Software Requirements Specification

CARPOOL MANAGEMENT SYSTEM

Submitted to:

Ms. YAKSHITA JAIN

Submitted by:

_____

AKHILESH KAPOOR (1711981021)

AKSHAY MAILIK (1711981025)

AKSHIT ARORA (1711981027)

_____

TABLE OF CONTENTS

INTRODUCTION

Carpooling (also car-sharing, ride-sharing and lift-sharing) is the sharing of car journeys so that more than one person travels in a car, and prevents the need for others to have to drive to a location themselves.

By having more people using one vehicle, carpooling reduces each person's travel costs such as: fuel costs, tolls, and the stress of driving. Carpooling is also a more environmentally friendly and sustainable way to travel as sharing journeys reduces air pollution, carbon emissions, traffic congestion on the roads, and the need for parking spaces. Authorities often encourage carpooling, especially during periods of high pollution or high fuel prices. Car sharing is a good way to use up the full seating capacity of a car, which would otherwise remain unused if it were just the driver using the car.

Carpool commuting is more popular for people who work in places with more jobs nearby, and who live in places with higher residential densities. Carpooling is significantly correlated with transport operating costs, including fuel prices and commute length, and with measures of social capital, such as time spent with others, time spent eating and drinking, and being unmarried.

1.1 Purpose

The purpose of this document is to build an online **Carpool Management System**. The document will be used to elaborate the functionality of the Carpool application. The document addresses a web-based application that is accessible from an Internet browser.

The document also describes the non-functional requirements such as the user interfaces. When the system is designed, it also considers the design constraints and other factors which will be used for the software. This application allows the users to perform tasks such as: search for the availability of a car, number of available seats, the time and date of availability of the car and provides the user with an option to book a seat in the car.

1.2 Intended Audience and Reading Suggestions

This project is a prototype for the carpooling system and it is restricted within the college premises. This has been implemented under the guidance of college professors. This project is useful for the taxi services that facilitate carpooling and as well as to the people who would like to save up on their commute costs by sharing a car.

1.3 Scope

Carpooling is environmental friendly, reduces cost of commute and it saves from traffic problems. Now a days, number of cars on road has increased tremendously this led to heavy traffic problems and some other environmental hazards like air pollution and noise pollution with the help of this we can share ride with people who are regularly coming from same place it could provide solution to these environmental problems.

With carpooling they can make their ride enjoyable while talking with fellow persons. It is good way to save on fuel consumption. One of the most important benefits of carpooling is saving of money not only in terms of fueling but also wear and tear ride will be reduced to great extent.

1.4 Definition

Car-Pooling:

Carpooling (also known as ride-sharing, lift-sharing), is shared use of a car, in particular for commuting to work, often by people who each have a car but travel together to save costs and in the interest of other socio-environmental benefits.

1.5 References

1. www.nrdc.org/reference/glossary/c.asp

2. en.wikipedia.org/wiki/Carpool

3. www.malmokongressbyra.se/ecomm2007/download/2_1.PDF

4. www.rideonline.com

5. www.mithfahren.com

6. www.carpoolworld.com

7. www.launchclinic.com/blog/2007/08/20/the-role-of-the-product-marketingmanager/

1.6 Overview

This document identifies the requirements, estimations and expectations within the scope of the "Car-Pooling: "Share My Ride" in terms of the development process, the product and administrative subjects, and to define the basic problem areas that have created the requirement.

GENERAL DESCRIPTION

2.1 Product Perspective

Our system is being implemented is a self-contained product and will be the first of its kind. It will be exclusively designed for the faculty, staff & students of our college i.e. NITC. Every day many persons including faculty, non-teaching staff, students etc., go to city by taking their own cars & faces traffic in road as well as in petrol pumps, also they have to bear whole maintenance of the ride therefore, we are creating a software that helps to reduce these problems through ride sharing & also gives a good company to the car owner.

2.2 Product Functions

The carpool functions will support the following functionality:

- Login – logging onto the web server
- Register - register in the carpooling
- Carpool search – search other members participating in the carpool
- Create Carpool – create carpooling for special events
- Join Carpool – join others carpool.
- View Carpool detail – view detail of the one-time event schedules.
- Delete Carpool detail - delete own created carpool.

