Software Requirements Specification

for

Student Engagement App

Version 1.0 approved

Prepared by Zachery Cox

Kyle McLendon

Farah Obaidullah

Mohini Patel

Kennesaw State University

September 27, 2016

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the Student Engagement App. It will illustrate the purpose and features of the application. This document is primarily intended to be proposed to a sponsor for approval and a reference for developing the first version of the application for the development team.

1.2 Document Conventions

GUI	*Graphical User Interface*
KSU	*Kennesaw State University*
RSS	*Rich site summary or really simple syndication*
SED	*Student Engagement Department*
UI	*User Interface*
UITS	*University Information Technology Services*

1.3 Intended Audience and Reading Suggestions

This document is more pertinent to the developers and clients interested in the project. However, marketing staff, project managers, and users can find need of some of the information in this document. This Software Requirements Specifications document contains information about the technical specifications of the program, what problems the application attempts to resolve, and its purpose.

For developers, the suggested sections are Section 1.1, Section 3, and Section 4.

For clients and users, the suggested sections are Section 1, Section 2, and Section 5.

For marketing staff, the suggested sections are Section 1, Section 4, and Section 5.

For project managers, every section in this document pertains to them.

1.4 Product Scope

The Student Engagement App is an application that provides a “Master Calendar” where the Student Engagement Department can easily view the current and future events that can be seen on the existing webpage. The staff will be able to add new events via the RSS feed which in return will be updated in the app. Additionally, some of the events will contain links to volunteer websites and other volunteer events requiring a forward address. This is beneficial as it allows students and

volunteers to have a well-organized and efficient way to sign up for events. The scope of who will be using it is geared for KSU faculty and staff and students. Since UITS did not approve of the department utilizing this application, this project will be a prototype that if successful, may be used in the future which is the goal of the project.

1.5 References

http://www.rssboard.org/rss-specification#ltcloudgtSubelementOfLtchannelgt

2. Overall Description

2.1 Product Perspective

For the scope of this project in its preliminary stage, it is its own self-contained product. It will pull from an RSS feed put forth by the Student Engagement Department and may contain redirect links to other websites or third party services, but this does not really make it a component of a larger system. This application will attempt to have most of its components coded, designed, and created by the developers.

2.2 Product Functions

- Ability for a user to add/edit/delete/view events (Determined by account permissions)
- The user can view the calendar in a web browser or via a mobile application
- Users can be redirected through the calendar to the appropriate location
- Users can view future events up to a set amount of time

2.3 User Classes and Characteristics

We will have only one type of user using this app. Because the most sensitive information that is listed within the app is RSS feed data, there is no need for any additional users at this time. Being the only user, it is up to the individual on what is added to the RSS feeds.

2.4 Operating Environment

Since this is a proof of concept, the operating environment does not need to be super advanced and the backend can be run from any computer running the newest version of windows. If the application is received well, upgrades may be necessary. At the time of this creation, the calendar app will be able to be edited and viewed in any modern web browser on any desktop or mobile device whether it is an Apple or Android product. There is no specific environment that is needed to run and maintain the application at this time.

2.5 Design and Implementation Constraints

The only constraints and limitations are if Kennesaw State University changed their RSS procedures. For the scope of the project, there are no hardware limitations or special software

needed to run. If the RSS feed is not working up to par, a simple MongoDB database may need to be created. For the function of the calendar, nothing too intensive will be needed as it will be view only for the majority of the users.

2.6 User Documentation

There is no user documentation needed since it will be for a select few people at first and will be unofficial.

2.7 Assumptions and Dependencies

One of the purposes of this project is to test ourselves at learning something new and to develop an application from scratch instead of building off an existing framework or premade library. Since this is the case, we may come across something that was originally planned that we may not be able to figure out how to do. We will of course do the best job we can to solve any issues that may occur. As far as any licenses, hardware, or software limitations there are none at the writing of this document, and as far as external factors are concerned, it just depends on if the many KSU departments continue to use their RSS feed for events to populate the calendar.

