California State University, San Bernardino

School of Computer Science & Engineering

Software Requirement Specification

For

Development of a Cloud-Based Reward Application

by

Erwin Toni Soekianto

soekiane@coyote.csusb.edu

COMMITTEE

Dr. David Turner

Dr. Arturo Concepcion

Dr. Richard Botting

December 9, 2012

Table of Contents

Table of Figures

1. .... I ntroduction

1.1 Purpose

The purpose of this document is to present an overview of the project. It will explain the purpose and features of the system, the interfaces of the system, and what the system will do. This document is intended for the project advisor and committee members and will be proposed to the California State University San Bernardino, School of Computer Science and Engineering for its approval.

1.2 Abstract

The purpose of this project is to investigate the use of cloud-based services to implement software applications. For this purpose, a prototype of a reward application using badges will be developed to illustrate the emerging paradigms, including scalable, hosted runtime environments, NoSQL databases, and mobile access. Specifically, the project will illustrate the use of Node.js to implement server-side logic in Javascript. The project will also illustrate the use of MongoDB as a means to persist application data in a document-based database. The project will make use the Node.js runtime environment provided through by the Heroku company as a cloud service. It will also make use of the MongoDB database provided by the MongoLab company also as a cloud-based service.

In additional to Node.js and MongoDB, client-side web technologies may be used such as HTML (5), Javascript, CSS, and jQuery Mobile. We would also be using the cloud-based version control system GitHub, which is used to manage source code and as a means to deploy changes to a live application.

The goal of the application is to help organizations or groups interact with their members in a fun way. It attempts to keep members engaged by giving badges as rewards

for efforts or accomplishments. To enable viral marketing of the application, it will be integrated into one or more social networking sites such as Facebook or Twitter.

1.3 Definition, Acronyms and Abbreviation

Term	Definition
SQL	Structured Query Language
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
SDK	Software Development Kit
TBD	To Be Determined
MongoDB	MongoDB is scalable, high performance, open source NoSQL database
Node.js	Platform built on Chrome's Java Script runtime
Heroku	Cloud computing service vendor
sharding	A method of load balancing through horizontal partitioning of databases
jQuery Mobile	Unified HTML5-based user interface for mobile, built based on jQuery

1.4 Reference

[1] Heroku < http://www.heroku.com >

[2] MongoDB < http://www.mongodb.org/ >

[3] Node.JS < http://nodejs.org/ >

[4] GitHub < https://github.com/ >

1.5 Overview

A simple way to describe this application is that is is a fun and simple way to keep the members of any organization or groups engaged by giving rewards for their achievements using badges. The idea is to allow interaction between the organizers or badge issuers and the members in a fun way, and hopefully the rewards could encourage members to be more active and involved.

This application is based on the concept of gamification of reputation and social mechanics where it allows badge issuers to add badge into a badge collection, and give the badge to their members. Members would be able to show their earned badges via a social networking site. Each badge represents a reward for achieving a small or big achievement.

2. Overall Description

This section explains the components of project.

2.1 Technology Perspective

2.1.1 Cloud Computing

Cloud computing is a model which makes use of computer hardware and software that are accessed through the Internet as services. There are several choices of cloud computing services available, but for this project we choose the one provided by Heroku, the cloud computing partner of Facebook. [1]

The reasons are that Heroku lets you use and publish an application that people can use right away with no cost and obligation, and you can take advantage of the same scalable technologies that Facebook applications are built on, and attain a similar level of reliability, performance and security.

