Software Requirements Specification

For

GoChron

Version 1.0 approved
September 11, 2015

Prepared by:

Nagapandu Potti
Akshitha Ainapuri

Aman Singh
Rakesh Dammalapati

Shantanu Kande
Jithendra Yella

TABLE OF CONTENTS

Revision History

Name	Date	Reason For Changes	Version
goChron SRS	09/08/15	NA	1.0

1. INTRODUCTION

1.1. PURPOSE:

This software requirement specification document provides the software functional and nonfunctional requirements of the goChron application. All the requirements specified here are committed for the release 1.0. This document is meant to be used by the goChron team that will develop, implement and verify the correct functioning of the system.

1.2. DOCUMENT CONVENTIONS:

- **Calendar:** Schedule of all the events happening at UF among students, clubs, organisations and departments; including administrative deadlines.
- **Event:** Planned gathering of set of people for a meaningful purpose such as entertainment, knowledge sharing and many such.
- **Admin:** Individual or a organization who has the privilege to create an event.
- **Group:** A user can create a group if he wants to post regular events pertaining to a organization.
- **Subscription:** Follow a particular verified group or request for a notifications with a particular keyword.
- **Notifications:** Anytime admin adds any event, subscribers will receive formal notification of the announcement.
- **Alerts:** Notification(via Email) is sent to the user when an event that matches his/her interests is added to the calendar.
- **Public Event:** Event schedule and details can be accessed by everyone and requires no approval to register the event.
- **Closed Event**: Event Schedule and details can be accessed by everyone but requires admin's approval to register the event.
- **Private Event:** The event will not be seen on the calendar to the public, only the admin who created the event can invite others to join.

Every requirement stated in this software requirement specification document does have its own priority.

1.3. INTENDED AUDIENCE AND READING SUGGESTIONS:

This document is intended for all individuals participating in and/or supervising the goChron project. Readers interested in a brief overview of the product should focus on the rest of Part 1 (Introduction), as well as Part 2 of the document (Overall Description), which provide a brief overview of each aspect of the project as a whole. These readers may also be interested in Part 6 (Key Milestones) which lays out a concise timeline of the project.
Readers who wish to explore the features of goChron in more detail should read on to Part 3 (System Features), which expands upon the information laid out in the main overview. Part 4 (External Interface Requirements) offers further technical details, including information on the user interface and software platforms on which the application will run.
Readers interested in the non-technical aspects of the project should read Part 5, which covers performance, safety, security, and various other attributes that will be important to users. Readers who have not found the information they are looking for should check Part 8 (Other Requirements), which includes any additional information which does not fit logically into the other sections.

1.4. PROJECT SCOPE:

goChron aims at simplifying the way students could keep track of events happening at UF by bringing onto one single platform in a single view with much enhanced features such as subscription to events, suggestions on what to attend, alerts to remind you, notifications when an event that interests you is added, showing currently trending events and sync option with third party calendars! In addition, it could give option to create events such as Public/Private/Closed by giving control to the user with filters/abstraction while creating the events.

1.5. REFERENCES:

- https://www.cise.ufl.edu/class/cen3031sp13/SRS_Example_1_2011.pdf
- http://usabilitypost.com/2009/04/15/8-characteristics-of-successful-user-interfaces/

2. OVERALL DESCRIPTION

2.1. PRODUCT PERSPECTIVE:

goChron is a replacement for UF Calendar available at calendar.ufl.edu and is essentially a web application that finds itself in use in universities which have numerous events going on at a given time. This project acts as a replacement in a way that integrates all the events related to different organizations and clubs and makes them available at one single place and also maintaining the security of the events at the same time by controlling the events' visibility.