2.3 User Classes and Characteristics

Participant (PRIORITY):

- **Admin (HIGH) -**
  He will check the information given by the guest user & provide him user-id and password. He has to maintain the priorities given by the car owners regarding the person for which they are comfortable in journey. He can provide new password if user wants.

- **User (MEDIUM) -**
  He can access his profile. He can change and update his profile, change password, create new car-pool, update pool already created, delete the pool created by himself, join new pool according to places and timings.

- **Guest (LOW) -**
  The guest can see the schedule to different places with search option by providing the corresponding date and time and he can register on the website by providing the required details. But he cannot join the carpool and create carpool without getting registered.

2.4 Operating Environment

- The computer must be linked up to a server loaded with the Database. The users will require for peripherals a mouse and a keyboard.
- It should be compatible with all the well-known and widely used web browsers like Internet Explorer, Mozilla Firefox, Google Chrome, and Opera.

2.5 Design and Implementation Constraints

- Its front end should be implemented using JAVA, NODE JS and Java-script.
- The back end should be managed by the MYSQL.
- It will be using the Andromeda server provided by NITC.
- The Backup and recovery functionalities will be constrained by the servers and the data.
- Storage facilities provided by Andromeda.

2.6 User Documentation

- This website will be a user – friendly and created in such way that the person having little or no knowledge of the website can use this; therefore no user manual will be required.
- Other queries will be solves in FAQ'S.

2.7 Assumptions

- It is assumed that the user should familiar with computer &having internet connection in the system.
- It is assumed that all information given by the user is correct regarding personal & scheduling information.
- Our system mainly depends on the users & highly affected when the information given for pooling is not followed.

FUNCTIONAL REQUIREMENTS

3.1 General application requirements

- **Login:**
  Since all the operations that can be done using the application requires both the driver and passenger to be logged in, they can use the login forms of either Google Plus or Facebook. For this matter, the user is prompted to connect the app to his account and then proceed for sign in/up. After the user authorizes the application to access his social media account, the server retrieves his info. If he has never logged to the application before, a new account is created for him.

- **Modify profile information:**
  All users can modify their profile information. The profile information contain: name, phone number, email, type/color of car if any. The user can edit this information easily in order to be contacted and recognized.

TERM DEFINITION

1. **DRIVER** - Any person that owns a car and wants to go from one place to another can publishes his trip on the application.

2. **PASSENGER** - Any person that doesn't own a car and wants to join a driver in a trip he posted and agrees to all the condition specified (price and general behavior).

3. **REGULAR TRIP** - A onetime long distance planned travel between two points (usually cities) with a defined departure time and price.

4. **FREQUENT TRIP** - A short to medium distance frequent (daily/weekly) between a neighborhood and a workplace, school or other point of interest.

5. **SOCIAL MEDIA SHARING -** In order to attract more users to the application and help users find passengers, users should be able to share their activity on the application on social media. A suggestion for sharing trips' creation, trips'

registration or check-in should pop-up whenever those previous actions are performed. The sharing should be authorized by the users and not done automatically by the application.

6. **RATE DRIVER/PASSENGER -** Both the driver and passenger can rate each other in other to gain reputation. The importance of the rating is to encourage users to be helpful and nice during the trip so that they gain popularity in the application. It is also a way to ensure users of who can be trusted or not. The ratings represent a relative guarantee for the users to trust each other.

3.2 Regular trips

- **Create new regular trip:**
  The driver can create a new trip to be displayed when passengers search for trips. The application will prompt the driver for information of the regular trip which consists of destination, origin, meeting point (which can be pointed on a map), departure time/date , estimated arrival time and traveling preferences (number of free spots, price, size of bags, smoking/non-smoking, pets, stops). After providing this information, the user publishes it in order to find passengers. Upon the creation of the trip, a user can share the trip he just created in social media to find passengers to drive with.

