Software Requirements Specification

for

gatorEvents

Version 1.0
September 9, 2015

IC3

Prepared by:

Gagan Sharma
Vineet Kaushik

Shubham Gupta
Chang Long

Jiayong Li
Zhaohe Xu

Table of Contents

Revision History

Name	Date	Reason for Changes	Version

1. Introduction

1.1 Purpose

gatorEvents is meant to serve the UF community by providing them with real time updates on events happening around the campus. It seamlessly binds the students, faculty and other staff members with the organizers of the events through a mobile application interface and a web back-end system

1.2 Document Conventions

This document follows no special typographical conventions. It is formatted in accordance with < *IEEE 830-1998*> and is simple enough to be read by anyone among the intended audience without detailed instructions.

The requirements of high priority features are largely inherited by other features. However, such instances have been clearly mentioned in the corresponding sections.

Some abbreviations used in the document are listed below along with their full-forms:
SRS – Software Requirements Specification
UF – University of Florida
NGO – Non- Governmental organization
OS – Operating System
FAQs – Frequently Asked Questions

1.3 Intended Audience and Reading Suggestions

The intended audience of this SRS are primarily users of the Android and Web application (students/faculty/staff, event organizers), developers, testers and the Project Manager/Scrum Master.)

Section Breakdown

- **Overall Description:** Overall Description provides basic information about all aspects of the project. This can be used by all readers of this SRS. Users can find a detailed manual under User Documentation.

- **System Features:** This covers a detailed description of various functions in the application and the web back-end. This is again relevant for all readers.

- **External Interface Requirements:** Explains the various interfaces over which project is spread. This section pertains to developers and testers.

- **Other Nonfunctional Requirements:** Requirements extending beyond interface descriptions can be found here. This is again meant for testers and developers.

- **Key Milestones:** This comprises the stages of development and milestones for the aforementioned stages. This is meant for the Project Manager/Scrum Master.

- **Key Resource Requirements/Other Requirements:** Key Resource Requirements section mentions the necessary requirements to run the application and the web interface. Other Requirements covers any secondary requirements that may enhance the performance/experience of the application. This is again meant for all readers.

1.4 Project Scope

Through gatorEvents the UF community, largely consisting of students/faculty/staff, can access information about all the events happening in UF. Events can be from a multitude of categories like culture, entertainment, sports, academic, career etc. The user can access all relevant information using an Android application. Events can be searched using keywords or by browsing the various categories displayed in the application. User can add categories into their list of favorites to receive notifications and updates about events from those categories.

The same application can be deployed for and used by other universities as well. This system can also be scaled to add other entities e.g. NGOs, housing communities, under its purview.

1.5 References

- www.gatorZone.com
- developer.android.com/guide
- https://www.google.com/design/spec/material-design/introduction.html
- http://calendar.ufl.edu/

2. Overall Description

2.1 Product Perspective

Information regarding events happening in UF is scattered over numerous channels e.g. sporting events are publicized on gatorZone, career/employment related information is provided on UF Career Resource and other events are usually posted on various Facebook pages.
gatorEvents addresses the problem of the non-availability of an easy-to-access-and-use events forum for the UF community.

2.2 Product Features

The Application will provide the user with a functionality of viewing and selecting events based on various categories listed. They can also mark categories as their favorites and will be notified for all events from those categories.
They can search the events they want to attend based on generic keywords, by the name of organizations/events or by the time slot that suits them.
A user can also sync or import events calendar (customized as per requirements). They mark themselves as attendees and get notifications for the same.

2.3 User Classes and Characteristics

This app will primarily be used by UF Students with Android Smartphones with basic experience in using such devices. The Web-Based back-end will be used by event organizers posting the events. They should have general experience in operating a computer and basic knowledge of the internet and filling online forms.
For security and spam control, the web based back-end will be accessible only to registered organizers.

2.4 Operating Environment

Users (students/faculty/staff) will require an Android smartphone. The minimum version of the OS supported by the application shall be Ice-Cream-Sandwich (Android OS v4.0.4) .The web application will be able to run on any modern browser e.g. Chrome, Firefox, IE9+.
Other requirements include access to the Internet on smartphone or a Tablet (via 3G, 4G LTE or Wi-Fi).
For Hardware requirement, user's phone should have, at minimum, 512MB of RAM and 10 MB of free disk space.