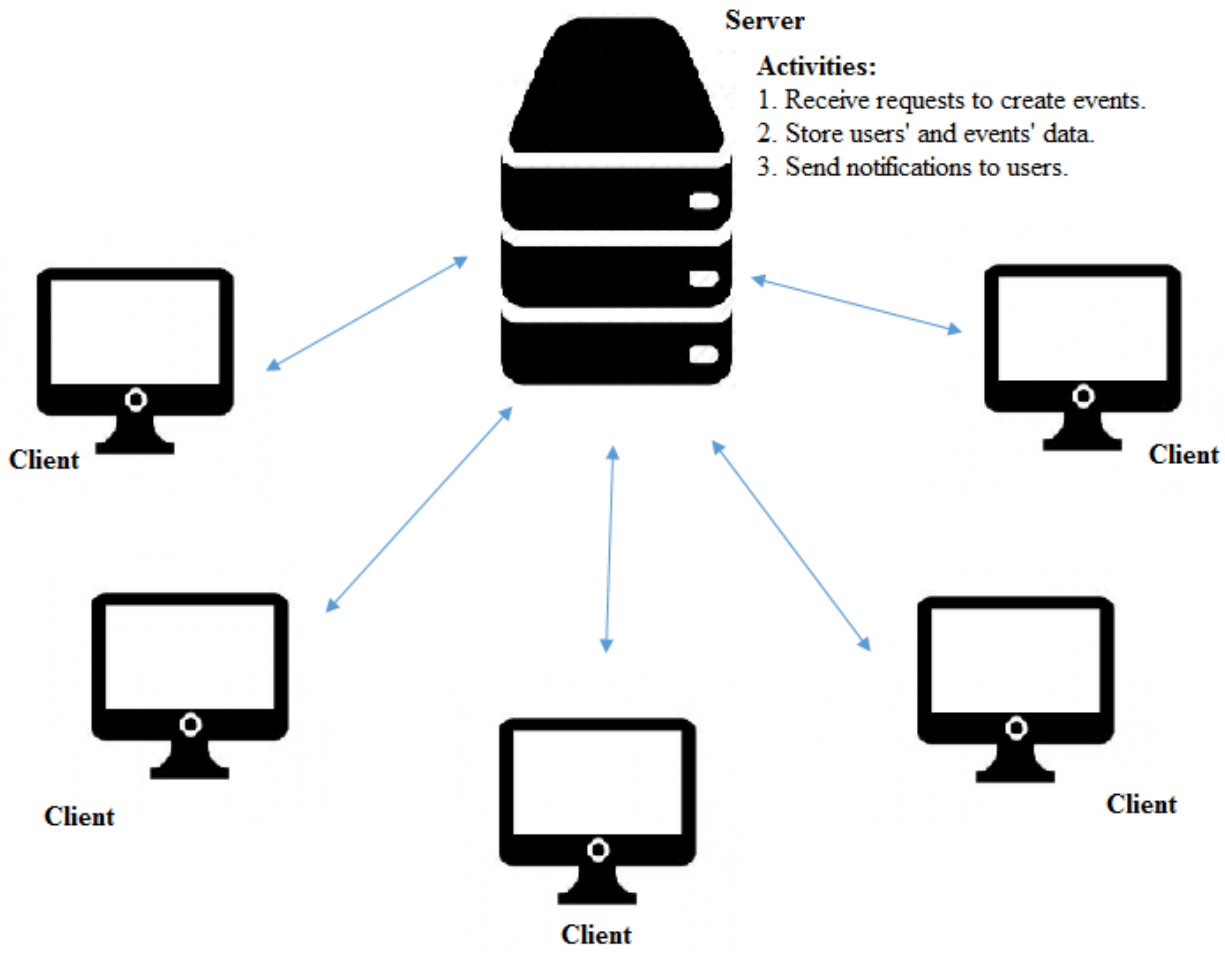

Figure 1

2.2. PRODUCT FEATURES:

The significant features provided by the application goChron are:

- Create an event providing the attributes for the event:
    - All users can create events, selecting the type of event and permissions.
- Join an event:
    - The users who are interested in joining the event may do so directly for a public event or through authorization for other events
- Search and subscribe to organizations etc., to receive notifications:
    - Users can search for clubs, organizations, etc., and subscribe to them to receive notifications about events posted by them.
- Send and receive invitations:
    - For closed and private events, the one who created the event can send out invitations and they can be received by the intended users.
- Login using email to view events upcoming and ongoing:
    - After successful login using their email id associated with UF, the user can view various upcoming and ongoing events that are visible to the user.

2.3. USER CLASSES AND CHARECTERISTICS:

goChron comes in use in universities where many organizations try to connect to students by providing all those events schedule at a glance.

