*Florida International University*

*School of Computing and Information Sciences*

Software Engineering Focus

Final Deliverable

Project Title: Smart Stormwater / Smart City System 2.0

**Team Members:** Leandro Gonzalez, Luis Herrnsdorf

**Product Owner(s)**: Emilio Lopez

**Mentor(s)**: Emilio Lopez

**Instructor**: Masoud Sadjadi

The MIT License (MIT)

Copyright (c) *2016 Florida International University*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

***Abstract***

*This document presents the information necessary to gain a good understanding of the Smart Stormwater - Smart City System 2.0. The document introduces the objectives and motivations behind the Smart Stormwater - Smart City System 2.0, as well as how the issue was approached. All information about the user stories worked on during the realization of this project is included here, along with the UML diagrams for each one of them. Screenshots of the final system are also included in this document.*

**Table of Contents**

**Introduction** ………………………………………………………………………………………………………………………….. 7

Current System ……….…………………………………………………………………………………………………………... 7

Purpose of New System …………………………………………………..…………………………………………………... 7

**User Stories**

Implemented User Stories ……………………………………..….……………………………………………………….. 8

Pending User Stories ………………………………………………………...………………………………..….………… 21

**Project Plan**

Hardware and Software Resources ……………………………………………………...……………………….… 22

Sprints Plan ……………………………………………………………..………………………………………………………. 23

*Sprint 1*  …………………………………………………………………………………………………………………………... 23

*Sprint 2*  …………………………………………………………………………………………………………………………... 23

*Sprint 3*  …………………………………………………………………………………………………………………………... 24

*Sprint 4*  …………………………………………………………………………………………………………………………... 24

*Sprint 5*  …………………………………………………………………………………………………………………………... 25

*Sprint 6*  …………………………………………………………………………………………………………………………... 26

*Sprint 7*  …………………………………………………………………………………………………………………………... 26

**System Design**

Architectural Patterns ………………………………………………………………………………………………….. 29

System and Subsystem Decomposition …………………………………………………………………………….… 31

Deployment Diagram …………………………………………………………………………………………………….…... 32

Design Patterns ………………………………………………………………………………………………………….….... 33

**System Validation**  ………………………………………………………………………………………………………………. 34

**Glossary**  ……………………………………………………………………………………………………………………………….38

**Appendix**  ……………………………………………………………………………………………………………………………….39

Appendix A - UML Diagrams ………………………………………………………………………………………………. 39

Appendix B - User Interface Design …………………………………………………………………………….…... 64

Appendix C - Sprint Review Reports …………………………………………………………………………...…… 73

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents ……………………………………………………………………………………….….………………….....… 80

**References** ………………………………………………………………………………………………………………...………...81

# 

# Introduction

SOP Technologies is an environmental technologies startup that aims at saving our planet by helping stop ocean pollution and preventing floods while providing costs savings to communities. The Smart Stormater / Smart City System 2.0 aims to provide cities with a system to collect, analyze and share data collected by IoT devices located in the stormwater system. The system will allow cities to manage users and devices, assign devices to users, and have users receive important notifications about the fill level of the devices. The functionality provided by this system will allow cities to guarantee that no waste is going into the ocean, while providing costs savings by having users receive critical notifications through email.

## 

## Current System

The current system is owned and maintained by a third party company, which makes it very hard for software changes to be approved. In addition, the current system can only communicate with sensor devices provided by the company, thus restricting SOP to use only one type of device.

## Purpose of New System

The purpose of the New System is to provide SOP Technologies with a proprietary web application that allows cities to easily manage their users, and analyze the data collected by the IoT devices. Having a proprietary application will allow SOP to add new functionality periodically. In addition, the new system will be capable of communicating with different types of devices.

# 

# 

# User Stories

The following section provides the detailed user stories that were implemented in this iteration of the Smart Stormwater / Smart City System 2.0 project. These user stories served as the basis for the implementation of the project’s features. This section also shows the user stories that are to be considered for future development.

## Implemented User Stories

### Sprint 1

No User Stories were implemented in this Sprint. Sprint 1 was used to contact Product Owner, read project documentation, and decide on technologies to use.

### Sprint 2

**User Story #670**: Account Registration

**Description**: As an Admin I want to be able to create new user accounts (regular users or admins) so that I can give access to others.

