

Raider Point

Software Requirements Specification (SRS) Document

SRS-Scrummies(Final).docx

Version 2.1

February 5, 2018

Scrummies



1 Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft 1.0	Scrummies Team	Opening draft of software requirements specifications	01/22/2018
Draft 1.1	Scrummies Team	First turn-in draft	1/26/2018
Draft 2.0	H. Hoyat	First version of final SRS (corrections made from Draft 1.1)	2/1/2018
Draft 2.1	H. Hoyat	Final version of SRS	2/5/2018

2 Contents

3	Introduction	5
3.1	Purpose	5
3.2	Description	5
3.3	Overview	5
3.4	Glossary of Terms	6
3.5	Future Business Potential	6
4	Overall Description	7
4.1	Product Perspective	7
4.1.1	System Interfaces	7
4.1.2	Memory Constraints	7
4.2	Product Functions	7
4.3	Similar Systems	7
4.4	User Characteristics	8
4.5	User Objectives	8
4.6	Constraints	8
5	Specific Requirements	9
5.1	Functional Requirements	9
5.2	User Interface Requirements	13
5.2.1	User Interface: Graphical (GUI)	13
5.2.2	Diagnostics (Error Reporting and Usage Logs)	13
5.3	System Requirements	14
5.3.1	Hardware Interfaces	14
5.3.2	Communications Interfaces	14
5.3.3	Software Interfaces	14
5.4	Domain Requirements/Constraints	14
5.5	Non-Functional Requirements	15
5.5.1	Reliability	15
5.5.2	Availability	15
5.5.3	Security	15
5.5.4	Maintainability	15
5.6	Logical Database Requirements	16
6	Software Life Cycle Model	17

6.1	Choice of Software Life Cycle Model	17
6.2	Justification for Choice of Model	17

3 Introduction

3.1 Purpose

The purpose of this document is to give a detailed description of the “Raider Point” application. It will illustrate the purpose of the application, the application’s functionalities, and how the application will be developed. This document is primarily for the use of the current development team, any individuals who will work to develop the system in the future, and project stakeholders.

3.2 Description

Raider Point is an application that helps people find and remember events happening on the Mount Union campus. The application will consist of both a mobile application and a web portal.

The mobile application will let users search for and discover campus events using a mobile device, and set reminders for those events which will be sent to them as push notifications. Users will be notified of upcoming events happening within a ten-meter radius of their current location and which match their event preferences. Users can then choose whether or not they would like to set a reminder for it.

The web portal will allow authorized individuals to update and maintain information that will be used by the mobile application. Raider Point will also utilize a backend database stored on Mount Union’s web-server (the silver server) to hold any data needed to make the application functional.

For the time being, Raider Point will only be available for the Android environment.

A point of frustration for numerous members of the Mount Union community is the difficulty in learning about events taking place on campus, as well as the ability to spread the word about upcoming events. Raider Point will not only be an efficient way to keep up-to-date with campus affairs, but also provide a better method for campus organizations to notify others about their events. Since this application will be used on mobile devices, users will easily be able to check for campus events even when they are on the go.

3.3 Overview

This document presents the overall requirements and specifications for the Raider Point project. General ideas and concepts will start, followed by more specific requirements and techniques. The document will conclude with our method of approach to be used throughout this project life cycle.

3.4 Glossary of Terms

Agile Life Cycle Model - describes an approach to software development under which requirements and solutions evolve through the collaborative effort of self-organizing cross-functional teams and their customer/end users.

Android - a mobile operating system developed by Google.

Backend Database - a database that is accessed by users indirectly through an external application rather than by application programming stored within the database itself or by low level manipulation of the data.

Database Server - a computer program that provides database services to other computer programs or to computers, as defined by the client-server model.

Event Type - refers to the category which an event falls in (ex: academic, art, sport, etc.).

Graphical User Interface (GUI) - a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators

Model-View-Controller (MVC) framework - a software architectural pattern that divides a given software application into three parts to separate internal information from information presented to the viewer.

