

# Software Requirements Specification

CS 258 Software Engineering

February 2020

The following annotated template shall be used to complete the Software Requirements Specification (SRS) assignment of CS 258.

## **Template Usage:**

The text contained within angle brackets ('<', '>') shall be replaced by your project-specific information and/or details. For example, Car Pooling for IITI will be replaced with either 'Smart Home' or 'Sensor Network'.

Italicized text is included to briefly annotate the purpose of each section within this template. This text should not appear in the final version of your submitted SRS.

This cover page is not a part of the final template and should be removed before your SRS is submitted.

## **Acknowledgements:**

Sections of this document are based on the IEEE Guide to Software Requirements Specification (ANSI/IEEE Std. 830-1984). The SRS templates of Dr. Orest Pilskalns (WSU, Vancouver) and Jack Hagemester (WSU, Pullman) have also been used as guides in developing this.

# Car Pooling for IITI

## Software Requirements Specification

0.1

14-02-2020

Adarsh Baghel  
Ruchir Mehta  
Namani Sreeharsh  
Shaikh Ubaid

Prepared for  
CS 258 Software Engineering  
Spring 2019



## Revision History

Date	Description	Author	Comments
<date>	<Version 1>	<Your Name>	<First Revision>

## Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	<Your Name>		

# Table of Contents

<b>REVISION HISTORY</b>	<b>II</b>
<b>DOCUMENT APPROVAL</b>	<b>II</b>
<b>1. INTRODUCTION</b>	<b>I</b>
1.1 PURPOSE	I
1.2 SCOPE	I
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	I
1.4 REFERENCES	I
1.5 OVERVIEW	I
<b>2. GENERAL DESCRIPTION</b>	<b>II</b>
2.1 PRODUCT PERSPECTIVE	II
2.2 PRODUCT FUNCTIONS	II
2.3 USER CHARACTERISTICS	II
2.4 GENERAL CONSTRAINTS	II
2.5 ASSUMPTIONS AND DEPENDENCIES	II
<b>3. SPECIFIC REQUIREMENTS</b>	<b>II</b>
3.1 EXTERNAL INTERFACE REQUIREMENTS	III
3.1.1 <i>User Interfaces</i>	<i>iii</i>
3.1.2 <i>Hardware Interfaces</i>	<i>iii</i>
3.1.3 <i>Software Interfaces</i>	<i>iii</i>
3.1.4 <i>Communications Interfaces</i>	<i>iii</i>
3.2 FUNCTIONAL REQUIREMENTS	III
3.2.1 <i>&lt;Functional Requirement or Feature #1&gt;</i>	<i>iii</i>
3.2.2 <i>&lt;Functional Requirement or Feature #2&gt;</i>	<i>iii</i>
3.3 CLASSES / OBJECTS	III
3.3.1 <i>&lt;Class / Object #1&gt;</i>	<i>iii</i>
3.3.2 <i>&lt;Class / Object #2&gt;</i>	<i>iii</i>
3.4 NON-FUNCTIONAL REQUIREMENTS	III
3.4.1 <i>Performance</i>	<i>iv</i>
3.4.2 <i>Reliability</i>	<i>iv</i>
3.4.3 <i>Availability</i>	<i>iv</i>
3.4.4 <i>Security</i>	<i>iv</i>
3.4.5 <i>Maintainability</i>	<i>iv</i>
3.4.6 <i>Portability</i>	<i>iv</i>
3.5 INVERSE REQUIREMENTS	IV
3.6 DESIGN CONSTRAINTS	IV
3.7 LOGICAL DATABASE REQUIREMENTS	IV
3.8 OTHER REQUIREMENTS	IV
<b>A. APPENDICES</b>	<b>IV</b>
A.1 APPENDIX 1	IV
A.2 APPENDIX 2	IV

# 1. Introduction

*This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.*

*Since inappropriate planning of the cities, there has been a big problem of traffic in most cities of India. People spend much of their time in traffic every day. In Addition to this many vehicles in traffic makes rapid oil consumption, there has been an uprising problem of air pollution. Oil supplies are very limited all over the world and oil prices are extremely expensive in our country. Therefore, most of the people have to take buses and since the number of public transportation vehicles are not sufficient, they travel under uncomfortable conditions. There are some attempts to solve these problems, however, they focus only on intercity transportation. We came up with an effective solution as [www.uniteonwheels.com](http://www.uniteonwheels.com). Our project will be used for both intercity and urban transportations all over India. As a result, our system will be designed to solve these problems and deficiencies of other systems .*