3. External Interface Requirements

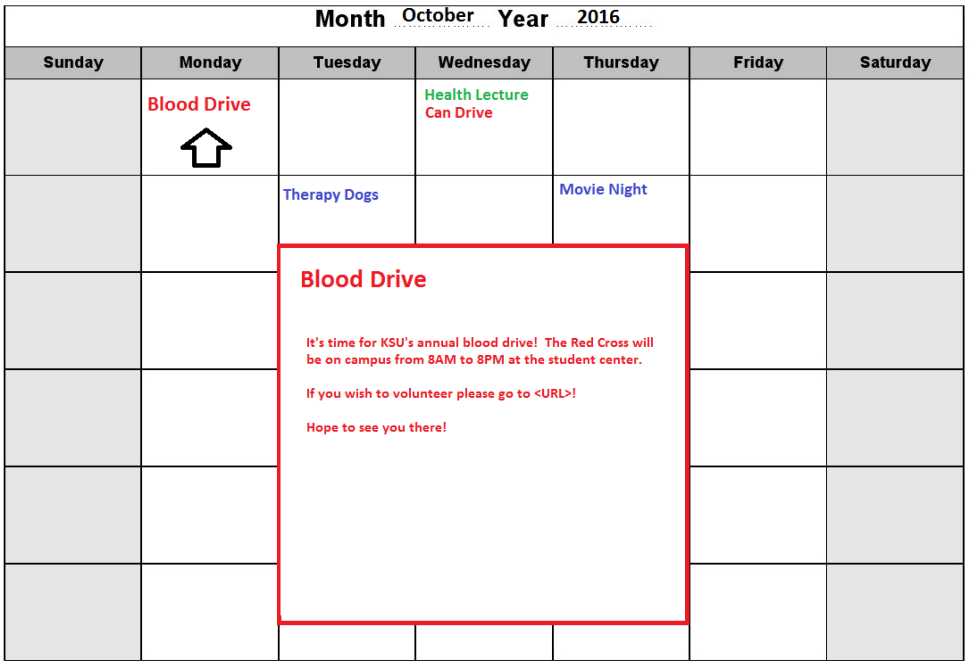
3.1 User Interfaces

The UI will have a simple design as we want it to resemble a normal desk calendar (or other calendar app) that is used. The main screen will be the month and all its days. The days will be made of boxes containing information on what events are occurring that day and will have a color code on the type of event. For example, a volunteer event will be a different color than a lecture type event. At first look, the event in each day will simply have the name of the event so as not to clutter the main screen. We have two design strategies for adding and editing events that we will compare as the project progresses. One design is to have a pop-in window that will appear if you want to view more details of an event or edit it. The other design would be to have all the information on one screen. In other words, the box of the day will still only have the name of the event but if the user clicks on the title or the box, the lower portion of the window will list all the details and editing information. Both design choices will be tested to see which is the more efficient.

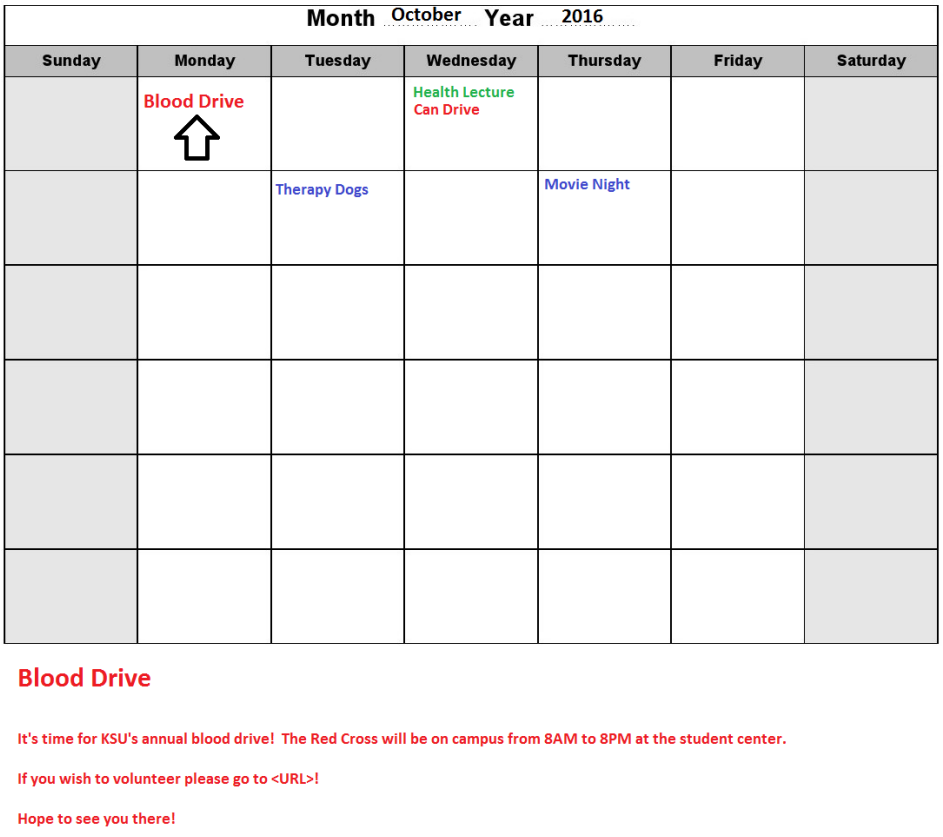
Sketch of what the main screen will look like in browser as well as on a mobile device.

Month October **Year** 2016

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
	Blood Drive		Health Lecture Can Drive			
		Therapy Dogs		Movie Night		



Sketch of what a “pop-in” window would look like when a user clicks on the title of the event



Sketch of what the “one screen” design would look like

NOTE: All design choices are subject to change and evolve as the project progresses and features are added.