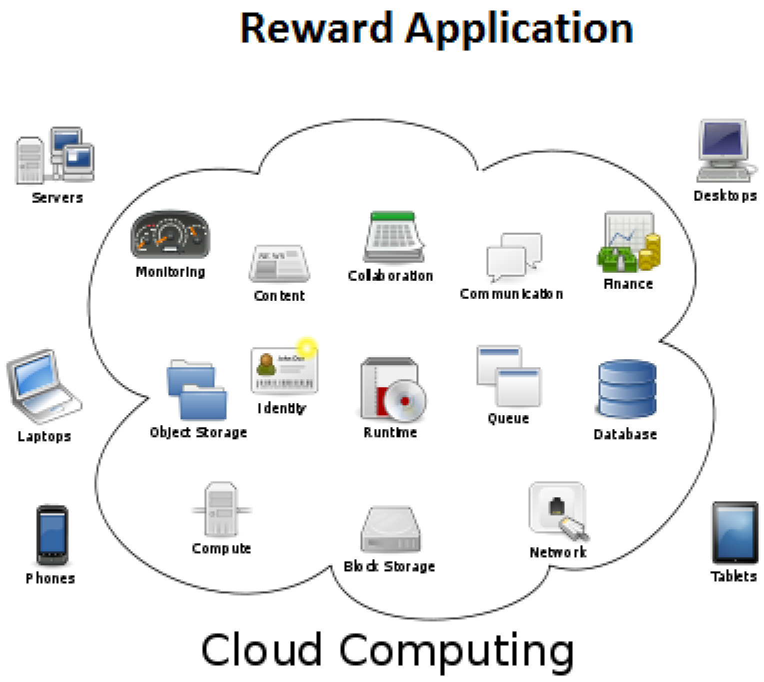


Figure 1 Cloud Computing Description

2.1.2 Node.js Language

Cloud-based services support apps written in several different programming languages, such as Java, Python, PHP, Javascript, Ruby and many more. For this project we would use Javascript running in a Node.js context. Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices. [3]

2.1.3 NoSQL Database

There are many different types of cloud-based datastore services to choose from. For this project we will use MongoDB, as it works well Node.js and Heroku. MongoDB is a scalable, high-performance, open source, NoSQL document-based database. MongoDB features include document-oriented storage, indexes, replication, high availability, auto-sharding, and querying. [2]

2.1.4 Git

Git is a distributed version control system. This project uses git with GitHub, a cloud-based provider of remote git repository storage. Heroku uses git as a means to deploy web applications to its servers. Git allows easy creation of testing, staging, and production versions of the application. [4]

2.2 Product Perspective

2.2.1 System Interfaces – Deployment Diagram

This following diagram is the deployment diagram of application.