## 1.1 Purpose

*The purpose of this document is to give a detailed description of the requirements for the “Car Pooling for IITI” (CPIITI) software. It will illustrate the purpose and complete declaration for the development of the system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.*

## 1.2 Scope

The “Car Pooling for IITI” is a GPS-based mobile application which helps people to find the closest vehicle based on the user’s current position and other specifications like time of departure, price, contact information of passengers travelling and number of seats available. This information will act as the basis for the search results displayed to the user. The application should be free to download from either a mobile phone application store or similar services. Furthermore, the software needs both Internet and GPS connection to fetch and display results. All system information is maintained in a database, which is located on firebase( an online db service). The application uses the mobile phone’s GPS navigator. By using the GPS-Navigator, users can view available vehicles on a map and can navigate towards them. The application also has the capability of representing both summary and detailed information about the available vehicles which will take the passenger to the required destination .

This application will bring about a big revolution in sharing vehicles thus reducing pollution and traffic in cities. This will be operated from both the passenger intended to travel and the passenger already travelling in a vehicle and willing to share their ride.

### 1.3 Definitions, Acronyms, and Abbreviations

*This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS.*

<b>TERMS</b>	<b>DEFINITION</b>
<i>CLUSTER</i>	<i>Group of people travelling in the same automobile.</i>
<i>CLUSTER ADMIN</i>	<i>The person who acts as a gateway for communicating about the whereabouts of the cluster with the prospective passengers.</i>
<i>CLUSTER MEMBERS</i>	<i>All members of the cluster including cluster admin.</i>
<i>FAVOURITE MODE</i>	<i>Mode which shows the positions(locations) of the connected friends only.</i>
<i>PUBLIC MODE</i>	<i>Mode which enables the users to track the routes of all the available clusters (inclusive of our favourite people) .</i>
<i>TOGGLE BUTTON</i>	<i>Mode which toggles visibility of cluster's route to preferred people and anonymously to the public.</i>
<i>PASSENGERS</i>	

### 1.4 References

- (1) <https://senior.ceng.metu.edu.tr/2014/such/documents/SRS.pdf>
- (2) [IEEE STD 1233-1998, IEEE Guide for Developing System Requirements Specifications](#)
- (3) [IEEE STD 830-1998, IEEE Recommended Practice for Software Requirements Specifications](#)

### 1.5 Overview

- (1) *In the rest of the SRS we have discussed the user and system, constraints and requirements. Further we will discuss the limitations of the application. Getting started documentation has also been mentioned in this SRS.*
- (2) *SRS is organised in a way firstly describing the features offered by our application moving to the user, system constraints and some general constraints.*

## 2. General Description

*This section of the SRS should describe the general factors that affect 'the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.*

### 2.1 Product Perspective

*/\*This subsection of the SRS puts the product into perspective with other related products or projects. (See the IEEE Guide to SRS for more details).\*/*

*We have disseminated this section into Product Functions, User Characteristics, Product Perspective and some General constraints.*

### 2.2 Product Functions

*/\*This subsection of the SRS should provide a summary of the functions that the software will perform. \*/*

**#Sign Up:** *Users need to sign up to use the app. The users should have a username and password. After filling their name, surname, email, age, designation, phone and gender information, they register into the system.*

**#Sign In:** *If a user is signed up, s/he can sign in the system by filling username and password boxes.*

**#Sign Out:** *A user may need to sign out of the system. He/She can do it by clicking the sign out button which is placed on every page.*

**#Accepting passengers preferences:** *This functionality allows passengers to enter their departure point, arrival point, mode of transport(car,autos,cabs,busses), date and time of departure.There will be certain tolerance limit set by the user itself(for this we will have to set lower tolerance limit(before preferred time) and upper tolerance limit(after preferred time) for extracting recommendations). User also can select the clusters which he/she wants to join i.e., if he/she wants to join a favourite*

**#Personalised suggestions:** *Our CarPooling Application analyses the database frequently and gives personalised suggestions regarding best departing time(based on time tolerances)(this is different from preferred time as it may happen that a big cluster may begin their journey at some time just before or after the preferred time, so our app may request us to depart at the nearest time possible to the preferred time) , travelling time, mode of transport(car,autos,cabs,busses) and nearest pickup locations and all other locations of pickup(there will be a certain criterion (distance)for suggesting a place as the best point for boarding the vehicle which is certainly walkable from the current position of the user)(this is different from our point of departure as sometimes, it may happen that a big cluster may depart from a point near the preferred pickup point , so our app may suggest them to depart from the changed point) (this functionality also*



encompasses route to that point)so that number of clusters is minimised at any point of time, which is beneficial to all the users, as per head cost is significantly reduced.