3.2 Hardware Interfaces

Because of the requirements and the needs, the hardware that we are currently supporting are mobile devices, more specifically Android and iOS. In the future there may be a possibility for desktop support. Keeping an open mind for the future.

3.3 Software Interfaces

In effort to keep this app’s execution time low, we are going to be minimizing the external software source. We will be pulling data from the existing RSS feed that the three calendars have set up already. If there are any other calendars the user wants to import to the calendar, the user can do so.

3.4 Communications Interfaces

Within this app, we will be implementing a few communication functions. Redirects and RSS feeds will be our main communication interfaces. Links on the app can be redirected to the device’s

browser. While the RSS feed will allow the device to subscribe to such events via a lightweight http protocol.

4. System Features

4.1 Import Events from RSS feed

4.1.1 Description and Priority

High priority and will be part of the base version of app. Will allow us to create a single, master, calendar from such RSS sources. Allows us to grab events from other calendars.

4.1.2 Stimulus/Response Sequences

Planning on having this function whenever you start the app. We could also have a refresh button on the homepage to execute this manually. When this function is called, it will try to fetch the RSS feed data, pull it in, and update the master calendar.

4.1.3 Functional Requirements

An Internet connection will be required to sync new events. Errors that may occur should be handled as “no update”. If connection is not present, then nothing will update for new events cannot be received.

REQ-1: Internet Connection

4.2 Redirect Links

4.2.1 Description and Priority

We were asked to have redirect links to certain sites on our app. The links should point to blood drive and volunteering signup pages to allow uses to sign up for such events. This is a high priority and is required.

4.2.2 Stimulus/Response Sequences

After launching the app, on the homepage, we will either have the links displayed on the main page under the master calendar or we will have a tab for the user to click on to view links. It is not set in stone yet but we will see how the GUI behaves and decide accordingly.

4.2.3 Functional Requirements

Requirements that I can think of is sending the link path to the preferred browser on the device which the app may need access to do so. This may not be a requirement for we may be able to just allow the user to click on a panel/button and it behaves the same as the user clicking on a link in a text message.

REQ-2: Possible permissions

4.3 Contact Information

4.3.1 Description and Priority

As requested, we were asked to allow them to put contact information in the app. Such possible fields as: departments, names, addresses, office numbers, building, office number, email, etc. Because this app will be used as a resource, this feature holds a very high priority.

4.3.2 Stimulus/Response Sequences

The user will be able to view this data by either a link to another view that holds such information put in a tab, or will be at the bottom of the page. Implementation of this is still under discussion but it is still going to be up to how the GUI behaves and decide on the best design.

4.3.3 Functional Requirements

There are no requirements associated with this function.

4.4 Application Personalization

4.4.1 Description and Priority

This feature can be placed into two parts: personalization of the applications theme and events. Theme functionality will allow users to change the different colors within the app itself. The app will also need to have the ability to filter the events. Event filtering is very high priority, while the theme changing would be medium priority.

4.4.2 Stimulus/Response Sequences

Theme can be edited within a tab on the homepage. After pressing “Theme Settings”, it will allow a few options for different color schemes the user can use. The filter will have a button on the top of the screen where the user can decide on filtering options.

4.4.3 Functional Requirements

There are no requirements associated with this function.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- UI manipulation should take no longer than 2 seconds due to using JavaScript.
- RSS feed reader should take no longer than 120 seconds as that is the timeout exception within many similar function calls for RSS feed reader.

5.2 Safety Requirements

There shouldn't be any safety concerns

5.3 Security Requirements

For our base application, we will not be implementing any security functionality for we will only be grabbing public data from RSS feed. In a future functionality, such as creating local events, we may use local encryption.

5.4 Software Quality Attributes

Modular, portability, universal

5.5 Business Rules

All users can perform all functions for there is no administrator functionality aside from local.

6. Other Requirements

Unless pursued by the Student Engagement Department at KSU, this app will have no affiliation with Kennesaw State University. We are aiming to create this app on an iOS and Android platform only.

Appendix A: Glossary

Application
Executable software designed to fulfill a specific set of business functions individually or in cooperation with other applications.

Client
The user point-of-entry for an application.

Component
One of the parts that make up a system.

Constraint
A restriction, limitation, or regulation that limit a given course of action.

Database
A set of related data tables and other database objects that are organized as a group.

Encryption
The coding and scrambling of message to prevent unauthorized access to or understanding of data being stored or transmitted.

Hardware
Physical computer equipment and peripherals used to process, store, or transmit software applications or data.

Master Calendar
Layman's term for the application

MongoDB
An open source, document-oriented database designed with both scalability and developer agility in mind.

Permission
The authorities assigned to an end user to perform operations on data objects.

Requirement
A condition or capability needed by the customer to solve a problem or achieve an objective.

RSS "Rich Site Summary"
A format for delivering regularly changing web contents.

User
Someone who interacts with the application