- **Search for regular trips and reservation:**
  When a passenger needs to find a driver for a destination, he can use a search form which asks for destination, origin, departure date/time. He can also specify the travelling preferences. When he finds a suitable trip, he can reserve a spot easily in by taping a button which will send a notification to the driver telling him that a passenger has reserved.

- **Check-in trip:**
  Whenever the driver or passenger arrives at the meeting point at the time agreed upon, he can check-in the meeting point in order to notify the other user and to show his punctuality. The application will use the devices GPS in order to make sure that the users are in the meeting point. When someone checks in, a notification is sent to all the carpoolers saying that somebody is in the meeting point.

NON-FUNCTIONAL REQUIREMENTS

- **Performance:**
  The application has to offer a very quick response time as the meeting between the driver and passengers is done through notifications. In other words, the server should be able to treat notifications and propagate them instantly. The application should handle 1000 users sending queries at the same time.

- **Scalability:**
  The application should respond properly to a high increase of users. It should be able to handle from 10,000 users to 100,000 users.

- **Extensibility:**
  The application should by extensible in order to support multiple platforms including iOS, Windows Phone and Web.

- **Availability:**
  Since a lot of information about the trips and check in are available in the application, it has to be highly available and guarantees a good server up-time. The server should allow only 1 hour down time per year which is 99.99% up-time.

- **Privacy and Security:**
  The application should ensure the privacy of the users including the trips they take part in, their social media accounts and their accounts. The login system should also be robust where only authorized users can post and edit their own information.
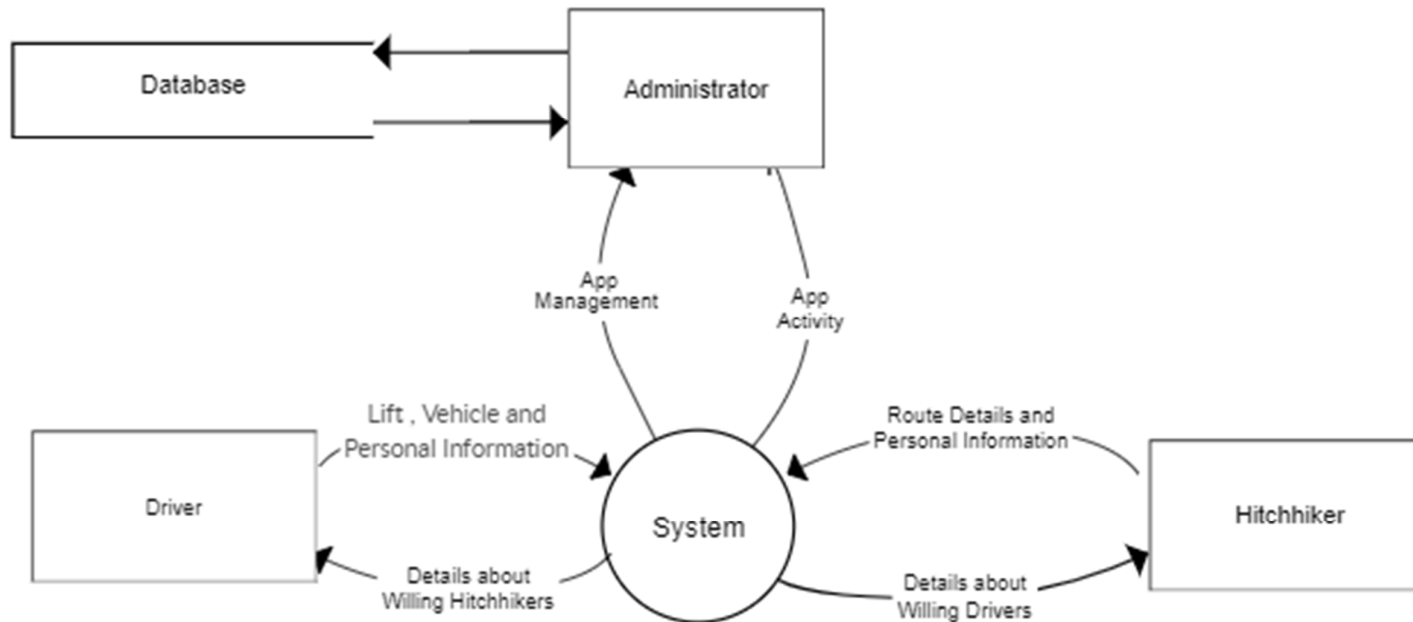
- **Maintainability:**
  Since the application may be developed in the future by adding other features, it should be easily maintainable. 14 II) Technology enablers the choice of technology enablers that will be used for the development of the application is essential for its success. The technology enablers should provide a suitable way to fulfill the requirements stated before. Principles of enterprise class applications should be kept in mind during the choice of these technologies.