**#Adding clusters to the database:** Our application also needs user's amicable co-ordination. If users are ready to accompany some people then, this may also become a potential cluster and adding this to the database will improve the reachability of the app. So, we enable users of the app with the best possible UI to add their current location, their departure time, departure point, arrival point. This is also integrated with personal chat with the passengers of the cluster for knowing the conditions of the traffic and also the feasibility of their inclusion in their cluster i.e, if there exists heavy traffic in the route planned before then, by personal chat existing passengers may request him/her to choose another cluster over them or begin a new cluster.

**#Send Message:** The users can communicate with each other by sending messages.

**#Reply To A Message:** After receiving a message, the user can read the message and reply to it.

**#Block User:** When a user receives a disturbing message, s/he can block the user who send that message.

**#See cluster's route:** This functionality enables the user to see a particular cluster's current route and this is preceded by a set of all available clusters ready to accommodate additional passengers.

**#Disable additional inclusions:** This functionality disables the cluster to allow additional passengers. Passengers can do this if they feel that they can't accommodate additional passengers and it stops covering their details.

**#Feedback from the customers:** This functionality enables the customer to give their feedback/suggestions regarding the service provided by the app and it lets the developers know what functionalities are to be added from time to time on a regular basis . Users rate the application by stars ranging from 1 to 5 inclusive of floating numbers ending with .5 .

## 2.3 User Characteristics

*This subsection of the SRS should describe those general characteristics of the eventual users of the product that will affect the specific requirements. (See the IEEE Guide to SRS for more details).*

Users of this application belong to the community of IIT INDORE in the preliminary stage of the app. Users need to enter the details and know the details at any point of time , mentioned in the section of 2.2.They need to go to the city or some other tourist spots present in the vicinity of IIT INDORE . For that, they need to know where and when maximum number of passengers are using a particular automobile and list of groups of people who are ready to accommodate more people. People needing to use the pooling service will only use the application. They may also be ready to leave from the place

## 2.4 General Constraints

*This subsection of the SRS should provide a general description of any other items that will limit the developer's options for designing the system. (See the IEEE Guide to SRS for a partial list of possible general constraints).*

- **Flutter: version 1.0**
- **Firebase: version 1.0**
- **Delivery Date: 14th Feb 2020**
- **Budget Constraints: 10,000\$**

## 2.5 Assumptions and Dependencies

*This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.*

- *The device should be a “SmartPhone”, not a Feature Phone.*
- *Minimum RAM(Random Access Memory): 1 GB*
- *Minimum Memory: 100 mb*
- *Android Version: 6.0 MarshMallow*
- *Location Service: The device must have a GPS(Global positioning system) receiver hardware installed in it.*
- *Internet Connection: An active internet connection is a must to use this application.*

## 3. Specific Requirements

*This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project's software design, implementation, and testing.*

*Each requirement in this section should be:*

- *Correct*
- *Traceable (both forward and backward to prior/future artifacts)*
- *Unambiguous*
- *Verifiable (i.e., testable)*
- *Prioritized (with respect to importance and/or stability)*
- *Complete*
- *Consistent*
- *Uniquely identifiable (usually via numbering like 3.4.5.6)*

*Attention should be paid to carefully organize the requirements presented in this section so that they may be easily accessed and understood. Furthermore, this SRS is not a software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.*

### 3.1 External Interface Requirements

#### 3.1.1 User Interfaces

This software product is developed for drivers and hitchhikers. Products will be deployed to an application and all users of the system will access the system through the application based interface which includes multiple pages according to the system functionality, for instance, for login functionality there will be a login page. To access the system, every user has a unique username and password. In addition, there will be a database which stores and manipulates all the data about the users. App will only be the interface for all the user data which is stored by database and the execution of provided functionalities. After the sign up, user information will be transferred to the database. In the sign up process, Google sign in will be used to authenticate users. After that point, users can register through the web interface. After logging in, users will be able to log out whenever they want.