2.5 Design and Implementation Constraints

The application may face a limitation while implementing calendar syncing. If a user has disabled calendar syncing with any other application, they may not be able to import events calendar to their phone.
The application may not be able to notify user with the real time alerts if user loses Internet connectivity. Such alerts may include last minute change in event schedule, location or any other unforeseen circumstances.

2.6 User Documentation

Common user queries shall be answered in the form of FAQs. For further information and queries the user can send an email to Application support.

2.7 Assumptions and Dependencies

The assumption that may affect the usability of the app is that all the registered organizers will post all their upcoming events on gatorEvents. If they don't do so then the application stands completely useless.
Another assumption is that while a user is accessing our app, his connectivity with our Web-Server is not using more than 5 MB/s. This will make sure for 100% up-time for at least a million users.
For the Milestones added, the assumption is that the team will be able to master the required technologies well in advance as per the requirements of implementation. Any lag to achieve the same may hamper the progress and completion of the project.

3. System Features

The project is divided into two platforms: an Android app and web-based backend for posting events. The features under these platforms are:

3.1 User Registration

Android App

3.1.1 Description and Priority

The user of the android app shall have the ability to register themselves with the system. This is optional for the users of the app.

3.1.2 Stimulus/Response Sequences

Step 1: Launch the application from the home screen.

Step 2: Click the register button.

Step 3: User must fill in their information e.g. Name, UFL ID.

Step 4: Enter verification code.

Step 5: Jump to home page.

3.1.3 Functional Requirements

The sole functional requirement for this feature to work on the application is:
REQ-1: The application should be able to make an HTTP connection to the database.

Web

3.1.1 Description and Priority

Event organizers must register themselves with the system, using their UF IDs if applicable. Organizers must register before posting events.

3.1.2 Stimulus/Response Sequences

Step 1: Launch the web application using the link.

Step 2: Click the register button.

Step 3: Organizer must fill in their information e.g. Organization Name, Email ID.

Step 4: Verify using email.

Step 5: Jump to home page.

3.1.3 Functional Requirements

The sole functional requirement for this feature to work on the application is:

REQ-2: Connection to the internet.

3.2 Categories

Android App

3.2.1 Description and Priority

End users shall have the ability to browse through multiple categories under which the various events are listed e.g. Sports, Recreation, Club, Recreation etc.

3.2.2 Stimulus/Response Sequences

Step 1: Launch the application from the home screen.

Step 2: Click categories button to browse the page listing all the categories

Step 3: Select a category to view upcoming events.

3.2.3 Functional Requirements

The sole functional requirement for this feature to work on the application is same as **REQ 1.**

3.3 Post Events

Web

3.3.1 Description and Priority

This feature is the main functionality of the web back-end system. Event organizers can post their information through this function.

3.3.2 Stimulus/Response Sequences

Step 1: Once logged in, organizers click the "add event" button.

Step 2: Enter details in the template form as provided by the website e.g. title of the event, time and location, description of the activity and attachments (if any).

Step 3: Submit completed template.

3.3.3 Functional Requirements

The sole functional requirement for this feature to work on the application is same as **REQ 2**.

3.4 Favorites

Android App

3.4.1 Description and Priority

A secondary but important feature wherein users can mark their favorite categories. These categories are then listed under the favorite tag accessible through the main menu. Users should be able to "un-favorite" a category as well.

3.4.2 Stimulus/Response Sequences

Step 1: Click main menu icon on the navigation bar.

Step 2: Click categories button to browse the page listing all the categories. To view favorites go to Step 5.

Step 3: Swipe right and click to add category under favorites.(if category not favorite already)

Step 4: Swipe left and click to remove category from favorites. (if category is favorite already)

Step 5: Click main menu icon on the navigation bar.

Step 6: Click "Favorites" to view favorite categories.

3.4.3 Functional Requirements

Functional Requirements same as **REQ-1**.

