Web Version Part 1

By Alan

Table of Contents

Table of Contents ii

Revision History iii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. System Features 4

3.1 System Feature 1 4

3.2 System Feature 2 (and so on) 4

4. External Interface Requirements 5

4.1 User Interfaces 5

4.2 Hardware Interfaces 5

4.3 Software Interfaces 5

4.4 Communications Interfaces 5

5. Other Nonfunctional Requirements 6

5.1 Performance Requirements 6

5.2 Safety Requirements 6

5.3 Security Requirements 6

5.4 Software Quality Attributes 6

6. Key Milestones 7

7. Key Resource Requirements 8

8. Other Requirements 9

Appendix A: Glossary 9

Appendix B: Analysis Models 9

Appendix C: Issues List 9

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The web version of the EventsUF is to allowed staffs of University of Florida, students from different organizations and shops located inside the campus (e.g. Starbucks, Subway) to post their recent activities or discount or special announcement. In other words, the web version is where they get the access to post all these information, and all eventually shown on the Android application news feed.

## Document Conventions

## Intended Audience and Reading Suggestions

The intended audience of the web version is mainly campus staffs, students and companies work staffs. The system and procedures shall be easy to understand. The web users only need to follow the constructions and fill all the necessary information when register for an account to post information through the web application.

## Project Scope

The purpose of the web version of EventsUF is to provide a convenient way for trustworthy information provider to post their information to all the other users. When register for an account, they will be asking to provide with some specific information to verify their identity, after which they will be allowed to log in to the system. The only problem a user need to worry about is which activity they actually would like to attend.

## References

# Overall Description

## Product Perspective

The web version is mainly responsible for verify identity, collection different information and then show all the information on the Android mobile application. The identity of the information provider should be checked prior to their access to post any information. At the same time, users can message the web managers in case there is any incorrect or outdated information was posted. The web manager will verify those posts manually and then decide what to do with them.

## Product Features

The web version contains four main features, which are user registration and log in, posting and deleting activities, receive messages (both users and web managers), and to manually modify information status (e.g. manager can delete a post, information can modify the content if necessary.)

## User Classes and Characteristics

There are two types of user classes for the web version. First of all, the ordinary information provider. They can only post their news and edit the information from themselves. For example, gator beach volleyball club post an information about their first group meeting for the fall semester. When the post this information, they can still change the content of their post. On the other hand, the web manager has the access to modify the status of all the information posted, like a post that is reported to be fake by other users, the web manager can delete the post from the database after they check the authentication of the post.

## Operating Environment

There is no specific limited hardware platform for the web version. It will be implemented based on the Flask frame work in python. We will run the website on Apache server, which will save us from tons of hardware and software environment problems.

## Design and Implementation Constraints

Main security consideration is that web system should check whether a user has logged in. If not, system should return the registration/log in page to the user and ask them to complete the log in or sign in process. There will be more details about the security consideration in the following sections.

## User Documentation

Since the web version of Events will be extremely to understand and there will be notifications and tips for users who firstly use the systems. Therefore there is no tutorials or user manuals..

## Assumptions and Dependencies

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## User Log in and Registration

3.1.1 Description and Priority

This feature is of high priority. Before an user is allowed to post any information through EventsUF, one must be identified it identity. Otherwise we cannot be sure about the authentication of the information. All users who try to post an information through the system must register or log in first, then will be allowed to do some further actions. This would an important step to make sure information in Events are from reliable people and organizations.

3.1.2 Stimulus/Response Sequences

When a user goes to the EventsUF web and try to post a new message to all the other users, the system will check whether the he has logged in. If not, a log in page should be returned to the user. The user will have to log in or register an account before he can take further actions.

3.1.3 Functional Requirements

When a new user tries to register an account of EventsUF, several critical information will be needed and must be verified before successfully create an account. First of all, if the user represents a student club, a campus organization or any other group that recognized by UF, they should be having a ‘.Edu’ email address. We will require: their email address (which will be tested and later will be taken as their username), their belonging organizations, their real name, password. But if the new user represents a shop inside the campus, which means they might not have a ‘.Edu’ email address, we will ask them to upload a picture of their working ID to the system, and web manager will verify their information manually.

## Post Information

3.1.1 Description and Priority

This feature is the main functionality of the whole system, of the second priority. User can post their information through this function.

3.1.2 Stimulus/Response Sequences

When a user log in to the system, then it can create a new post activity (not the http post activity). The system will have a templet and ask the user to fill in critical information, which are: title of the activity, organizations, time and locations, detail descriptions of the activity and attachments (if any). All the information mentioned above will be mandatory to fill in. After everything is set the information will be posted.

3.1.3 Functional Requirements

## Edit Post

3.1.1 Description and Priority

This feature is second priority. In real lives, events are likely to change their times or locations due to various unexpected factors. This features allow users to edit the information they post before and the updated information will be shown on user’s mobile application new feed. On the other hand, the web manager will have the access to delete a post if it is reasonable to do so.

3.1.2 Stimulus/Response Sequences

A user can check its post history. Once a change of time, location or other related information is needed, users can click the ‘edit’ button and the page will jump to the templet page with updated information so far in the blanks. The user can edit existed information and then submit the new version of the post to the server. Then the edit version of the news will be put in the new feed of the other users.

3.1.3 Functional Requirements

(I really don’t know how should I write this part……)

## Receive Report (For Web Manager)

3.1.1 Description and Priority

There is a chance that a user’s account is stolen or they simply made a mistake and post some fake or uncivilized information on the internet. In this case, any other users can report this post to the web manager. This feature is only for web manager and it is of second priority.

3.1.2 Stimulus/Response Sequences

When the web manager log in to the system, it will be a different page from the ordinary information provider. There will be a inbox of all reports from users. In each report, there will also be a link, generated by the post, to the reported post. The manager will have to check this information by himself then decide whether to delete this message or not.

3.1.3 Functional Requirements

External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

# Key Milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Milestone** | **Target Completion Date** | **Comments** |
| 1. |  |  |  |
| 2. |  |  |  |

# Key Resource Requirements

*<Identify the skill or expertise required for each major project activity. Identify any appropriate internal staff resources available. Indicate whether a resource only can commit part-time in the comments. If no internal resources are available, indicate the outside sources needed for the required skills. You may need more than one outside resource.>*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Major Project Activities | Skill/Expertise Required | Internal  Resource | External Resource | Issues/Constraints |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>