Megabyte - a multiple of the unit byte for digital information.

Raider Point - name of entire project. Used to refer to all software needed to make the application functional.

Scrummies - team name, quoted from Dr. Cindric when describing how a scrum life cycle model will be an ineffective way to approach a project.

Silver server - the web server used by Mount Union.

System Administrator - a person using Raider Point through the web portal.

User - a person using Raider Point through the mobile application.

User Type - refers to the category a user falls in (ex: student, faculty, staff, public).

Web Portal - specially designed website that brings information from diverse sources.

Web Server - a computer system that processes requests via HTTP, the basic network protocol used to distribute information on the World Wide Web.

3.5 Future Business Potential

Given the completion of this application, the University of Mount Union would be able to adopt either the idea or the actual application itself, and implement it into the Mount Union Mobile App.

4 Overall Description

4.1 Product Perspective

4.1.1 System Interfaces

Raider Point will consist of four system interfaces: one mobile application interface, one web portal interface, one GPS interface, and one database interface. The following list specifies each one in detail.

- Mobile Application - used to find campus events and receive event notifications. This graphical user interface will enable the user to interact with the Raider Point application.
- Web Portal - used to manage user and event information. This website will only be accessible to system administrators who will update Raider Point with new events and system information.
- GPS - used to determine the user's current location and any buildings on campus within ten meters of the user. The GPS application of the user's mobile device will be used by Raider Point to determine the events to which the user will be alerted.
- Database - used to store any data used by Raider Point (ex: user login information, user settings, event information). The data stored in the database will be pulled by the mobile application to display to the user, and will be accessed by the web portal to add and modify data.

4.1.2 Memory Constraints

Obtaining user location from a mobile device is often strenuous on the device's memory. To avoid overloading the operating system, the application will only be allowed to use twenty megabytes of memory while the application is running. The maximum amount of hard drive space will also be twenty megabytes.

4.2 Product Functions

Raider Point will allow users to search for various campus events, view detailed information about the events, set their event preferences and event notification settings, be notified of events happening near their current location, and choose which events they would like to be reminded of. It will send event reminders to users in the form of push notifications.

Raider Point will also allow authorized individuals to update and add data to the application's backend database through the web portal.

4.3 Similar Systems

There are numerous other systems similar to Raider Point, such as Urbanspoon which allows users to find restaurants near them, view restaurant information, and make reservations. This app is similar to the functionality Raider Point will be built for.

4.4 User Characteristics

The intended user for Raider Point is anybody in the Mount Union community who would like to keep updated on campus events, and any individuals tasked with maintaining the application's database. Users could include students, faculty, staff, and the public.

Since one of the goals of the application is to promote a positive user experience for any level of user, the educational level, experience, and technical expertise needed to use Raider Point from the mobile application side is the same as that for other simple location-based applications, such as Google Maps.

To update the backend database, even though the individual (system administrator) will need access to the web portal, the web portal will be user friendly and easy to understand, requiring minimal technical experience or expertise.

4.5 User Objectives

Raider Point is a mobile application that allows users to view and receive reminders of campus events. The following list includes the major features the system will support.

- Search and view information about campus events
- Receive reminders for campus events
- Be notified of campus events happening within a ten-meter radius of current location
- Update data from the backend database (system administrators only)

4.6 Constraints

The mobile application is constrained by the system interface to the GPS navigation system within the user's mobile device. Since there are multiple GPS systems of varying accuracy, and user movement within a confined time frame is unpredictable, the reliability of pinpointing a user's location will have to be monitored throughout the design process.

The Internet connection is also a constraining factor. Since the mobile application will retrieve data from the database over wifi, it is crucial to have an Internet connection for the app to function properly.

Both the web portal and mobile application will be constrained by the capacity of the backend database. Since data is shared between both the mobile application and the web portal, the database may be forced to queue incoming requests for data and increase the time it takes to display information to the user.