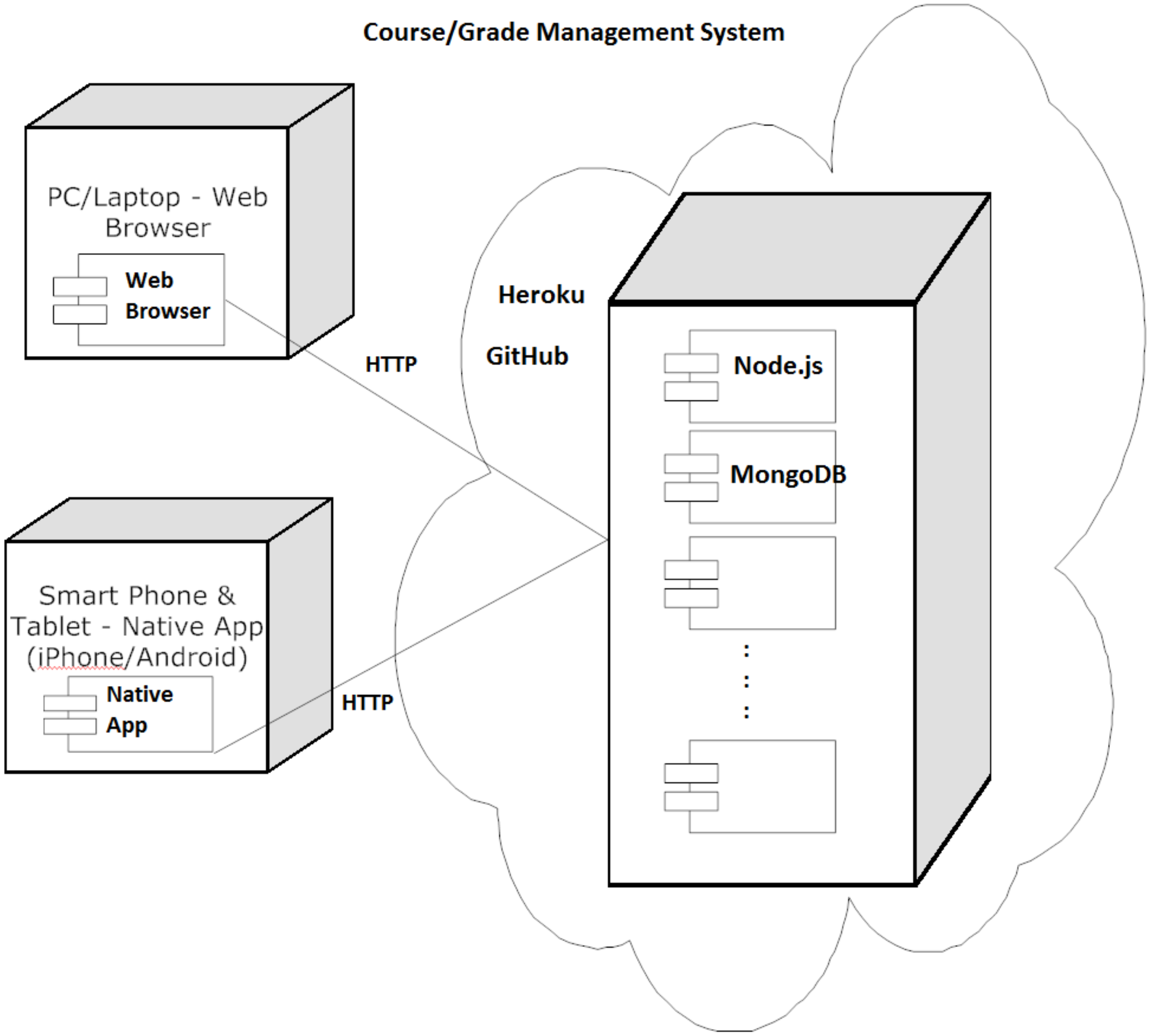

Course/Grade Management System
PC/Laptop - Web Browser
Web Browser
HTTP
Smart Phone & Tablet - Native App (iPhone/Android)
Native App
HTTP
Heroku
GitHub
Node.js
MongoDB

Figure 2 Deployment Diagram

2.2.2 User Interfaces

This application consists of three user interfaces. The first interface is the website itself, which is mobile device friendly and will serve to provide the navigation between all of other interfaces in the application. The second interface will be the badge issuers' interface where it will show the badges collection and member list. And the third interface will be the badge recipients' interface where it will show badges earned and buttons to share to social networking sites.

2.2.3 Software Interfaces

This application requires a web browser to be viewed. Only web browsers that support HTML5, javascript and graphic-enabled are supported. Polyfills will be used to emulate HTML5 features that this application relies on to support some older browsers that do not support HTML5.

2.2.4 Communication Interfaces

This application will be implemented using JavaScript and possibly HTML5. All client-side communication with the application shall use HTTP from the client's internet browser. Server-side communication will be controlled by the web-server.

2.2.5 Memory

This application will require 64 MB of RAM to be viewed.

2.2.6 Operation

This application will be maintained during the development of the application and operated 24 hours a day, 7 days a week throughout the year. The maintenance will be conducted by the author, and the hosting will be provided by Heroku.

2.2.7 Site Adaption Requirements

The user's device must have internet connection enabled and web browser to access the application.

2.3 Product Functions – Use Case Diagram

This is the use case diagram of the application below.