The various user classes that are anticipated to use goChron are:

- Students
- Organization staff responsible for event handling.

The following is the user class diagram for goChron:

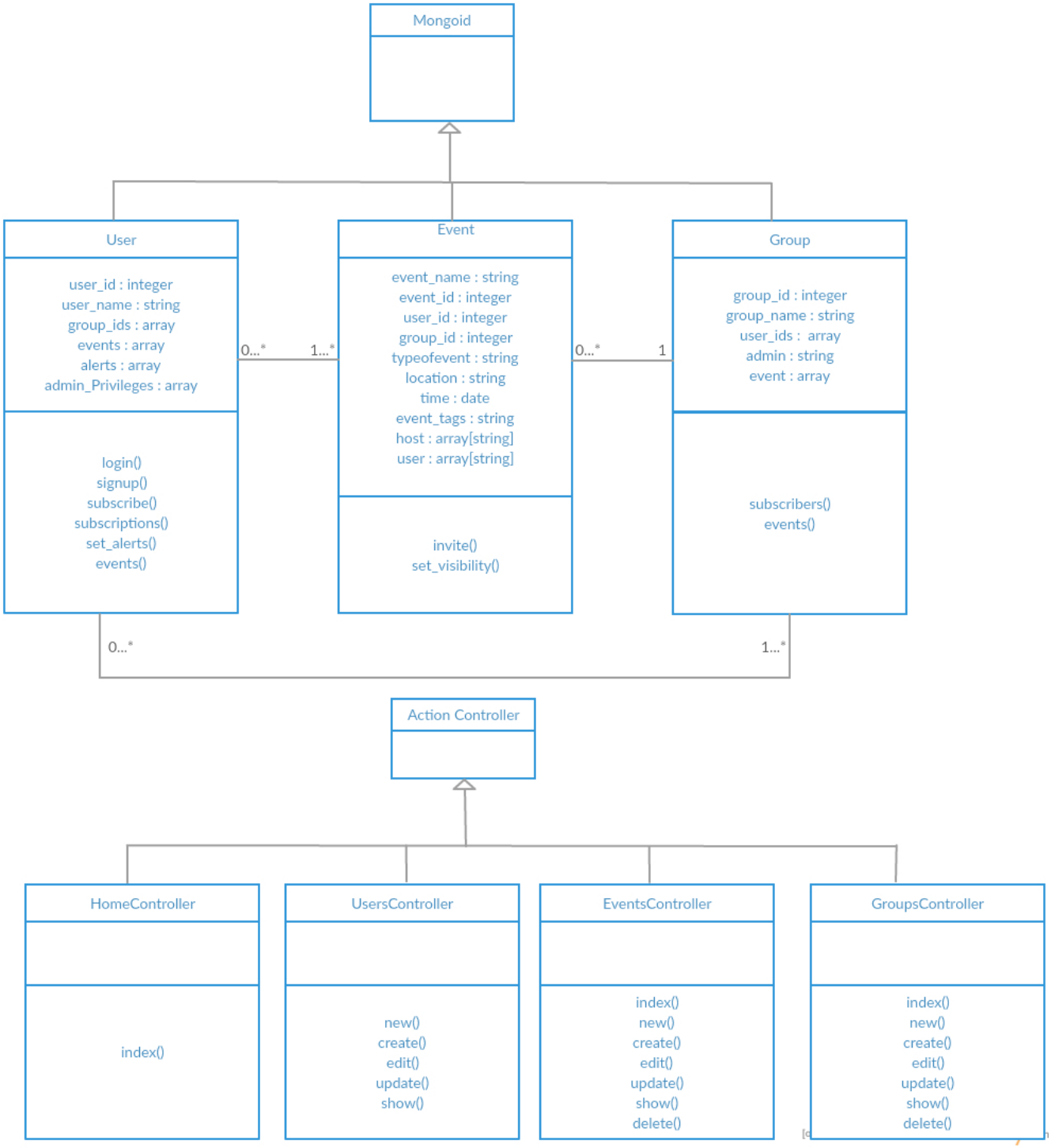

Figure 2

2.4. OPERATING ENVIRONMENT:

The GoChron project is a web application which will be accessible by any modern web browser. Although there there is no supporting native app for smartphones, it will be compatible for usage in mobile browsers.
The project will be hosted on an Amazon EC2 server, which is basically a web server that provides scalable computing capacity on the cloud. A very basic (free) version of the server is Linux t2.micro instance with 1GB RAM, 8GB storage space, 2.5 GHz processor and 750 hours of free usage per month for 1 year only. For production usage, servers will have to be scaled.