#### 3.1.2 Hardware Interfaces

The system runs on a mobile device, using android OS. So there is no such hardware interface, it will be managed by the in-built OS.

#### 3.1.3 Software Interfaces

The system has Google Map API as a subsystem. Google Map subsystem has their own app based interface which is a map consisting of roads and locations in a desired area and users can easily interact with this system.

#### 3.1.4 Communications Interfaces

In communication between driver and hitchhiker, For communication between users and drivers, a chatting portal will be used, the system shall support messaging functionality and users will be able to send and receive messages through the remote mobile devices.

### 3.2 Functional Requirements

*This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.*

#### 3.2.1 <Functional Requirement or Feature #1>

##### 3.2.1.1 Introduction

Such an application is required to make travelling between long distances faster and cheaper and thus reducing traffic and pollution in big cities also leading in reduction of rapid fuel consumption.

**3.2.1.2 Inputs****3.2.1.3 Processing**

#Ratings for passengers  
 #Shortest Distance  
 #Queries from the database

**3.2.1.4 Outputs**

#Locations of available

**3.2.1.5 Error Handling**

<https://docs.google.com/document/d/1UcafS0RwejjJJzYF3KaWfQOIYI5dT8yGBuGddHsiXBQ/edit?usp=sharing>

**3.2.1. Sign In**

Use Case ID	UC2
Actor(s)	User
Description	User Log In
Preconditions	The user shall be able to sign in to the system.
Post conditions	Users will be able to use the system.
Precedence	Mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. The user opens the app and enters his phone no to the system.</li> <li>2. User presses the login button.</li> <li>3. User enters his or her OTP received by phone .</li> </ol>

Alternative Flow(s)	<p>Flow 1:</p> <ol style="list-style-type: none"> <li>1. If the user enters the wrong OTP information, the warning message for example "Wrong OTP information" will be shown to the user.</li> </ol> <p>Flow 2:</p> <ol style="list-style-type: none"> <li>2. If the user enters his or her OTP correctly, the user will be redirected to the application relevant page of the system.</li> </ol>
---------------------	---

### 3.2.1. Sign Out

Use Case ID	UC3
Actor(s)	User
Description	User Log Out
Preconditions	The user shall be able to log out into the system.
Post conditions	Users will be able to leave the system.
Precedence	Not mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. User presses the log out button.</li> <li>2. User leaves the system.</li> <li>3. The app's login page will be loaded.</li> </ol>

### 3.2.2. Add Vehicle Available Notification.

Use Case ID	UC4
Actor(s)	User

Description	Users shall be able to add route from the map.
Preconditions	The user shall be able to sign in to the system.
Post conditions	Users shall be retrieve transportation requests from the other users.
Precedence	Mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. User shall enter her or his profile page.</li> <li>2. User shall press the new transportation button.</li> <li>3. The transportation page will be loaded.</li> <li>4. User enters departure time, available seats and iteration of transportation like “one time” or “periodic”.</li> <li>5. 5. User draws a route on the map panel.</li> </ol>

### 3.2.3. Delete Add Vehicle Available Notification.

Use Case ID	UC5
Actor(s)	User
Description	Users shall be able to delete route.
Preconditions	The user shall add a transportation route before.
Post conditions	Users cannot see the route which is deleted by the user.
Precedence	No mandatory

Normal flow of event	<ol style="list-style-type: none"> <li>1. User shall presses my transportations button.</li> <li>2. My transportations page will be loaded.</li> <li>3. User selects the route he or she wants to delete.</li> <li>4. Delete button is clicked.</li> <li>5. The user deletes the route.</li> </ol>
----------------------	--

### 3.2.4. Request Transportation Route

Use Case ID	UC6
Actor(s)	User
Description	Users shall be able to request route.
Preconditions	The user shall search the route.
Post conditions	Users will be able to contact the driver who owns the route.
Precedence	No mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. User presses the request route button.</li> </ol>

