



Software Requirements Specification

for

Ride Share Mobile App

Version 1.0

Prepared by

Group Name: Group 1

**James Hayes
David Parrott
Adam Bennett
Ryan Slyter
Brett Carter**

**11152406
11239947
11269004
10799467
11268678**

**james.hayes@wsu.edu
david.m.parrott@wsu.edu
adam.m.bennett@wsu.edu
ryan.slyter100@email.wsu.edu
brett.carter@wsu.edu**

Date: 5th of November

Contents

REVISIONS

1 INTRODUCTION

- 1.1 DOCUMENT PURPOSE
- 1.2 PRODUCT SCOPE
- 1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW
- 1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS
- 1.5 DOCUMENT CONVENTIONS
- 1.6 REFERENCES AND ACKNOWLEDGMENTS

2 OVERALL DESCRIPTION

- 2.1 PRODUCT PERSPECTIVE
- 2.2 PRODUCT FUNCTIONALITY
- 2.3 USERS AND CHARACTERISTICS
- 2.4 OPERATING ENVIRONMENT
- 2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS
- 2.6 ASSUMPTIONS AND DEPENDENCIES

3 SPECIFIC REQUIREMENTS

- 3.1 EXTERNAL INTERFACE REQUIREMENTS
- 3.2 FUNCTIONAL REQUIREMENTS
- 3.3 BEHAVIOUR REQUIREMENTS

4 OTHER NON-FUNCTIONAL REQUIREMENTS

- 4.1 PERFORMANCE REQUIREMENTS

APPENDIX A – DATA DICTIONARY

APPENDIX B - GROUP LOG

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Ryan Slyter Adam Bennet James Hayes David Parrott Brett Carter	Initial Version.	10/21/14

1 Introduction

TeamSnap software serves sports teams of all variations by enabling them to better manage their team members and schedules. As a rapidly growing company, TeamSnap has developed a need for 3rd party applications, especially in the mobile area of applications, to further support customer of the company's primary offerings. It is the purpose of the Ride Share Mobile App for TeamSnap to provide customers with a means to coordinate giving and receiving rides to a sports event that is on the team's schedule. In this section one find the overall purpose, scope, and ancillary descriptions of this document that will help further describe the product.

1.1 Document Purpose

Ride Share Mobile App for TeamSnap is a third party iOS application. Currently the version 1.0 prototype is underway, with initial designs being laid out. This SRS covers the entire application, from front end graphical user interface to how the back end of the product transfers customer information.

1.2 Product Scope

Ride Share Mobile App for TeamSnap is designed to provide support for TeamSnap in a specific way, as it is a third party application. The primary benefit of the application is that iOS users who are customers of TeamSnap can coordinate with other members of their team for carpooling to scheduled sporting events. Not only does this allow users to easily find rides to scheduled events, but the synchronization of the application provides a unified means of communication with other team members, as the application queries TeamSnap's own stored player information. Because the application is developed for iOS 7.0, it will be available to almost all customer using a iPhone.

The goal of the application is that users find a direct value in using the product, therefore encouraging other members of a given team to purchase the application. This will gather further support and customer loyalty to both the app and the TeamSnap service.

1.3 Intended Audience and Document Overview

The intended audience of this SRS document is our overseeing project professor, Dr. Xinghui Zhao, and TeamSnap's Chief Product Officer, Andrew Berkowitz. The document is meant as a means for them to easily see and refer to our overall product goals and commitments. It also serves an ancillary purpose of allows the project team to make sure that intended goals and functionalities of the product are being met.

1.4 Definitions, Acronyms and Abbreviations

API: Application Programming Interface

DB: Database
GM: Google Maps
IaaS: Infrastructure-as-a-Service
iOS: Apple's operating system for its iPhone product.
MVC: Model View Controller
RSMA: Ride Share Mobile App (our application)
Restful: TeamSnap's database API
TS: TeamSnap

1.5 Document Conventions

There are no specific conventions in this document other than the acronyms that are used, which were outlined in section 1.4.

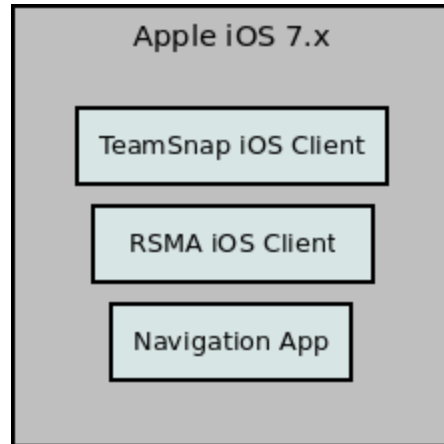
1.6 References and Acknowledgments

There are no references for this document. A lone acknowledgement goes out to TeamSnap for their support during the development of this project.