5 Specific Requirements

5.1 Functional Requirements

Priority Scale: Low (1) – Medium (2) – High (3)

1. Low: Items that can be eliminated should the need arise, without adversely affecting the product. These items are not urgent and not as important to the final product.

2. Medium: Items that are desired by the customer and/or users of the system, but that may be postponed until a future release. These items are not urgent and but are important parts of the final product.

3. High: Items that are mission critical and without which the system cannot function in a manner that is satisfactory to the customer. These items are urgently needed and important to the success of the final product.

Functional Requirement 1

Name: Locate User

Description: When a user is within ten meters of a campus building, the application will determine which building(s) the user is within range of.

Priority: 3

Technical Issues: The user will have to have GPS enabled on the mobile device they are using.

Risks: The user will not be able to see events happening around them.

Dependencies: None.

Functional Requirement 2

Name: Event Notifications

Description: When within range of a building, the user will be sent push notifications for any events happening in that building within the next 48 hours that match their event preferences.

Priority: 3

Technical Issues: The user will have to be connected to wifi and the database server.

Risks: The user will not be notified of campus events.

Dependencies: 1, 6, 8

Functional Requirement 3

Name: Notification Response

Description: When a user receives a notification for an event, the user will be able to view or dismiss the event. If the user chooses to view the event, the user will be able to view details about the event in the mobile application. If the user chooses to dismiss the event, the user will no longer receive notifications for that specific event.

Priority: 3

Technical Issues: The user will have to be connected to wifi and the database server to view the event.

Risks: The user will continue to receive notifications for the same events.

Dependencies: 2, 6, 8, 11

Functional Requirement 4

Name: Event Reminder

Description: Users will be able to receive reminders about events they wish to attend. These reminders will come in the form of push notifications to the user's mobile device and come at a specified time before the event starts. Users will be able to either view or dismiss the event reminder. Viewing the event reminder will allow the user to view details about the event within the mobile application. Dismissing the event reminder will close the push notification and the user will no longer receive a reminder for that specific event.

Priority: 3

Technical Issues: The user will have to be connected to wifi and the database server to view the event.

Risks: Users will not be appropriately reminded of events.

Dependencies: 6, 9, 11

Functional Requirement 5

Name: New User Account

Description: Users who do not currently have login credentials for the Mount Union portal will be able to set up a new account for themselves upon opening the mobile application. Setting up a user account will allow the user to create a username and password for themselves. Logging in will bring the user to their profile page.

Priority: 3

Technical Issues: The user will have to be connected to wifi and the database server.

Risks: A new user will not be able to use the mobile application.

Dependencies: None.

Functional Requirement 6

Name: User Login

Description: Users who have login credentials for the Mount Union portal will already have a user account in the system and can login using their Mount Union username and password. Logging in will bring the user to their profile page.

Priority: 3

Technical Issues: The user will have to be connected to wifi and the database server.

Risks: The user will not be able to use the mobile application.

Dependencies: None.

Functional Requirement 7

Name: Profile or User Account Page

Description: Page dedicated for showing saved events that the user is interested in. From here, the user can view information for their events.

Priority: 3

Technical Issues: The user will have to be connected to wifi and the database server.

Risks: The user will not be able to view their profile information.

Dependencies: 6, 11

Functional Requirement 8

Name: User Event Preferences

Description: Users will be able to choose which type of campus events they will receive event notifications for (i.e. sporting events, academic events, theatre events).

Priority: 3

Technical Issues: The user will have to be connected to wifi and the database server.

Risks: Users will receive notifications for events they are not interested in.

Dependencies: 6

Functional Requirement 9

Name: User Reminder Preferences

Description: Users will be able to choose when they will receive reminders for events.

Priority: 3

Technical Issues: The user will have to be connected to wifi and the database server.

Risks: Users will not be appropriately reminded of events.

Dependencies: 6

Functional Requirement 10

Name: All Campus Events

Description: Users will be able to see all campus events happening within the next 48 hours in a list format, organized by building.