### 3.2.5. Search Transportation Route

Use Case ID	UC7
Actor(s)	User
Description	Users shall be able to search the route.
Preconditions	The user shall sign in to the system.
Post conditions	Users will be able to select route from the available route list.
Precedence	No mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. User fills "from" input field.</li> <li>2. User fills the "to" input field.</li> <li>3. User presses the search button.</li> </ol>

Alternative Flow(s)	<p>Flow 1:</p> <ol style="list-style-type: none"> <li>1. User forgets to fill “from” or “to” input field.</li> <li>2. The related warning message is shown to the user to fill the input fields properly.</li> </ol>
	<ol style="list-style-type: none"> <li>1. User fills the input fields properly.</li> <li>2. The available routes will be listed.</li> </ol>

### 3.2.6. Send Message

Use Case ID	UC8
Actor(s)	User
Description	Users shall be able to send messages through the system.
Preconditions	The user shall sign in to the system.
Post conditions	Users will be able to communicate with each other.
Precedence	No mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. User enters the profile page of the user who is intended to be communicated.</li> <li>2. User presses the send message button.</li> <li>3. The message page will be loaded.</li> <li>4. User types the content of the message.</li> <li>5. User presses the send button to send the message content.</li> <li>6. The message content will be</li> </ol>



	stored and viewed in the message panel.
--	---

### 3.2.7. Reply to Message

Use Case ID	UC9
Actor(s)	User
Description	User shall be able to reply to incoming message through the system.
Preconditions	The user shall receive the message from other user.
Post conditions	Users will be able to communicate with each other.
Precedence	No mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. User presses the message box button.</li> <li>2. The message box page will be loaded.</li> <li>3. User clicks the intended row from the message list.</li> <li>4. The message page will be loaded.</li> <li>5. User types the content of the message.</li> <li>6. User presses the send button to send the message content.</li> <li>7. The message content will be stored and viewed in the message panel.</li> </ol>

### 3.2.8. Rate User

Use Case ID	UC11
Actor(s)	User
Description	Users shall be able to rate the driver through the system.
Preconditions	The transportation route shall be completed with driver and hitchhiker.
Post conditions	The driver's rating will be updated.
Precedence	No mandatory
Normal flow of event	<ol style="list-style-type: none"> <li>1. After the transportation, the hitchhiker login to the system.</li> <li>2. Popup window is open.</li> </ol>
Alternative Flow(s)	<p>Flow 1:</p> <ol style="list-style-type: none"> <li>1. Hitchhiker clicks the star icon to rate the driver's related attitude.</li> </ol> <p>Flow 2:</p> <ol style="list-style-type: none"> <li>1. Hitchhiker clicks close icon.</li> <li>2. The popup window will be closed.</li> </ol>

### 3.2.2 <Functional Requirement or Feature #2>

## 3.3 Classes / Objects

### 3.3.1 <Class / Object #1>

#### 3.3.1.1 Attributes

#### 3.3.1.2 Functions

<Reference to functional requirements and/or use cases>

### 3.3.2 <Class / Object #2>

...

## 3.4 Non-Functional Requirements

*Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).*

**3.4.1 Performance:** Location of members travelling in a vehicle should be updated in database without any delay and this live location should also be reflected in other user's application.

**3.4.2 Reliability:** Our software ensures location accurately upto 5-mts.

**3.4.3 Availability :** Our software is made to ensure 24/7 service.

**3.4.4 Security:** One can join a ride in both in the Favorite mode and Public mode where when the person is in Favorite mode, he can join rides of his friends only while when he is in Public mode he can join all those rides where the passengers are willing to accept anonymous ride.

**3.4.5 Maintainability:** The application code is refactored, separated into different modules (modularity) so that it is easier to understand and debug. This ensures that the code is less complex and developer friendly.

**3.4.6 Portability:** Our application is easy to download from the Google PlayStore and easy to transfer between users in form of an application bundle known as apk file which is easily available on our website [www.uniteonwheels.in](http://www.uniteonwheels.in).

## 3.6 Design Constraints

*Specify design constraints imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.*

### 3.7 Logical Database Requirements

*Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.*

*A database will be used to store current location of the passengers travelling and that location will be accessed by the passengers who are intended to travel in the vehicles*

### 3.8 Other Requirements

*Catchall section for any additional requirements.*

## A. Appendices

*Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS's overall set of requirements.*

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

### A.1 Appendix 1

### A.2 Appendix 2