defining the hubs, or sub networks where users should be grouped up with the other users sharing the similar paths, according to users wishes.
- The system should request an authentication process before initiating with users account.
- The system should provide a further authentication service to sign in before the users use the system, if they log off.
- The system should provide search and matching tools to search from the all users using the system.
- The system should allow users to rate and give a written feedback to other users.
- The system matches a driver with a hitchhiker and give both appropriate notifications.
- Matched users can talk to each other via using built-in messaging system.
- Both hitchhikers and drivers can specify their preferences for the trip in order to increase the satisfaction of both parties.
- Users will be chosen their location and destination point using Google Maps API.
- The system offers each user a set of drivers/hitchhikers that generated according to their preferences, by swiping right or left users can show their willingness or not.
- All users will be authenticated with their school email addresses in order to ensure all

users are university students.

- The system will give brief information about the traffic rules inside the university's campus in order to assure that driver is aware of the restrictions in the respective university.

Usability

- The interface should be user-friendly, simple and straightforward.
- The user interface should be intractable.
- Internet connection is required for using the application.

Efficiency

- The matching algorithm should be fast and efficient in order to give responses to multiple users.
- The route finder algorithm should be fast for calculating routes for users.

Extensibility

- The system should allow addition of future feature developments and should be up to date for each new Android.

Reliability

- The matching of the users should be precise with their preferences.

Compatibility

- The mobile application must be compatible with the most of the android phones in the market.

1.3. Definitions, acronyms, and abbreviations

MySQL: Widely used, an open source database managment system.

SHA256: A hashing algorithm that used for encripting data in one direction, mostly used for password storing.

HTTP: HTTP is the underlying protocol used by the World Wide Web and this protocol defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.

TCP/IP: Transmission Control Protocol/Internet Protocol, is a suite of communication protocols used to interconnect network devices on the internet.

2. CURRENT SYSTEM

In the recent years usage of car sharing apps increased significantly. All these applications aimed difference purposed. For instance, in USA only cars with at least 4 passengers can use left line and therefore most of the apps used in USA aims to find enough people for using left line. Lyft Line [1] is one of the example that used in America for this purpose.

In Turkey apps like BlaBlaCar [2] used as an alternative to travel between cities. With apps like BlaBlaCar people find drivers with same destination city and same start city and split the price of fuel.

Both of these applications do not aim a specific target user and everyone can use them as long as passengers accepted to pay some kind of payments or split fuel expense.