2.5. DESIGN AND IMPLEMENTATION CONSTRAINTS:

The major design constraint is to make the front end of the application compatible with any device screen. Also, since this is primarily a browser based app, smartphone users might have a UI experience only through mobile browsers unlike the native android/ios apps, but the idea of web application is to make it easy for the groups to add the events with utmost ease on a large scale.
Other constraints involve limited memory, storage space and processing power. As the traffic on the website grows, the backend architecture will have to be scaled up accordingly. For eg: Load balancers will have to be used for distributing requests to servers, database sharding and replication will have to be done for quicker access and fault tolerance, servers dedicated only for in-memory caching will be required.

2.6. USER DOCUMENTATION:

The primary goal of goChron is to facilitate to schedule and manage a subscriber's timetable. The application will be designed to be as simple to use as possible. Nonetheless, the subscribers may still require some help on how to use the goChron system. Thus, the application will contain a Help menu feature that provides supplementary information about various components of the system. It is going to be a list of topics covering each of the application's menus and features. At any time, the user can navigate to the required topic and obtain more information regarding the same.

2.7. ASSUMPTIONS AND DEPENDENCIES:

Time Dependencies: goChron is not a native mobile application. The implementation of goChron as a mobile app is totally dependent on the time spent designing and implementing the basic web application. The final decision whether to implement this feature will be made during the later stages of development.
Hardware Dependencies: None of the hardware dependencies are yet discovered.

3. SYSTEM FEATURES

goChron's system features can be categorized into two main categories namely core features and additional features. Features that are essential to the functionality of the goChron system are core features and form the body of the application. These features must be implemented in order to have a fully-functioning application. Additional features are not required for the application to function. They provide additional functionality that the customer might be interested in. The implementation of additional features depends on the time taken for the core features and can be even pushed to next version.

CORE FEATURES

3.1. USER REGISTRATION AND LOGIN:

3.1.1. DESCRIPTION:

On the home page, the user will be presented with options to login or register. All the new users can click the signup option and register to use goChron. On clicking signup button, the new user will be navigated to a registration form where the user needs to create an account with goChron using email address associated with UF. The user also enters a "display name", which will be the handle within the group. On submission of the above registration form, a link would be sent sent to the user on his registered email address for verification in order to complete the signup process. Completing this process will create and store the account details of the user in the database. All the registered users can login with their username and password.

3.1.2. STIMULUS/RESPONSE SEQUENCES:

Step 1: On clicking sign up, the user is prompted to enter uf email address and a display name.
Step 2: This information is sent to server, and server sends an email to the registered mail account.
Step 3: User will receive a confirmation link and the user has to click it in order to complete the registration process.
Step 4: All the user details are now stored in the database.
Step 5: Registration is completed and user can now go to home screen to login.

3.1.3. USER REQUIREMENTS:

VALID EMAIL ADDRESS: The user cannot proceed until a valid email address is entered. The application will verify that the user's input is consistent with the format of UF email address i.e xxxxx@ufl.edu. The authenticity of the user is also verified by sending a confirmation link to the same which the user has to verify by opening it.

3.1.4. SYSTEM REQUIREMENTS:

SECURE DATABASE SYSTEM: The application must insure that the user's information is securely stored.

3.2. ADD/RESCHEDULE AN EVENT:

3.2.1. DESCRIPTION:

When a user is logged in he will be able to add, delete or reschedule the events. On clicking a “Add new event” button, a form pops up which prompts to add event name, location and timings and also if the event is public private or closed. On clicking submit, the event will be appeared in the list of all events scheduled in the goChron Calendar depending on the type of visibility selected while creating the event. The user can also check the list of events he/she created by clicking on the button events created. The user can go to the list of events he/she created and then can reschedule or delete the events accordingly. The user can also reschedule/delete the events if he/she is made as an admin by the user who actually created it.