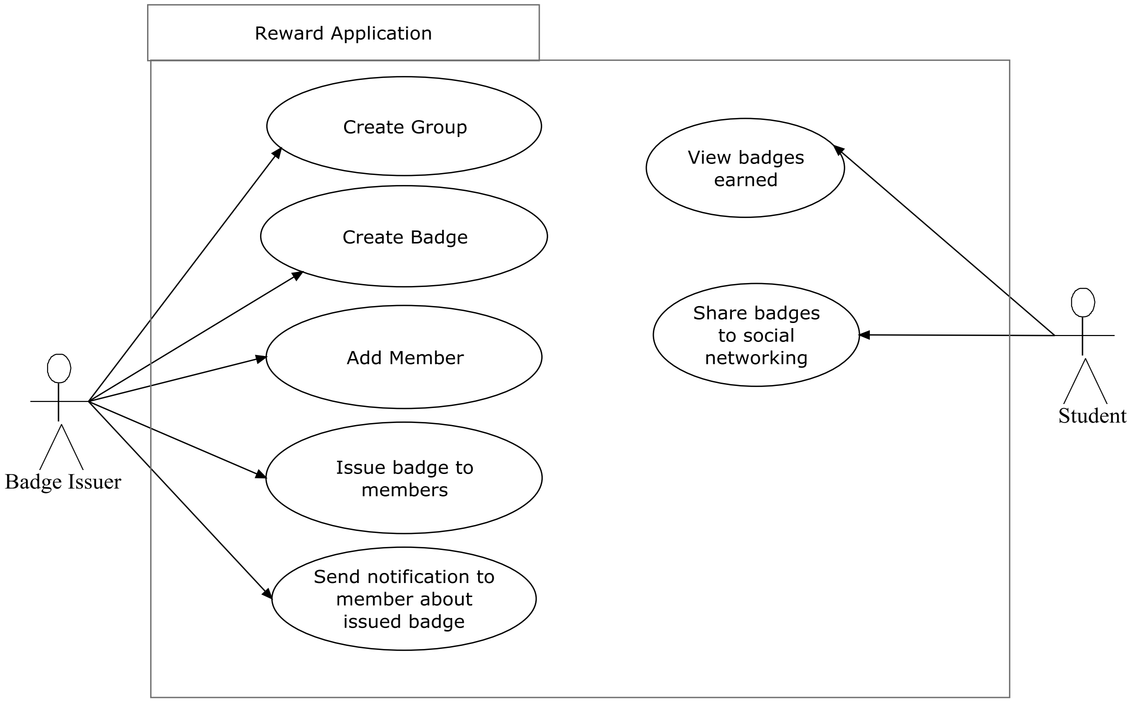


Figure 3 Use Case Diagram

2.4 User Characteristic

The target audience of this application is any person at the age of 13 and older. This application is meant to visually display badges earned and badges collection.

2.5 Constraint

This application shall be functioning and deployed for presentation for the users by end of the Master project. All program's faults must be identified and fixed prior to the final demonstration; a presentation of its progress will be delivered during the course of the project duration.

2.6 Assumption and Dependencies

This application assumes that Heroku will continue support compatibility of Node.js and MongoDB in their cloud computing services.

3. Specific Requirement

3.1 External Interfaces Requirement

The logo for this application is not decided yet. It will be decided and updated sometime before the completion of this project.

3.1.1 User Interfaces

When users first visit the application, there will be header on the top, footer on the bottom and main frame at the center of the page. The header will consist of the title of the page and back or logout button whenever necessary, footer would consist of action buttons available, and the content would show the information such as badges, members and menu.

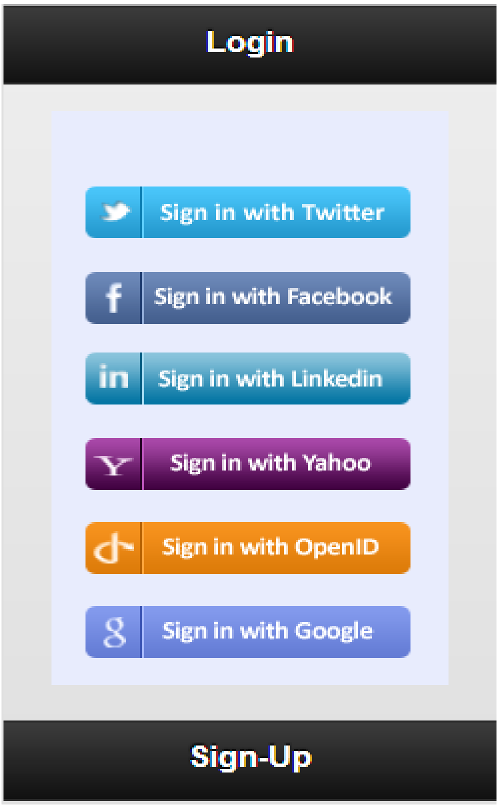
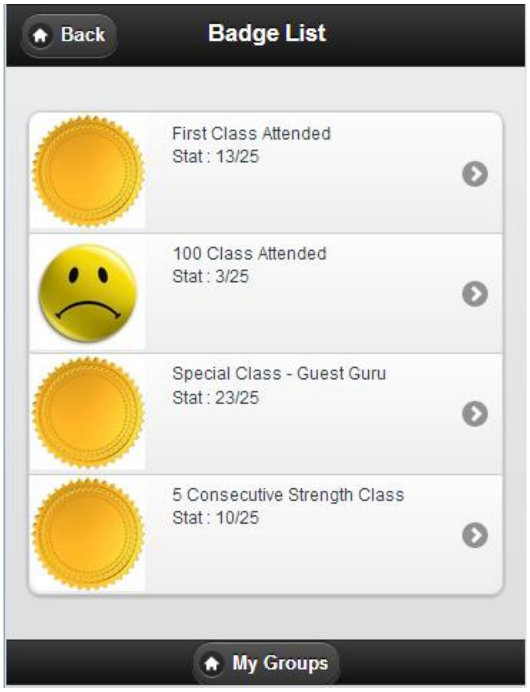