**Acceptance Criteria:**

1. The page can be reached on the browser.
2. Information is sent to the server and it is received.
3. The registration is done by recording the information to the database.
4. An ‘OK’ response is sent back if registration goes well, or an error response if it does not.
5. New users will receive an email with their credentials and a temporary password.

**Use Case:**

* Name: Add new user
* Actor: Admin, System
* Preconditions: Admin is logged in
* Description: On the main page, the admin clicks on the ‘Users’ tab on the side panel. The admin is redirected to a page with a table listing all of its sub-users, and a plus sign on the top right corner to add new users. The admin clicks on the plus sign and a floating form appears, where the admin inputs the new user’s email, temporary password, role, and department. Admin can click on cancel button and return to sub-users page. If all required fields are provided, admin can click on save button, and the new user will be created. Admin is the redirected to users page, where he/she will now see the new user listed.

**User Story #675:** Sign In

**Description**: As a user I would like to sign in to my account so that I am able to look at the information gathered by their devices and utilize the website’s features.

**Acceptance Criteria:**

1. Be able to get to the sign in page, enter their credentials, have the system verify the credentials, and log the user in to their main page.
2. The main page should reflect the changes made by the user logging in (ex. Notifications and username appears on top bar)
3. If the credentials provided are incorrect, there should be a notice indicating that is the case.
4. Users must have a Forgot Password option that will allow them to reset their password (by email or phone) in case they forget it

**User Story #696:** Sub-Users List Page

**Description**: As an Admin, I want to be able to have a Sub-Users page where I can create a new sub-user and view all the existing sub-users, so that I can be able to modify their information.

**Acceptance Criteria:**

1. The page is accessible from other parts of the website.
2. All immediate sub-users of the user logged in will be listed
3. There will be a button that can send the user to modify a sub-user’s information.

**Use Case:**

* Name: List all sub-users
* Actor: Admin, System
* Preconditions: Admin is logged in
* Description: Admin clicks on the ‘Users’ tab on the main page. The admin is redirected to a page containing a table that lists all of his/her sub-users (users created by the admin). The table displays the sub-users first name, last name, department, role, and username. User can click on a ‘Edit’ button to modify a sub-users information

**User Story #676:** Account Settings

**Description**: As a User I want to be able to change my profile information so that I can edit my phone number, password, and name.

**Acceptance Criteria:**

1. Once logged in, the user may go to the account settings page to find their information displayed appropriately (ex. the phone number is displayed in the phone field, email is displayed in the email field, etc.).
2. After editing the information, the user clicks the “Update Profile” button and the system updates any information that is changed to the database.
3. If there are any erroneous edits, there should be a notice nearby the field indicating the error.
4. Users can click on the “Change Password” button to change their password.
5. In order to change password, users must enter their current password, and provide the new password twice.

**Use Case:**

* Name: Update Profile Info
* Actor: User, System
* Preconditions: User is logged in
* Description: User clicks on ‘Profile’ dropdown on top right corner. User clicks on ‘Account’ option from the dropdown. User is redirected to a page containing the user’s information. User can see his/her role, department, email. User can edit his/her first name, last name, and phone number, and click on ‘Update Profile’ to save the changes. User can click on ‘Change Password’ button to change his/her password. User must provide current password, and the new password (twice). The current password must be correct, and the new password must match.

**User Story #713:** Logout

**Description**: As an admin or user, I wish to be able to sign out of my account so I can make sure no one else can access my account and I no longer need to access my account.

**Acceptance Criteria:**

1. The user must first be logged in and have access to the navigation bar.
2. The user must click the person icon and then be able to click on “Logout”.
3. The user will be taken to the sign in page and typing a restricted page of the web site will not send the user to that page due to not being signed in.

### Sprint 3

**User Story #674:** Device Registration

**Description**: As a user of the website, they wish to register a new device to their account to then be able to receive information from that device to their account.

**Acceptance Criteria:**

1. Be able to reach the device registration page, enter the device’s registration information, and have the system store the information in the database.
2. Have the device appear on the main page with any other devices already registered.
3. Have the device appear to select notification settings for that particular device.

**User Story #697:** Assign Devices to Sub-user

**Description**: As an admin, I want to be able to assign a device to a sub-user so that he/she can be in charge of the device.