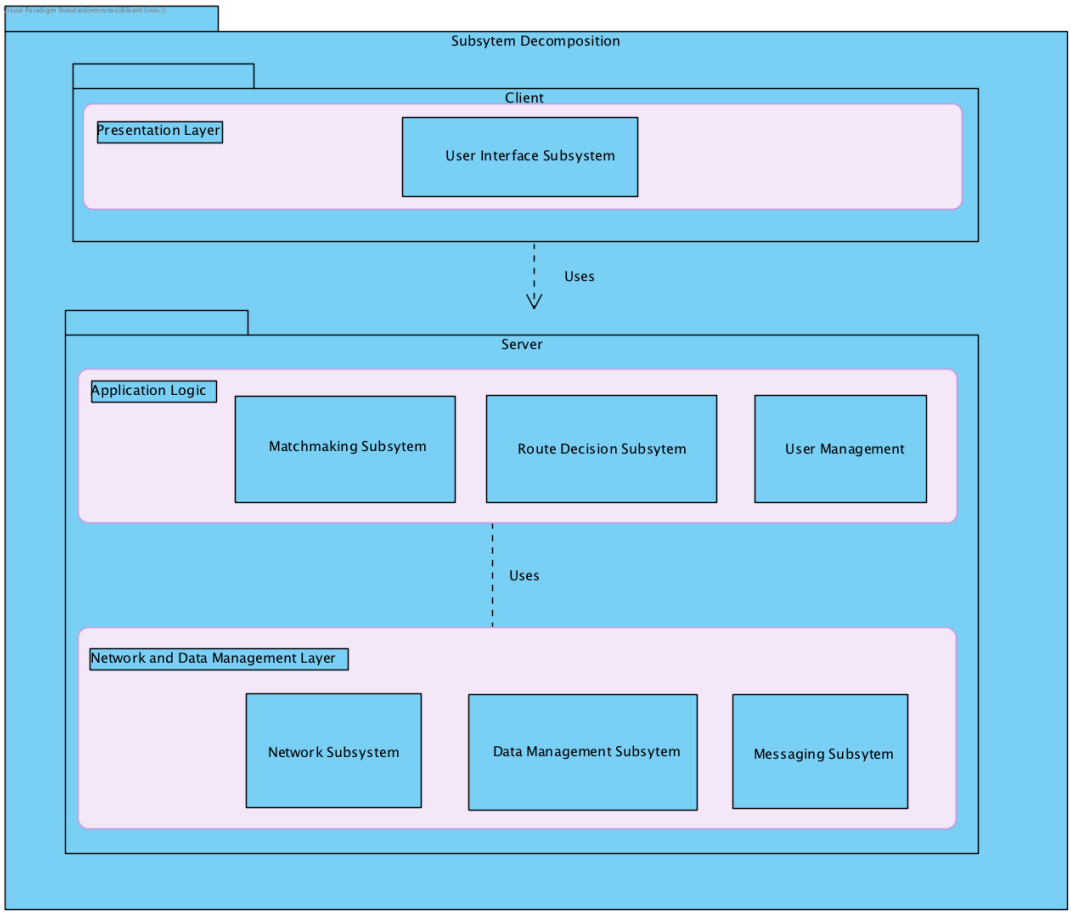
3. PROPOSED SOFTWARE ARCHITECTURE

3.1. OVERVIEW

CarBuds will be mobile application, aims to connect people who study in same universities to go their school together. In order to match users and chatting online back-end must be built on top of a server. Therefore, most suitable architecture is client-server. Subsystem decomposition is discussed in section 3.2. The relationship between the client and server is discussed in section 3.3 Hardware Software Mapping.

3.2. SUBSYSTEM DECOMPOSITION

3.2.1. PRESENTATION LAYER



Presentation layer contains user interface subsystem. This is the layer which user sees in the program. Therefore, presentation layer is the part that interacts with the user. This is client side of the program it interacts with the server according to users' interactions.