Priority: 3

Technical Issues: The user will have to be connected to wifi and the database server.

Risks: Users will not be able to view all events they are allowed to attend.

Dependencies: 6

Functional Requirement 11

Name: Event Description

Description: When viewing an event, users will be able to see the event details including event time, location, and privacy.

Priority: 3

Technical Issues: The user will have to be connected to wifi and the database server.

Risks: Users will not know information about the event that is crucial to making it possible for them to attend.

Dependencies: 6

Functional Requirement 12

Name: System Administrator Login

Description: System administrators will be able to login to the web portal in order to access and update database information.

Priority: 3

Technical Issues: The system administrator will have to be connected to wifi.

Risks: The system administrator will not be able to use the web portal.

Dependencies: None.

Functional Requirement 13

Name: Update Event Information

Description: System administrators will be able to use the web portal to update campus event information.

Priority: 3

Technical Issues: The system administrator will have to be connected to wifi.

Risks: Campus event information will not be updated, giving inaccurate information to users.

Dependencies: 12

Functional Requirement 14

Name: Event Search

Description: Users will be able to search for specific events on campus, and filter their search based on specific buildings, event type, and event time.

Priority: 2

Technical Issues: The user will have to be connected to wifi and the database server.

Risks: Users will not be able to narrow down their search criteria for events.

Dependencies: 6

Functional Requirement 15

Name: Event Privacy

Description: Users will only be able to see events for which they are allowed to attend. For example, student users will be able to see events that are open to students, but not allowed to see events that are only open to faculty.

Priority: 2

Technical Issues: The user will have to be connected to wifi and the database server.

Risks: Users will be able to see all campus events, including ones that would be inappropriate for them to know about.

Dependencies: 6

Functional Requirement 16

Name: Master Notification

Description: Users will be able to turn on and off all event notifications with one button inside the application.

Priority: 2

Technical Issues: The user will have to be connected to wifi and the database server.

Risks: Users will have to turn on and off event notifications for each individual event type.

Dependencies: 6, 8

Functional Requirement 17

Name: Messages

Description: Users will be able to individually send other users in the system messages.

Priority: 1

Technical Issues: The user will have to be connected to wifi.

Risks: Users will not be able to communicate with each other through the mobile application.

Dependencies: 6

Functional Requirement 18

Name: RSVP

Description: Users will be able to RSVP for an event, indicating that they plan to attend. This RSVP will be reflected as a total in the event description.

Priority: 1

Technical Issues: The user will have to be connected to wifi and the database server.

Risks: Users will not be able to indicate that they would like to attend an event nor see how many other users on the platform plan to attend.

Dependencies: 6, 11

5.2 User Interface Requirements

5.2.1 User Interface: Graphical (GUI)

Raider Point will provide two different GUIs - one for users and another for system administrators.

The GUI for the users will be accessible from the mobile application. Upon opening the application, users will be greeted with a login page where they can either login to their existing account or create a new account for themselves.

After logging into their account, users will land on their profile page where they can see their events. In addition, users will be able to navigate to their settings page where they can change their user settings such as event type and event notification preferences.

A third page to which users will be able to navigate is a campus layout view page where all events the user is allowed to attend are listed by building. On that page, users will be able to select different events to view an event's information and set-up a reminder for it. Users will also be able to search for specific events by building, time, and event type.

The web portal will be the GUI used by system administrators to update and maintain data stored in the backend database which is used by the mobile application to present information to the user. This GUI will have a login page so system administrators can login. Once logged in, they will land on a webpage that will allow them to add, delete, and update database information such as campus events, user information, and user login information.

5.2.2 Diagnostics (Error Reporting and Usage Logs)

Errors will be reported within a user's perspective. In the case of a user's login failing or the user making an illegal command in the mobile application, an error message to the user will appear notifying them of the glitch.

Errors will also be reported to system administrators from the web portal side if they attempt to make an illegal command or their login fails.