3.2.2. STIMULUS/RESPONSE SEQUENCES:

Step 1: The user selects "Create New Event" from the main group screen.
Step 2: The user must submit event name, timings, location, and type of event (public,private or closed) and hit “add event” button.
Step 3: After adding, the event will be successfully updated to database.
Step 4: User can see the list of events created by him and can perform different actions on it accordingly.

3.2.3. USER REQUIREMENTS:

VALID EMAIL ADDRESS: To add, delete or reschedule an event, user must log in with valid email address.

3.2.4. SYSTEM REQUIREMENTS:

Functionality includes that proper mapping of events to the users is done so that it will also appear
On their personalized goChron page. For every change in event, subscribers of that group should
Receive a notification.

3.3. JOIN/DROP AN EVENT:

3.3.1. DESCRIPTION:

When a user is logged in, he/she will be able to look up all the public and closed events listed on a particular day. He/she can click the event link and choose to join any public event. Once user joins the event, the event is now displayed in his/her calendar. If a user wants to join a closed event, he/she has to click on the link and request access from the admin of that event. The admin of the event might add or decline the request accordingly. If the user wants to join a private event, he/she has to get an invite from the event admin. When the admin of a private event sends the invite, the user will receive a notification and the user can choose to add or decline the invite. User can drop an event from his calendar any time he/she wants to by just clicking the link and choosing the option to drop.

3.3.2. STIMULUS/RESPONSE SEQUENCES:

PUBLIC EVENT:

Step 1: The user selects a public event from the list of events.
Step 2: The user chooses to join the event.
Step 3: After adding, the event will be successfully updated in the database for the user.
Step 4: User can see the list of events he is attending in a chronological order.

CLOSED EVENT:

Step 1: The user selects a closed event from the list of events.
Step 2: The user chooses to join the event.
Step 3: On requesting, the admin of the group is notified.
Step 4: Admin can accept/reject the invite.
Step 5: The action of the admin is updated in the database for the user.
Step 6: User can see the closed event in the list of events he/she is attending.
Step 7: User can see the list of events he is attending in a chronological order.

PRIVATE EVENT:

Step 1: Admin can choose to invite a user.
Step 2: The user is invited.
Step 3: The user can either accept or decline the invite.
Step 4: The user action is updated in the database for the user
Step 5: User can see the private event in the list of events he is attending.

ADDITIONAL FEATURES

3.4. GROUP CREATION AND MANAGEMENT:

3.4.1. DESCRIPTION AND PRIORITY:

Every user will be able to create and manage a group. A group is required when a series of events would happen. Generally admin of a particular organization can create a group so that he will be able to post all the events related to that organization at one place. The admin of the group will have a functionality to add/reschedule/delete an event posted in that group. The person who creates group will be responsible for adding and managing events for those groups.

3.4.2. STIMULUS/RESPONSE SEQUENCES:

Step 1: The user selects create new group option.
Step 2: The user must choose unique name to be used as the group's identifier.
Step 3: The user can add an event to the group.
Step 4: Users subscribed for group will get a notification on every event update.

3.4.3 USER REQUIREMENTS:

VALID EMAIL ADDRESS: The user cannot proceed until valid email addess is entered. The application will verify that the user's input is consistent with the format of an email address (XXXXXXX@ufl.edu)

3.4.4 SYSTEM REQUIREMENTS:

SECURE DATABASE SYSTEM: The application must insure that the user's information is securely stored.

3.5. SUBSCRIPTION:

3.5.1. DESCRIPTION:

The user can choose to get notified for certain events. The user can choose from the list of already existing UF clubs/departments/organizations and add them to his/her subscribed list. The user can also add some keywords to his/her subscribed list. Whenever an event from the subscribed group or an event with a matching key term is posted/modified/deleted by the event admin, the user will be notified of the update.

3.5.2. STIMULUS/RESPONSE SEQUENCES:

A user subscribes to a group.

- The user name is updated in the group's subscriber list in the database.
- An event is posted related to a group.
- The event is updated in the database.
- A notification will be sent to all the users in the subscribers list of that particular group.