3.5 Edit/Delete Events

Web

3.5.1 Description and Priority

This features allow organizers to edit the event they posted and the updated information will be shown on the user's mobile application. On the other hand, the web manager will have the access to delete a post if it is reasonable to do so.

3.5.2 Stimulus/Response Sequences

Step 1: Once logged in, organizers check their post history.

Step 2: Organizers can click "Edit" against the post they wish to edit and "Delete" against the post they wish to delete. For delete jump to Step 5.

Step 3: Edit event details as required.

Step 4: Click Submit.

Step 5: Click "Delete".

Step 6: User gets a notification if they have added the event to favorites.

3.5.3 Functional Requirements

The sole functional requirement for this feature to work on the application is same as **REQ 2.**

3.6 Notifications

Android App

3.6.1 Description and Priority

User gets system notifications on an urgency basis or a change of detail basis. Users can also access the notifications log within the app itself.

3.6.2 Stimulus/Response Sequences

Step 1: Click the main menu icon.

Step 2: Click notification button to browse the page listing all past notifications.

3.6.3 Functional Requirements

Same as **REQ-1**.

3.7 News Feed

Android App

3.7.1 Description and Priority

User gets an updated news feed based on events posted by organizers.

3.7.2 Stimulus/Response Sequences

Step 1: Launch the app from the home screen of the smartphone.

Step 2: Browse news feed on the home page of the app.

3.7.3 Functional Requirements

Same as **REQ-1**.

4. External Interface Requirements

4.1 User Interfaces

For the android application, the header shall contain the menu button, which on pressing will present a dropdown menu. This dropdown shall feature the salient features of the phone such as Categories and Favorites. It will also feature the "Help" section.

The above header will also contain the logo which shall work as the "Home" button i.e. clicking it will redirect the user to the main screen of the app.
The view group in the categories option shall contain buttons for every sub-category. Major categories will be placed on a slider that can be browsed sideways.

The design principles for the application will adhere to standard Material Design Guidelines set by Google for Android.

The web back-end registration page will feature a simple registration box, which will redirect to the sign up form on clicking sign up or the organizer's dashboard in case of successful login. Any errors during login will be displayed on the same page in red.

Screen Layout: Android App

Screen Layout: Web Application

4.2 Hardware Interfaces

The web back-end shall run on most modern browsers e.g. Chrome, Firefox, and Internet Explorer 9+. The browser can be on a desktop PC, Laptop Tab or phone.

The minimum android version for the mobile application is Android 4.0.4 (Ice Cream Sandwich). Any android based device capable of running Android 4.0.4 is sufficient for this application.

The data and control interactions will take place between the mobile and a server, and a web browser and the same server over TCP/IP using HTTP.

4.3 Software Interfaces

The application and the web interface are connected to a MySQL database running on a LAMP server. The minimum SDK for the Android application is API 15(ICS 4.0.4) and the target SDK is API 22 (Lollipop 5.1.1).

The application will be developed with the help of Oracle JDK and the Android SDK. The IDE being used to develop the application is Android Studio. The guides that will be referenced will be those available on developer.android.com.

The bulk of the data being exchanged will be text data in the form of JSON. There will also be some image data for the events. The data transfer will take place through TCP/IP.

4.4 Communications Interfaces

The application sends an email during the verification stage. The email is sent using an SMTP Server. This is common for the Android application and the Web application.
The web back-end shall run on most modern browsers such as Chrome, Firefox and IE9+.

The data will be wrapped in JSON during transfer. This transfer will take place using HTTP. Sensitive data such as passwords will be encrypted and salted to minimize risk in case of breach.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

100% uptime for entire UF community (Students, Teachers and Staff) is guaranteed. The bandwidth provided by the hosted Web Server is 6 TB/s. It can easily handle traffic for a million users, assuming each user is using 5 MB/s at one point.

The database of events will be normalized to prevent users from receiving redundant or irrelevant information for their query.

The user will be provided with a smooth interface to prevent any lag while using the application.

5.2 Safety Requirements

To ensure safety, organizers posting the events to the application shall have to register themselves via the web back-end. This obligation will make sure that only registered organizers can post events.

The Android application shall not exceed its limits while accessing data during operation. The application shall not access or tamper other data on the user's mobile or tablet. It cannot cause any damage to the phone or any other component.