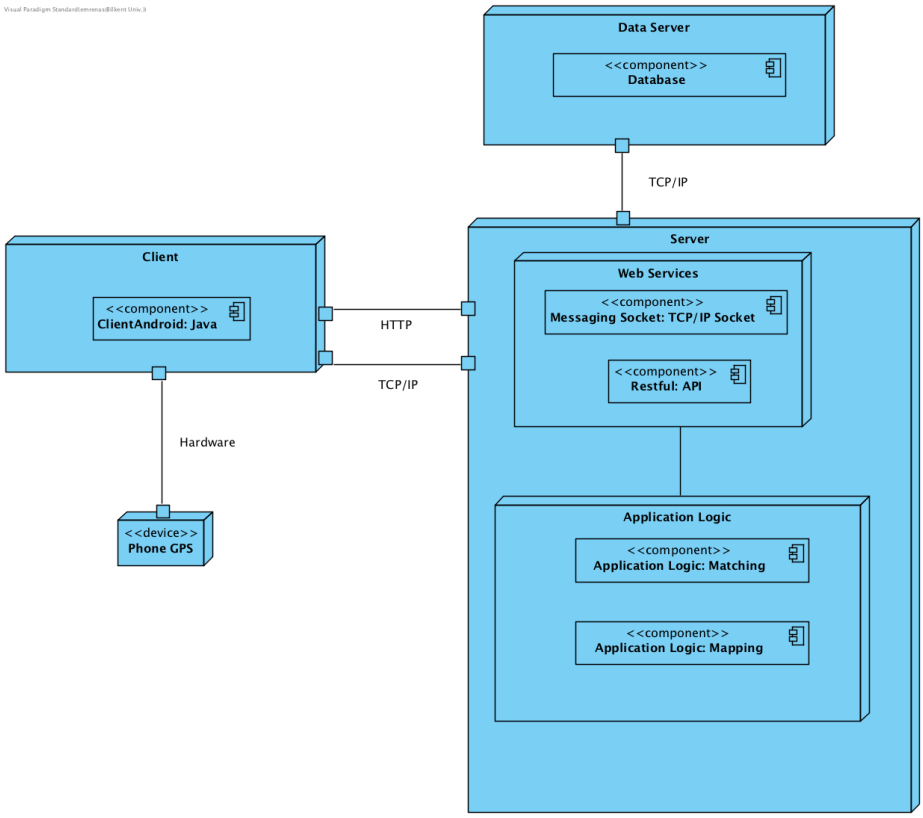
3.2.2. APPLICATION LOGIC

This layer contains matchmaking, route decision and user management subsystems. Matchmaking subsystem is the place where matching the users happens according to user's preferences. Route decision subsystem use matchmaking subsystem in order to get the detail of the user's location and getting directions for the driver to take the passenger. Since route decision subsystem communicate with matchmaking subsystem there are in the same layer. User management subsystem is for managing user information.

3.2.3. Network And Data Management Layer

Network subsystem is the part where communication between client-server communication happens. Data Management Subsystem stores the data of users' information such as their accounts, preferences, chat histories and also it is the subsystem where modifications on data can be happen. Messaging subsystem is the place where messaging occurs between users

3.3. HARDWARE/SOFTWARE MAPPING



For the above diagram, we used component and connecter view. There is only one client component which is for Smart phones running on Android operating system. This component will include graphical user interface and phone GPS will be connected to client. The Data Server will contain a database server which will be MySQL. Server will be Google Compute engine and will be connected Data Server via TCP/IP. Client and server will be connected through HTTP and TCP/IP. Application Logic will include Matching and Mapping. Web Services and Application Logic will also be connected via HTTP. Web Services include messaging socket for chatting and a restful API for sending and receiving information between the client, application logic and data server.

3.4. Persistent Data Management

Users' of the application can save their profile pictures in the program. For this purpose, we need a cloud storage that can be available all the time to access. On order to achieve this we plan to use Google Cloud Storage. Google Cloud Storage is an object storage service which designed for secure and durable storage.

We are going to store users' password encrypted with SHA-256 in order to protect users' account in case of a cyber-attack to our database server. Authentication of the system will use encrypted passwords to let users enter into the application.

For our database, we are going to use Google Cloud SQL. It is MySQL and PostgreSQL service. We are going to set up our database on MySQL. Google Cloud SQL offers high performance, scalability, and convenience.

3.5. Access Control and Security

CarBuds software require any user as a registered user in order to give access to its services. As a carpooling and hitchhiking application, system should provide maximum security about the information of the users from unauthorized third-parties.

System's control measures start at the very first stage of registering as a user. In order to register to CarBuds, user needs to be an active student of one of the verified universities. This measure is to provide real-world security due to car sharing. Another issue about security is keeping the user data safe due to privacy issues. So, application will be optimized in order to not harvest any user data for third-parties or any profit by using them.

Login process of the users also another concern of security since each account has the credentials about the user's personal life, which any leak can cause privacy interference. In order to access to the application, users' email or id and passwords should be validated. For keeping the password information, system uses SHA256 hashing algorithm. Therefore, even though any security breaches occur. The credential

of the users will be safe due to hashing. Since authentication for application is handled with API services of the application also we are reducing the risks of SQL injection

3.6. Global Software Control

- **Internal Control**

Internal control includes the authentication service that user enters his/her login information. By this way program will know if the user is registered or not using internal control. After program checked if the user is authenticated or not it can show user dependent graphical user interface. Other services like matchmaking or messaging are also under internal control of the software.