3.6. ALERTS:

3.6.1. DESCRIPTION:

The user can choose to receive alerts to his mail reminding him of the events he/she has to attend.
User can choose the option of "remind me" which is listed when he opens a event in his calendar. On choosing the option, the user will be reminded based on his custom settings.

3.6.2. STIMULUS/RESPONSE SEQUENCES:

A user chooses for alerts for a particular event.
- The user's list in the database will be updated with the events he/she chose for alerts.
- One day before the event the alert will be triggered for the user.
- The user will receive a mail reminding him of the event.

3.7. SYNCHRONIZE WITH OTHER CALENDAR:

3.7.1. DESCRIPTION:

The user can choose to sync his calendar to his google calendar to schedule his daily tasks.

3.7.2. STIMULUS/RESPONSE SEQUENCES:
- The user can go to settings and turn on the sync option.
- The user will be asked for the username and password of his google account.
- User calendar is then synced with google calendar

4. EXTERNAL INTERFACE REQUIREMENTS

4.1. USER INTERFACES:

The logical characteristics of the interface of goChron application adheres to the properties such as clear, efficient, concise, consistent, attractive, familiar, responsive and forgiving. Taking screen layout constraints into consideration, the user interface is being designed in such a way that the view of the web application is compatible to the mobile platform as well by the extensive use of bootstrap.

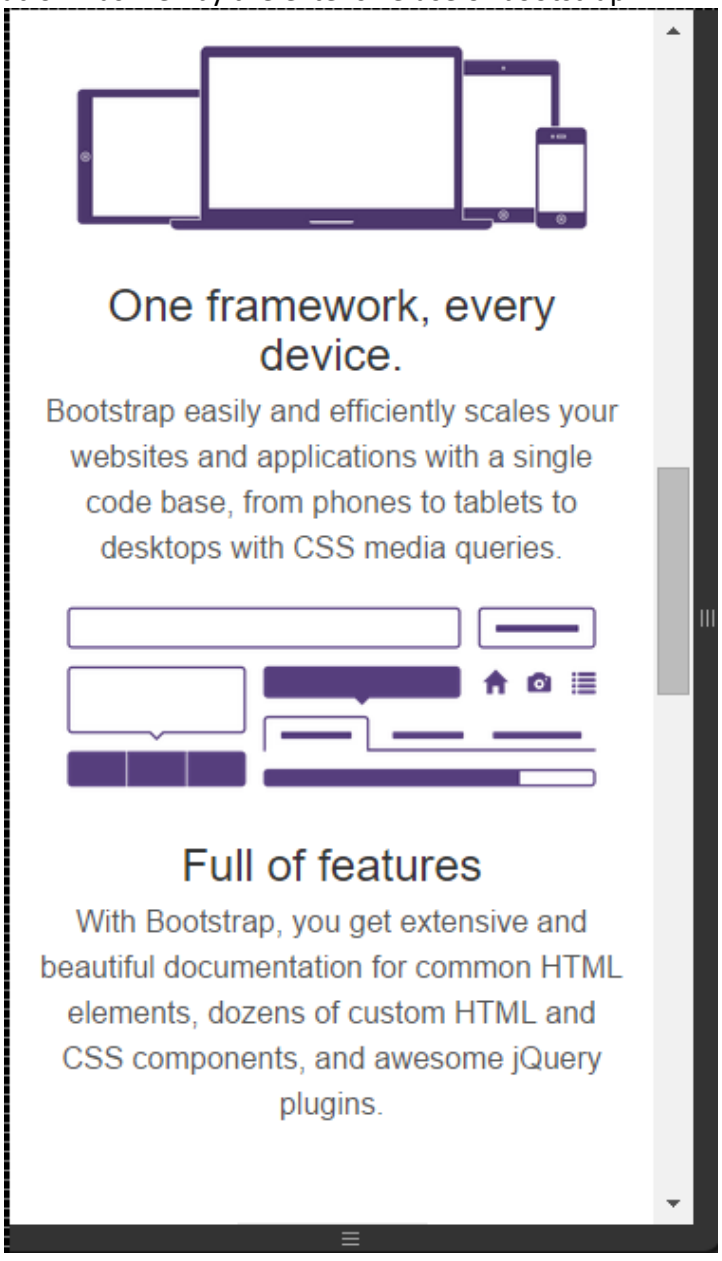

Figure 3

In addition, help button to be included to assist the users on how to use the website though most of the navigation and usage is easily self-explanatory because of the careful design constraints adopted.