Figure 4 Login Page



Badge List

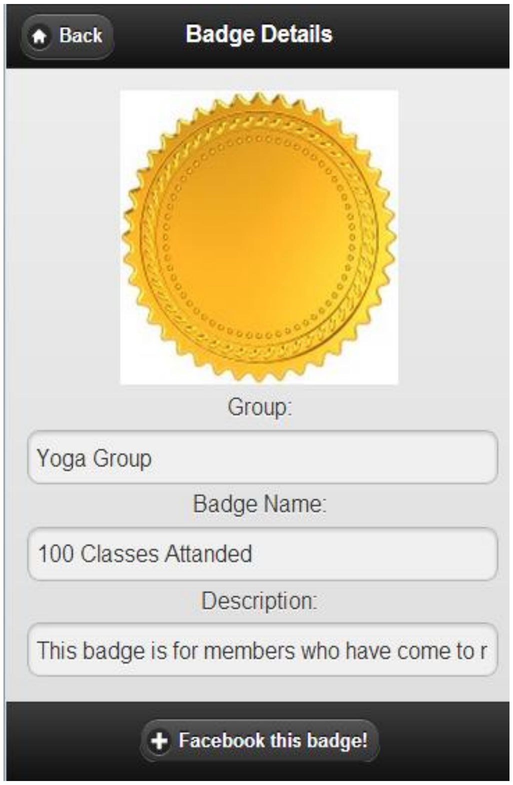
First Class Attended
Stat : 13/25

100 Class Attended
Stat : 3/25

Special Class - Guest Guru
Stat : 23/25

5 Consecutive Strength Class
Stat : 10/25

Figure 5 Member Page – Badge List



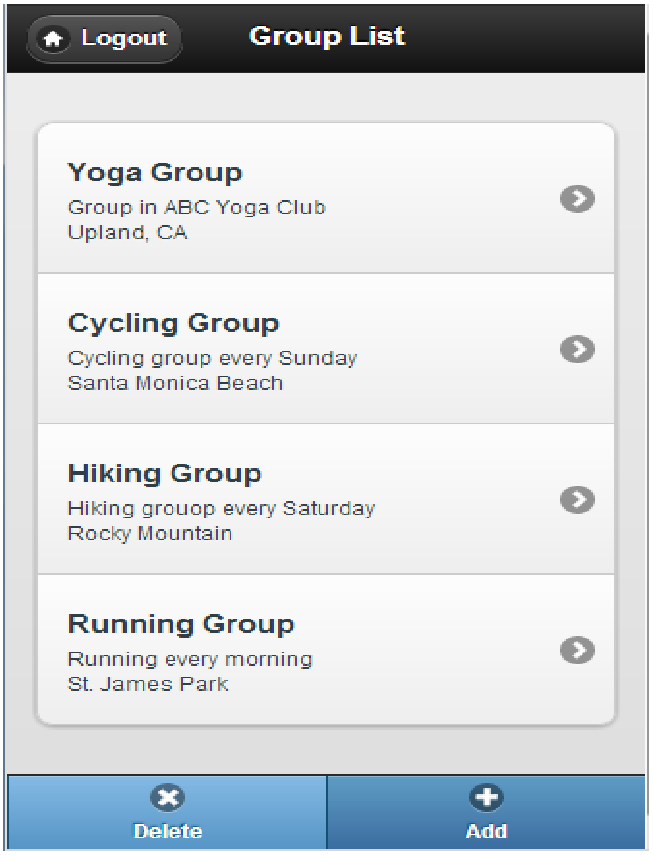
Badge Details

Group:
Yoga Group

Badge Name:
100 Classes Attanded

Description:
This badge is for members who have come to r

Figure 6 Member Page – Badge Details



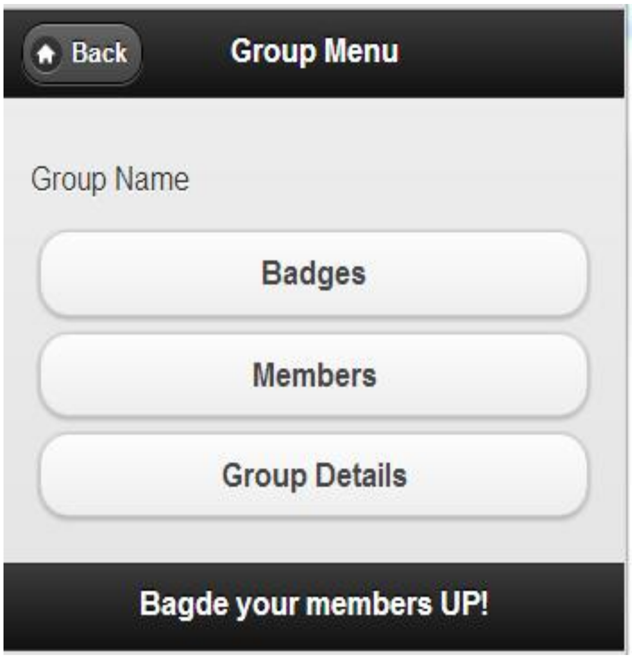