- **External Control**

The external control consists of the interaction between client and web services. This is the main communication between the user and the program itself. Web services and client connected to each other via HTTP for Restful API and via TCP/IP for real-time messaging. Data server and the restful API also connected via TCP/IP.

3.7. Boundary Conditions

The system for CarBuds shall have automated responses to the boundary conditions of initialization, termination and failure. The automated responses are explained in this section.

3.7.1. Initialization

The boundary condition of initialization should first start the database and the server infrastructure in order to set up the web services that required to

the Android application work. After the initial services and server started. User can either choose to login or register as a new user.

For both cases, since the web services up and running after initial start. All services and actions are available for the users with proper credentials for web services.

3.7.2. TERMINATION

The services which are allowed to run in the background of the client, does not have any time-out condition in order to logout the user from the client application. If user manually logouts or any unauthorized credentials about web services of the application server is detected, the session will be terminated. Although, client ends its connection with the server, unless any outside action has been done the server infrastructure will be always up and running.

3.7.3. ERROR

Since CarBuds depends on the internet connection to be able to communicate with the server. In case of a connectivity issue, client will disconnect its connection with the server and keeps a log in the client which contains information about time, location and reason of error in order to send later to the server when the internet connection established again.

4. Subsystem Services

The sub system services of the CarBuds software are explained in this section.

4.1. Application Client

Application client, handles all user interfaces and communications with the application server. No logic will be handled in this subsystem.

4.1.1. User Interface Subsystem

The User Interface Subsystem handles all the user interface operations. This includes all interfaces for login, register or any in-app screen in the CarBuds.

4.1.2. Server Communication Subsystem

The Server Communication Subsystem handles all the connections with the application server and application's web services in order to transmit the data that have been taken from User Interface Subsystem.

4.2. Application Server

Application Server handles all application logic and the networking and database management operations. All data flow will be handled in this subsystem.

4.2.1. Matchmaking Subsystem

The Matchmaking Subsystem is the main algorithmic logic subsystem in the application. Since CarBuds depends on matchmaking between a hitchhiker and a driver, in this subsystem all user preferences will be take in

consideration during the matchmaking process. This subsystem generates appropriate amount of possible matchmaking candidates for the users.

4.2.2. Route Decision Subsystem

After user decides his/her target destination both as hitchhiker or driver system firstly processes the routes of each user in order to make them quarriable for the matchmaking subsystem. Since Hitchhiking does not only involve finding an exact ride to the target destination, hitchhikers benefit some commonly used routes in order to make their journey to the target destination easier. Subsystem processes each route starting and finishing points and deciding possible midpoint stops in the route or alternative routes that still can be applicable for the users.

4.2.3. User Management Subsystem

Each user has its own profile page for both as hitchhiker and as driver that contains their information that they want to share with other users. Also, on the other hand, users have their preferences for their rides in order to be satisfied for most of the time for their trips. User Management Subsystem handles all the user data related operations and provide proper data for other application logic subsystems.

4.3. Application Network and Data Management

CarBuds' all web services, database related operations and communication protocols handled in this section.

4.3.1. Network Subsystem

Network Subsystem handles the communication operations between server, database and the connection between server and the client. Between client and server system uses HTTP Protocols in order to send and receive information which will be rendered to the user interface in the client side. Also handles all the query operations that comes from client side.

4.3.2. Messaging Subsystem

Messaging Subsystem handles the real time chatting between users of the application. Since each user is able to send messages to their matching users, Carbuds system handles the concurrency with a socket that runs with TCP/IP protocol which allows to process messages concurrently in real time. This provides a smoother and more responsive messaging.

4.3.3. Data Management Subsystem

Data Management Subsystem handles the database query operations for needed subsystems. Each subsystem depends on this subsystem because it contains and provides all the information about users' authentication credentials, matchmaking preferences, private messages and logs of the system.