2 Overall Description

2.1 Product Perspective

Ride Share Mobile App for TeamSnap is a self-contained application, but it is a third party app to TeamSnap and its family of services. It operates on Apple's iPhone operating system, iOS, and requires a newer generation of iPhone which can run iOS 7.0 or newer, the version of iOS which TeamSnap's own mobile device supports. Customers can easily use both applications on their phone. RSMA is directly related to TeamSnap and its services by the fact that customers, and therefore the application, need to obtain information about a customer and other member of the team. The application queries TeamSnap for information about the customer and team in order to complete its primary purpose of finding team members who are willing to carpool to a specific team event. (picture on next page).



Product Functionality

Major Functions:

Allow the user/team member to specify that they need a ride to an upcoming event.

Allow the user/team member to specify that they are available to give carpool ride and then notify users on the same team.

Inform users of where team members are via GM so that they have concise directions of how to pick up said team members.

Inform users who are receiving a ride of where the team member providing the ride is driving from.

2.2 Users and Characteristics

There are three types of users for RSMA: Players, Managers and Non-players.

Players: These users are on a team and are likely using the free version of TS or contributing as part of a team. They will be using the app to find rides amongst other team members.

Managers: Are in charge of the team and likely control the TS account for it. They will be less likely to use the RSMA to find a ride for themselves but may monitor it in the event a team member needs a ride and no team member can pick them up.

Non-players: Usually a parent of a child or children on a team that uses TS. They are likely to use the app for car pooling and last minute emergency rides. Also likely to use the app as a means of quickly and easily accessing the navigation features. We anticipate these users being of primary importance.

The most important users of this product will be Team Managers. They have the ability to add and remove team events and also act as team members that can give and receive rides (with editing events and giving rides being the most frequent).

2.3 Operating Environment

The RSMA client requires a device running iOS 7 or later and an active internet connection. The RSMA server can run on any Unix based web server with an active internet connection. Optimal environment for the RSMA server will have the database co-located (same instance/server or same network) to eliminate accesses to the DB via the internet.

2.4 Design and Implementation Constraints

Since RSMA is a web application designed to work with exposed APIs there is flexibility in design constraints. The RSMA server must be able to communicate with the TS RESTful API, the Google Maps API, the client app running on customer's iOS devices and a database used to store user and routing information. RSMA will utilize OATH 2.0 to allow users to authenticate with their TS credentials. This will allow RSMA to not store any authentication information once a user's token has been verified. The database containing user information will have user location information making it necessary to have a strong password protecting access to the database.

2.5 User Documentation

User documentation for RSMA will be minimal. A FAQ document available publicly on the web should be sufficient to answer any questions users have. Additionally support via email will be available.

Issues with unresponsiveness of the application are incumbent on the owner/user of their iPhone to read their phone's manual or to contact Apple.

2.6 Assumptions and Dependencies

The external factors of greatest concern are the external APIs we will be accessing. Changes to the TS, Google Maps, or iOS API have the potential to severely impact RSMA necessitating workarounds and updates to the software. As the number of user accounts grows we may find it necessary to change our server architecture to accommodate the additional load. We will be using Amazon Web Services (or a similar server rental service) to provide IaaS which will enable us to provision additional application-hosting resources as needed.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The screens will have the aesthetic as supplied by Xcode (so standard iOS greyscale with push buttons and text displays:

The app will open a main menu screen with several buttons:

NAR: button to push to notify other members that user needs a ride.

Rides I Give: takes the user to a secondary screen displaying current ride-shares confirmed

Events: takes the user to a secondary screen displaying the teams scheduled events

Next Ride: takes the user to a secondary screen displaying the next ride event they are involved in. They can click 'next' on that screen to display the event after, and so on.

Events: By clicking on an event from the list, the user is taken to a 3rd screen showing which team members need a ride for that events, along with a text box displaying any other pertinent details of that event.

It is important to mention that when the 'NAR' button is pushed from the main menu, users are notified via the built it notifications menu provided by iOS, which looks like this:



Selecting 'Yes' from an event from this menu takes the user back to the application.

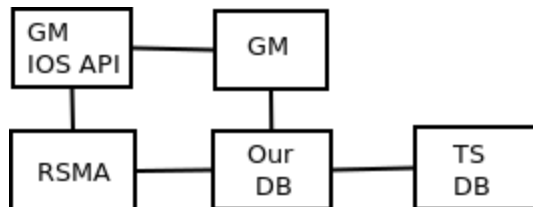
3.1.2 Hardware Interfaces

All hardware interfacing will be handled by iOS. By using the standard programming tools for iOS (Xcode) we are able to let the built in functionality to mask the hardware interface.