There could be two types of errors, one of it is technical error which is being minimized by inheriting testing strategies, and other type is the mistake committed by the user. goChron application includes a forgiving interface to bring back the user to the previous state in case of any mistakes committed by the user. This could include the undo feature. In addition, if the user encounters a non-existent page on the website, he/she will be able to view the alternative destinations or an appropriate messages.

Figure 4

In addition, the goChron application keeps its interface clear and concise by an intuitive GUI design adding dynamic features into its content. It is easy to use the goChron application because the interface looks similar to the below where each type of event could potentially be clearly distinguished. Say for e.g., different color of the dots corresponds to different kinds of events.

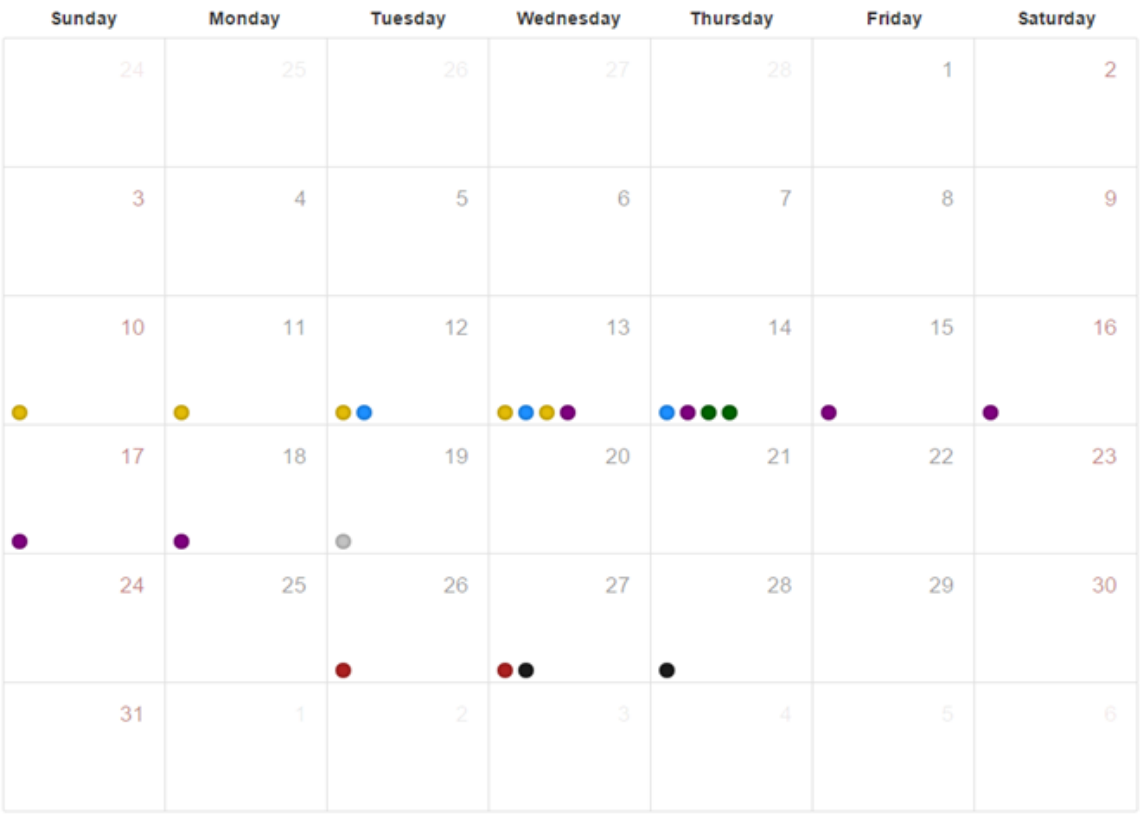

Figure 5

goChron application will have a consistent and a responsive interface across all its web pages. In addition, it would have an attractive and an efficient interface by predicting and including those functions user will be interested in to use them faster by making them easily accessible on the web page.