5.3 System Requirements

A remote database will be required for this application to work. The database will hold all user and event information, and for the time being be held on the Mount Union silver server. This system will also need a user to have a mobile device with the Android environment installed on it.

5.3.1 Hardware Interfaces

Since neither the mobile application nor the web portal have any designated hardware, it does not have any direct hardware interfaces. The physical GPS is managed by the GPS application in the mobile device and the hardware connection to the database server is managed by the underlying operating system on the mobile phone and web server.

5.3.2 Communications Interfaces

The communication between the different parts of the system is important since they depend on each other. The GPS application must be able to communicate to the mobile application and the mobile application must be able to communicate to the backend database. In addition, the web portal must communicate to the backend database as well. However, in what way the communication is specifically achieved is not important for the system and is handled by the underlying operating systems for both the mobile application and web portal.

5.3.3 Software Interfaces

Users will need an operating system on their mobile device for the Android environment which will allow them to use the mobile application side of Raider Point. Appropriate Android versions include:

- Android 5.0/5.1 Lollipop or newer

System administrators will need an up-to-date PC operating system and web browser to have access to the web portal. Such operating systems include:

- OSX 10.7 or newer
- Windows 7 or newer

5.4 Domain Requirements/Constraints

Raider Point will only be usable on a mobile device running Android.

5.5 Non-Functional Requirements

5.5.1 Reliability

The mobile application will reliably notify users of campus events based on the users' current location for as long as the user has an Internet connection and GPS tracking enabled on their device. The application will also reliably connect to the database server where the database is held for the application. This will ensure users are presented with the most current and accurate information. System administrators will reliably be able to connect to the web portal for as long as the database server is running and there is an Internet connection.

5.5.2 Availability

The application will be accessible to those who are able to connect to the campus network such as Mount Union students, faculty, and staff. For future iterations of the application, the remote database will have to be moved to a production server (off of the silver server) for it to be accessible to those not connected to the campus network, such as the public.

5.5.3 Security

Because Raider Point will involve user specific information, security is a priority for this application. The following practices will be implemented to address such issues:

- Updated database software - the database software used will consistently be updated to avoid viruses and malicious injections.
- Access controls - users will only have read privileges for the backend database and system administrators will have both read and write privileges for it. These access controls will minimize the number of people capable of making changes to the database, therefore lessening the chance for unintended database mistakes.
- Cryptography - information sent and received from the database server will be encrypted so that in the event of communication being intercepted, user sensitive information will not be compromised.

5.5.4 Maintainability

Raider Point will be developed using an MVC framework to make the application easy to further develop.

5.5.5 Scalability

Raider Point will be designed with a database that can be extended to include any number of users and campus events.

5.5.6 Portability

The Raider Point mobile application will be designed for mobility, allowing users to use the app wherever they are on campus.

5.6 Logical Database Requirements

Raider Point requires an SQL database used to store data tables. These data tables will be populated with user specific information (i.e. names, usernames/passwords, event preferences, user type) and campus event information (i.e. event type, event time, event details, event location). This database will be stored on a remote server, accessible to the mobile application and web portal given an Internet connection. Since the mobile application is to be used by the users, the mobile application will only be able to read information from the database, while the web portal for system administrators will be able to read and write information to it.

6.1 Choice of Software Life Cycle Model

The Kanban life cycle model will be the overall approach to develop the Raider Point application.

6.2 Justification for Choice of Model

Kanban is a type of Agile life cycle model used for managing the creation of products with an emphasis on continuous delivery, while not overburdening the development team. This is achieved by using a Kanban board, a tool used to visualize work and optimize the flow of the work among developers. The board is separated into columns representing different stages of the development process, while cards representing different tasks to be completed are placed on the board depending on which stage of the development process the task is currently in.

Kanban is a good choice for Raider Point since the application includes many features, making it important for all team members to know exactly how far a feature has progressed in the development process. Kanban would also allow the team to release multiple iterations of the application by prioritizing different features to pull from the backlog first, ensuring all necessary functionalities make it into the application.