3.1.3 Software Interfaces

RSMA will act as a bridge between several software interfaces: iOS, TS, and Google Maps. As the primary function of RSMA is aggregating and using specific information from TS to present to Google Maps and then return relevant results to the user via iOS there will necessarily be an SQL database utilized by RSMA. RSMA will take location information from TS, combine that with

decisions based on user-set preferences to query Google Maps, select which the results returned from Google Maps are suitable and then present them to the user.



3.1.4 Communications Interfaces

All communications with TS and Google Maps will be via HTTPS. Both of those services are accessed via a RESTful API. Communication with the RSMA server will take place over an encrypted protocol TBD.

3.2 Functional Requirements

3.2.1 Allow the user/team member to specify that they need a ride to an upcoming event.

- A button dedicated to requesting a ride will be placed on the home screen of the app allowing a user to quickly and simply request a ride.

3.2.2 Allow the user/team member to specify that they are available to give carpool ride and then notify users on the same team.

- Users will specify how much travel time they are willing to add to their trip to events during initial setup, and can alter these preferences at any time.
- When a ride request comes in all notified users will be able to accept or decline with one button press once the notification has been selected and RSMA has been launched.
- The first user to accept will then be designated as the driver for that rider to all other team members.

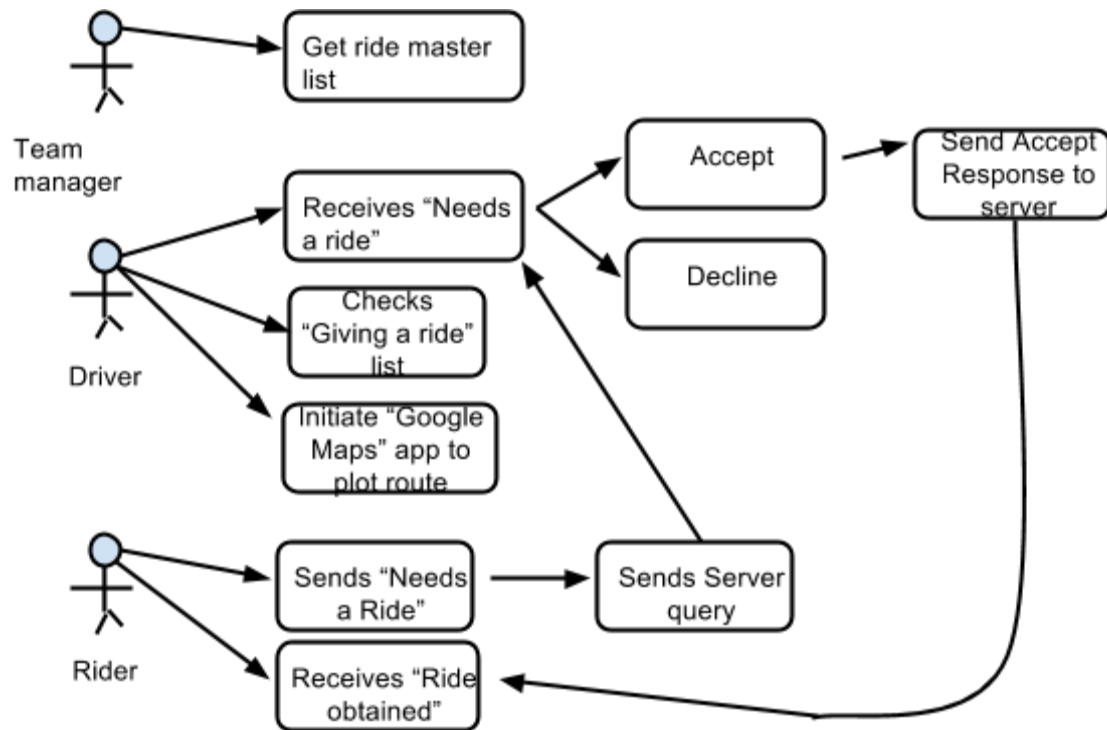
3.2.3 Inform users of where team members are via GM so that they have concise directions of how to pick up said team members.

Inform users who are receiving a ride of where the team member providing the ride is driving from.

- When a driver is found the rider can pull up the information on where the driver is coming from and estimated time to drive to their location.
- The driver is able to pull up a list of people they are giving rides to where they are located and the estimated time added to their trip.
- The driver can launch GM from the app to automatically plot their route to the event and any pickups en-route.

3.3 Behaviour Requirements

3.3.1 Use Case View



4 Other Non-functional Requirements

4.1 Safety and Security Requirements

- Encryption: We will generate a RSA key to secure user information such as username and password between the app and our database. Between our database and the TeamSnap server HTTPS will be used per the TeamSnap API.

Appendix A – Data Dictionary

(Future versions of this document will contain data dictionary items)

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist the Teaching Assistant to determine the effort put forth to produce this document>

Group Meetings:

9/19/14 - 10/17/14 3:00-4:00 Fridays: Discussed product design and SRS write up.