Yoga Group
Group in ABC Yoga Club
Upland, CA

Cycling Group
Cycling group every Sunday
Santa Monica Beach

Hiking Group
Hiking grouop every Saturday
Rocky Mountain

Running Group
Running every morning
St. James Park

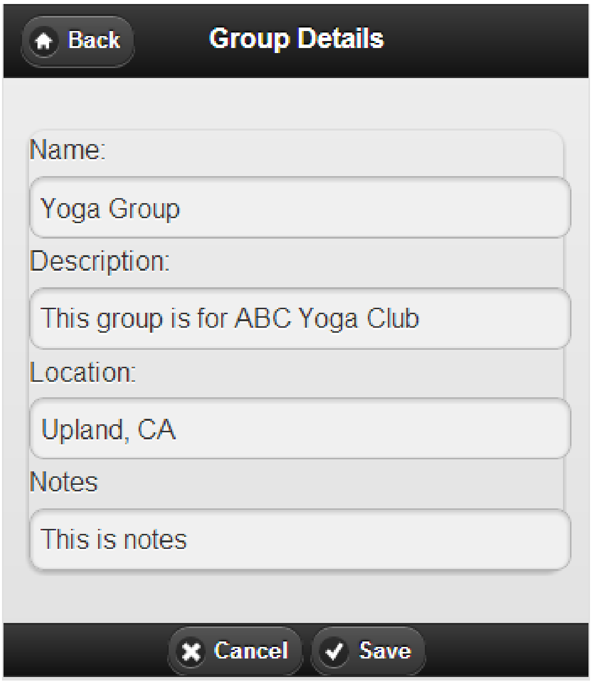
Figure 7 Issuer Page – Group List



Group Name

Bagde your members UP!

Figure 8 Issuer Page - Group Menu



Group Details

Name:

Yoga Group

Description:

This group is for ABC Yoga Club

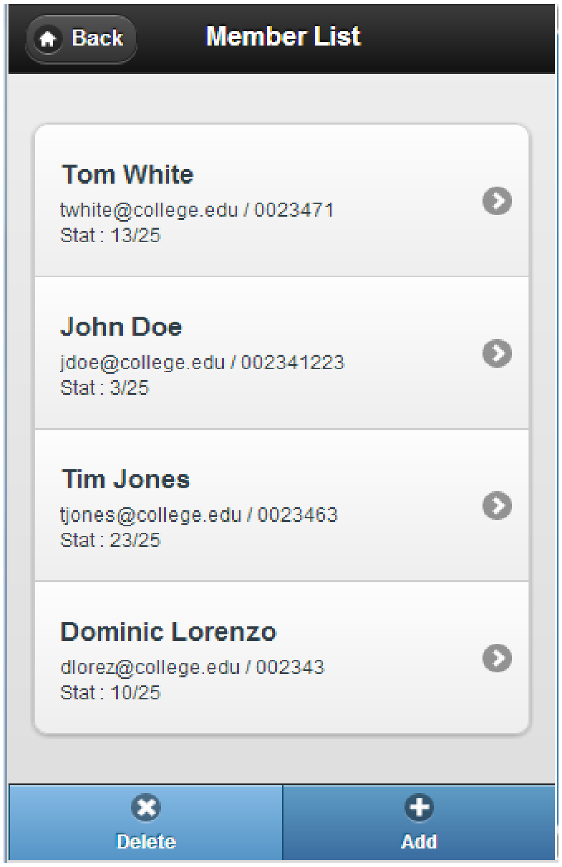
Location:

Upland, CA

Notes

This is notes

Figure 9 Issuer Page - Group Details



Member List

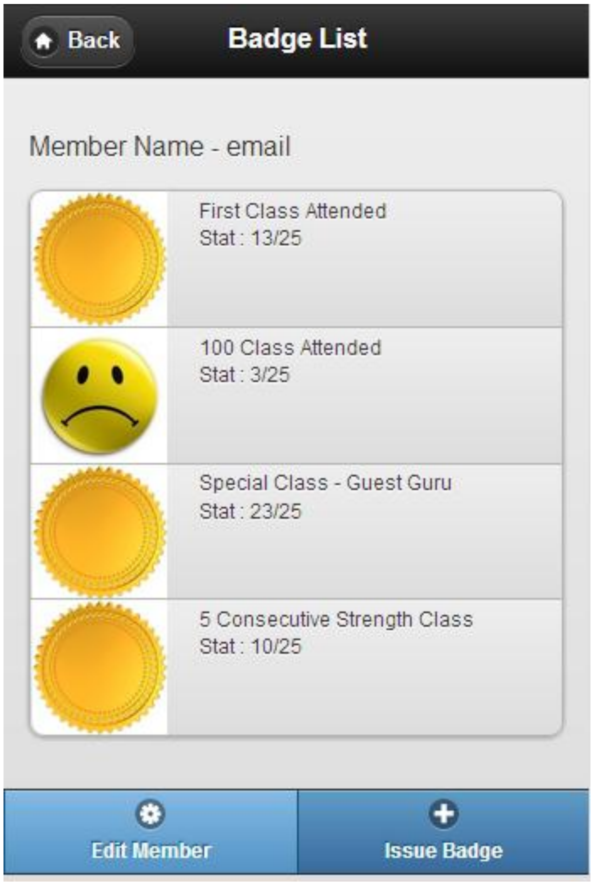
Tom White
twhite@college.edu / 0023471
Stat : 13/25