The user should refrain from using the application while driving or performing any other activity where user's attention should not be diverted.

5.3 Security Requirements

The registered user data and their respective check-ins shall be stored and encrypted using the most reliable encryption algorithms. Additionally, passwords shall be salted with a random salt sequence.

Moreover, the application will provide the user with registration through their email and verify them to ensure that only those who signed up are attending the event. This will prevent false attendees through Spam.

5.4 Software Quality Attributes

Being a Web-Server application, the app will be available all times.
Maintainability will be made sure with regular update releases, like Content updates, Security updates etc.

The App will provide portability and can be configured for iOS, Windows Phones or future Android OS releases.

The applications shall be written in modules and will adhere to standard coding practices to enable future developers to add their work with maximum ease possible. This along with adequate testing will ensure the maintainability of the software.

The applications will designed in a manner that will be intuitive for the average user. The designers shall conform to standard rules for placement and function of various components of the applications such as menu icon, scroll bars, banners and other navigation features to maximize understandability and usability for the end users. Also, the web application will have a responsive design to enhance its operation on smaller screens.

6. Key Milestones

#	Milestone	Target Completion Date	Comments
1.	Initial Design, Primary features, Database Design	SPRINT 1	Complete design for the web app and initial design and code for primary features of the Android Application. Complete design of the database.
2.	Final Design, secondary features	SPRINT 2	Complete design for both apps. Complete code for secondary features.
3.	Security	SPRINT 3	Migrate to HTTPS.

7. Key Resource Requirements

Major Project Activities	Skill/Expertise Required	Internal Resource	External Resource	Issues/Constraints
Design UI	Designing	Gagan, Vineet, Alan, Shubham	Open source designing tools – GIMP.	NA
Design DB	Database Designing, SQL query writing	Gagan, Shubham, Zhaohe	SQL developer	NA
Develop web application	Web Development	Alan, Chang	Text Editor – Sublime, Python 3.2	Development skills not formerly acquired.
Develop Android Application	Android App Development	Gagan, Shubham, Zhaohe, Vineet	Android Studio, Android SDK, Java JDK	Development skills not formerly acquired.
Testing the applications	Ability to write test cases.	All team members		Ability to write test cases yet to be acquired.

8. Other Requirements

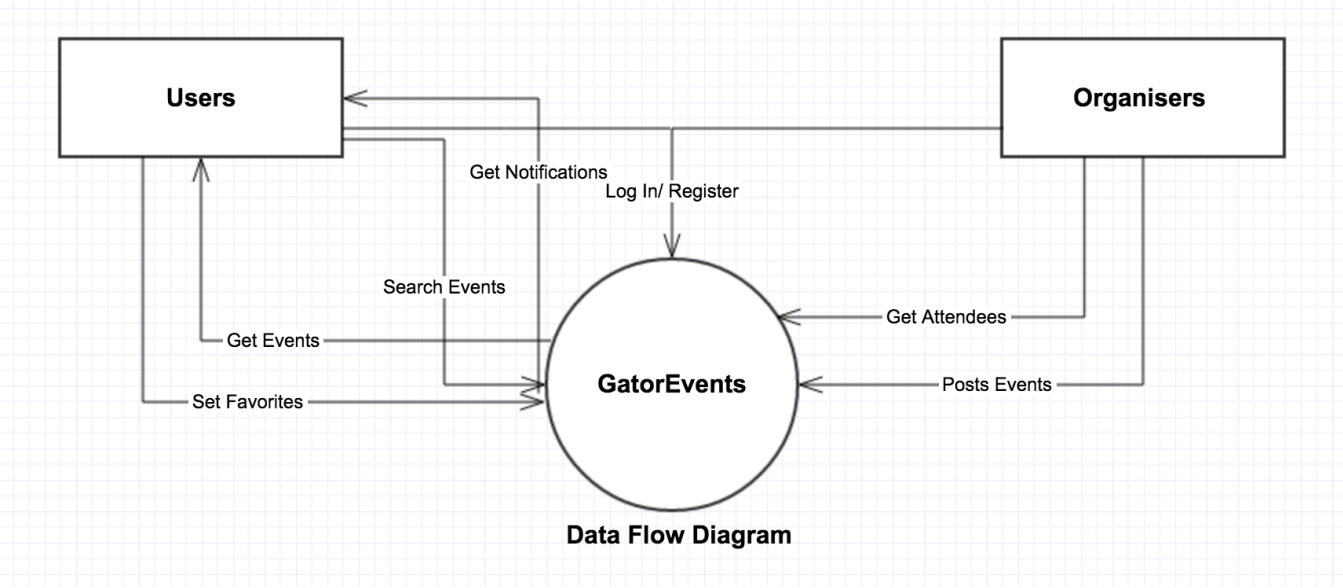
As mentioned previously the Android application and the web application are linked to a MySQL database operating on a LAMP server stack. While the application is available for download on Google Play Store, only those with a valid UFL gatorlink ID will be able to use the application. The source code will not be open for other developers to modify or peruse in its entirety. That right remains with the team members officially associated with the project.