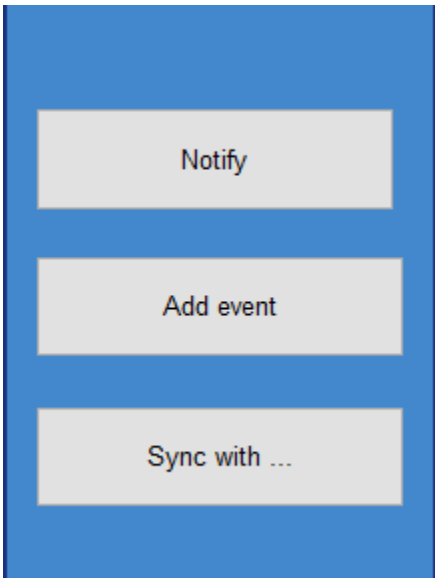

Figure 6

4.2. HARDWARE INTERFACES:

goChron is a web application that would work on Windows/Linux platform. It has the capability to work on any mobile platform as well with its web page aligning to the mobile browser. When the goChron server receives a request from the client, say ajax request, the server will send back the result and the page rendering happens at the client side thus reducing the load on the server. TCP/IP protocols, ssh and http protocols are used.

4.3. SOFTWARE INTERFACES:

4.3.1. INCOMING AND OUTGOING ITEMS:

- Outgoing data consists of emails including notifications, alerts by goChron server to the client.
- Incoming data consists of updates such as creating organizations, added events from the organizers, the details about the events the user has registered for, alerts the user has requested for and many such.

4.4. COMMUNICATION INTERFACES:

The goChron application will have a rails server. The rails server will communicate with the mongodb database and retrieve the information. Mongodb stores the complete information about user, events he/she registered for, notifications and alerts requested and complete information related to the events.

5. OTHER NONFUNCTIONAL REQUIREMENTS

5.1. PERFORMANCE REQUIREMENTS:

Performance shouldn't be an issue as goChron is consistent, efficient and also very easy to use. It is an independent application. The algorithm used is accurate as it makes sure only the subscribed list of users are notified about the posted events. goChron is very reliable as the subscriber is notified immediately as soon as an event is posted in any of his subscribed list of groups.

5.2. SAFETY REQUIREMENTS:

goChron will not affect any applications installed on the user's phone. It will not cause any damage to the device or its internal components.

5.3. SECURITY REQUIREMENTS:

This application assumes that only the subscriber / organization will have access to their calendar. Therefore, only an ufl login is required to verify the identity of the user. The identity is authenticated as it is password protected.

5.4. SOFTWARE QUALITY ATTRIBUTES:

goChron will be presented and organized such that it is visually appealing. The first priority for designing the GUI is usability. Thus, this application will be designed such that it is easy for the user to navigate. There will be notifications and alerts to provide updates to the subscribers. Also, a Help Menu option would be available to provide the user with instructions.

6. KEY MILESTONES

#	Milestone	Target Completion Date	Comments
1.	Sprint 1	10/15/2015	
2.	Sprint 2	11/25/2015	

7. KEY RESOURCE REQUIREMENTS

Major Project Activities	Skill/Expertise Required	Internal Resource	External Resource	Issues/Constraints
Ruby on Rails	Expert	Learning required		
Ruby	Expert	Learning required		
MongoDB	Expert	Learning required		
Redis	Expert	Learning required		
Elastic Search	Expert	Learning required		
jQuery	Expert	Learning required		
JavaScript	Expert	Learning required		
Html/CSS	Expert	Learning required		
Bootstrap	Expert	Learning required		

8. OTHER REQUIREMENTS

The organizations at UF should extensively use this goChron application and periodically update it with their events to keep the student community updated and make the most use of this software application.

APPENDIX A: ANALYSIS MODELS

MODEL CLASSES:

- User
- Alert (embedded document)
- Event
- Group
- Subscription

APPENDIX B: ISSUES LIST

As the goChron application is web based, it is easy for the organizations to update with the events.In addition, even the users can use it as a web based application and also in mobiles using their mobile browsers as the goChron's layout is compatible to various platforms. Depending on the time dependencies and constraints, this project may be extended to mobile apps in future.