John Doe
jdoe@college.edu / 002341223
Stat : 3/25

Tim Jones
tjones@college.edu / 0023463
Stat : 23/25

Dominic Lorenzo
dlorez@college.edu / 002343
Stat : 10/25

Figure 10 Issuer Page – Member List



Member Name - email

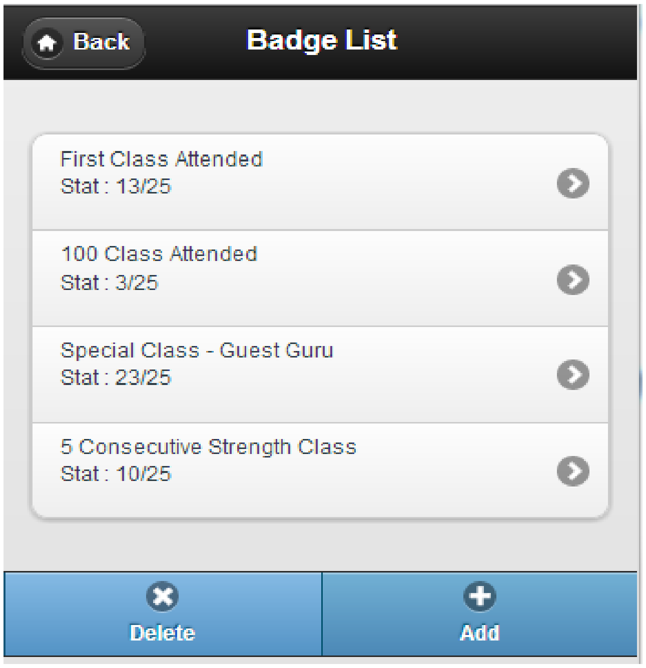
First Class Attended
Stat : 13/25

100 Class Attended
Stat : 3/25

Special Class - Guest Guru
Stat : 23/25

5 Consecutive Strength Class
Stat : 10/25

Figure 11 Issuer Page – Member Badge List



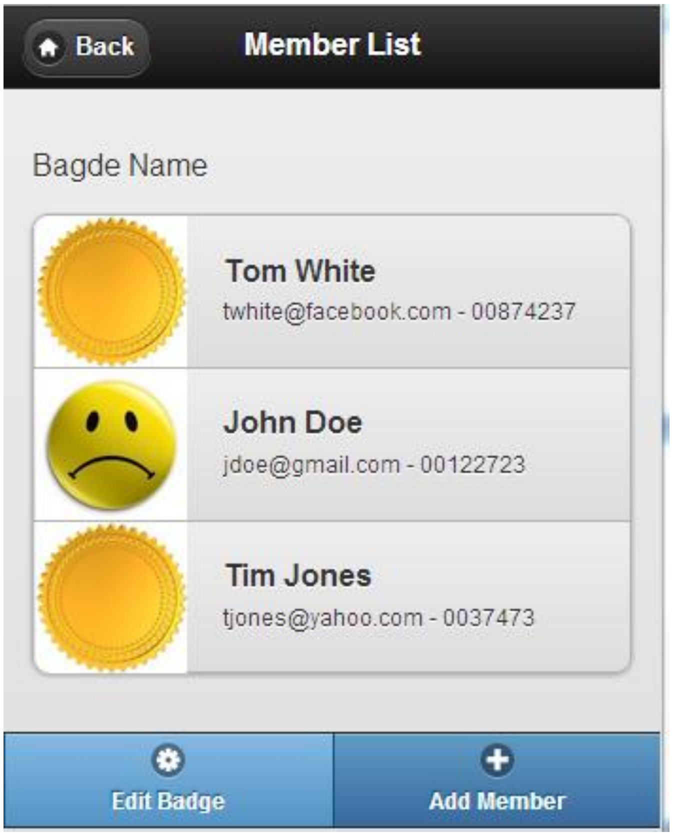
First Class Attended
Stat : 13/25

100 Class Attended
Stat : 3/25

Special Class - Guest Guru
Stat : 23/25

5 Consecutive Strength Class
Stat : 10/25

Figure 12 Issuer Page – Badge List



Bagde Name

Tom White
twhite@facebook.com - 00874237

John Doe
jdoe@gmail.com - 00122723

Tim Jones
tjones@yahoo.com - 0037473

Figure 13 Issuer Page – Badge Member List

3.1.2 Hardware Interfaces

3.1.2.1 Server Side

The application will be hosted in Heroku server. The web server is listening on port 80.

3.1.2.2 Client Side

The system is a web based application; clients are requiring using a high speed Internet connection and using a up-to-date web browser.

3.1.3 Software Interfaces

3.1.3.1 Javascript

JavaScript will be implemented throughout the website in order to display the correct feature the user requested.

3.1.3.2 HTML5

HTML5 may be implemented throughout the website in order to display the correct feature the user requested.

3.1.4 Communication Interfaces

This application is designed to be viewed on any internet browser provided that:

1. JavaScript is enabled
2. Images are enabled
3. HTML5 is compatible

Performance may vary slightly between browsers. However, the functionality of the site should not be impaired.

3.2 Functional Requirements

The function specified on this on this section directly correspond to work that will be conducted on this project.

3.2.1 Create Group

This functionality allows organizers or badge issuers to create groups, which will have a set badge collection.

3.2.2 Create Badge

This functionality allows badge issuers to create badge to be added to badge collection. A badge would have at least badge name and description.

3.2.3 Add Member

This functionality would allow organizers or badge issuers to add members into the groups as badge recipients. Member would have at least email address and name.

3.2.4 Issue Badges to Members

This functionality would allow badge issuers to issue badges to members.

3.2.5 Send Notification to Member about issued Badge

This functionality would allow badge issuers to send notification in the form of email, that they have issued new badge for the members.

3.2.6 View Badge Earned

This functionality would allow badge recipients to view all the badges that they have earned.

3.2.7 Share Badge to Social Networking

This functionality would allow badge recipients to share the badge to their social networking site such as Facebook or Twitter.

3.3 Performance Requirements

This application is going to be hosted in the Heroku cloud server, so the performance of this application would be high.

3.4 Design Constraints

TBD

3.5 Software System Attributes

The author will keep coding standard with proper commenting and documentation.

3.6 Other Requirements

TBD