Appendix A: Glossary

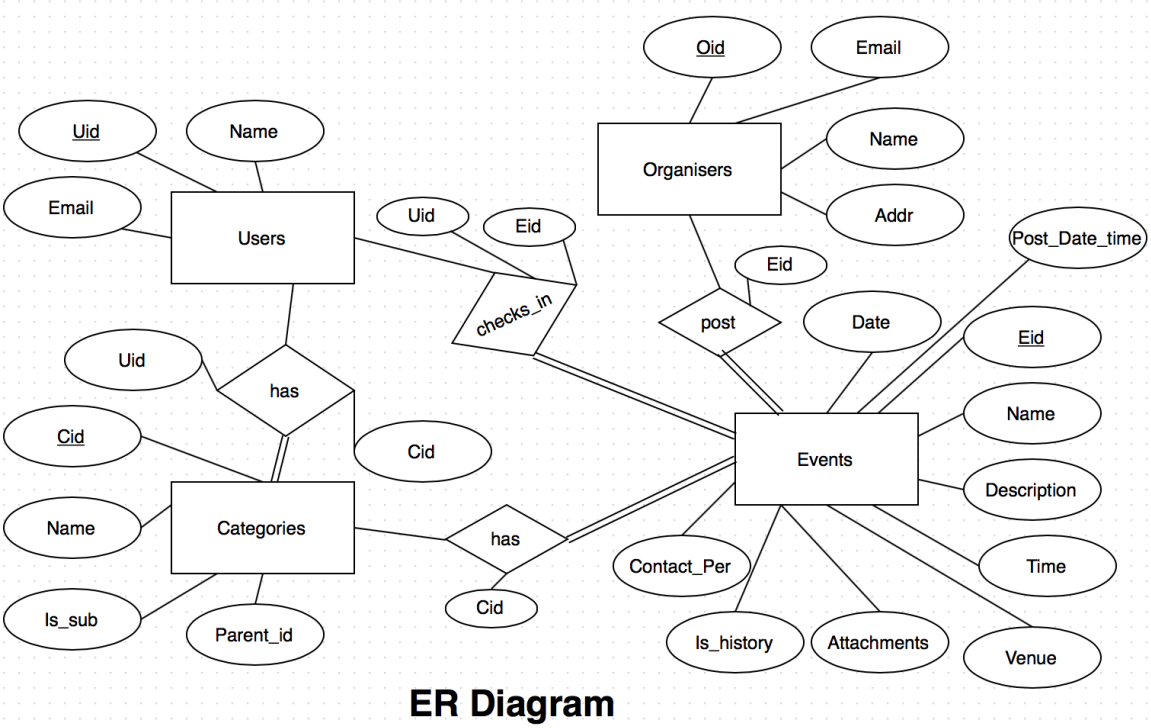
Some abbreviations used in the document are listed below along with their full-forms:

SRS – Software Requirements Specification
UF – University of Florida
NGO – Non- Governmental organization
OS – Operating System
FAQs – Frequently Asked Questions

Appendix B: Analysis Models



Data Flow Diagram



ER Diagram

Appendix C: Issues List

All team members have limited experience with development and designing such a system. Skills will have to be acquired on the fly.