SYSTEM ARCHITECTURE
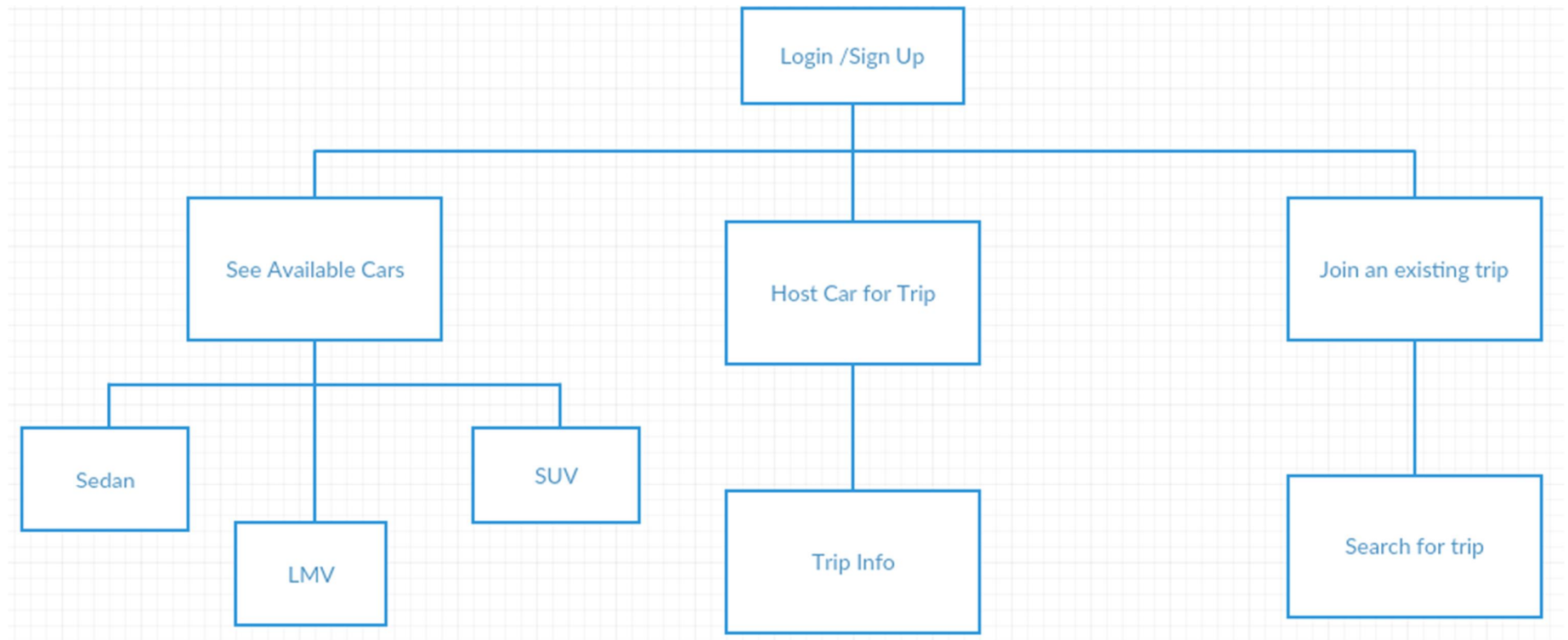
DATA MODEL

Data Flow Diagram Level – 0

CHITKARA UNIVERSITY

Data Flow Diagram Level – 1