**Acceptance Criteria:**

1. Clicking on a user on the sub-users page will highlight the selected user.
2. A table on the bottom of the screen will display the devices assigned to the selected sub-user
3. A plus button on the top right corner of the devices table allows assigning devices to the selected sub-user
4. Clicking the plus button opens a floating window displaying the devices assigned to the logged in user
5. User can multi-select devices and click on assign, thus assigning the selected devices to the selected user, or click cancel and cancel the operation
6. Upon assigning the device, the changes will be reflected on the device table of the sub-users page

**Use Case:**

* Name: Assign device to sub-user
* Actor: Admin
* Preconditions: Admin is logged in, admin has devices assigned
* Description: On the main page, the admin clicks on the ‘Users’ tab on the side panel. On the ‘Users List’ table, the admin selects a user by clicking on it. A ‘Devices per User’ table on the bottom of the screen shows all the devices assigned to the selected user. The admin clicks on a plus button on the top right corner of the ‘Users List’ table. A floating windows comes up, showing all devices assigned to the admin. Admin multi selects devices and clicks on ‘Assign’. The floating windows closes after the selected devices are assigned to the selected user.

**User Story #698:** Modify Sub-user’s Information

**Description**: As an admin, I want to be able to edit a sub-user’s information, so that I can be able to correct any false information and/or update a sub-user’s profile.

**Acceptance Criteria:**

1. The page can be accessed through the Sub-User list page via a “Modify” button
2. A floating window will appear, displaying the selected user’s information
3. The admin can edit the sub-user’s information
4. The admin can cancel at any time, returning to the sub-users page
5. Upon clicking ‘Update Profile’, the changes will be persisted to the database, the floating window will close, and the changes will be reflected in the sub-user table

**Use Case:**

* Name: Modify sub-user
* Actor: Admin
* Preconditions: Admin is logged in
* Description: On the main page, the admin clicks on the ‘Users’ tab on the side panel. On the ‘Users List’ table, the admin selects a user by clicking on it. Admin clicks on the ‘Modify’ button of the selected user, a floating windows comes up displaying the selected user’s profile information. The admin can edit the selected user’s information, and then click on either ‘Cancel’ or ‘Update Profile’. The changes are persisted to the database, the floating window closes and the ‘Users List’ page reflects the changes

**User Story #704:** See Device Assignments

**Description**: As an admin, I want to be able to see the sub-users assigned to my devices so that I am able to keep track of the users that have access to my devices.

**Acceptance Criteria:**

1. Admins can access the ‘Devices’ page from the main side panel of the application
2. The ‘Devices’ page contains a table displaying all the devices assigned to the logged in user
3. Clicking on a device will highlight the selected device
4. A ‘Users per Device’ table on the button of the screen will display all the users that have access to the selected device

**Use Case:**

* Name: See Device Assignment
* Actor: Admin
* Preconditions: Admin is logged in, admin has devices assigned
* Description: On the main page, the admin clicks on the Devices tab on the side panel. There is a Devices List’ table, which contains all the devices the admin has assigned. The admin selects a device by clicking on it. A Users per Device table on the bottom of the screen shows all the users that have the selected device assigned to them.

**User Story #708:** Unassign Devices

**Description**: As an admin, I want to be able to unassign a device from a particular sub-user so that I can manage the devices that my sub-users can have access to.

**Acceptance Criteria**:

1. When displaying the devices assigned to the selected user in the sub-users page, admins can click on a ‘Remove’ button next to each device
2. The device assignment will be removed in the database
3. The devices table will reflect the changes, as the removed device will no longer appear for that sub-user

**Use Case:**

* Name: Unassign Device
* Actor: Admin
* Preconditions: Admin is logged in
* Description: On the main page, the admin clicks on the ‘Users’ tab on the side panel. On the ‘Users List’ table, the admin selects a user by clicking on it. A ‘Devices per User’ table on the bottom of the screen shows all the devices assigned to the selected user. The admin click on a Remove button next to a device. The device will be removed from the list of assigned devices to that particular sub-user

### Sprint 4

**User Story #731:** Delete Sub-users

**Description**: As an Admin, I want to be able to delete a sub-user’s account, so that I can revoke a user’s access to the system.

**Acceptance Criteria**:

1. The deleted user’s account will be removed from Firebase Authentication, making the credentials invalid.
2. If the deleted user is an admin, all users under said admin are assigned as sub-users to the parent of the deleted user.
3. All device assignment settings for the deleted user are deleted.

**Use Case:**

* Name: Delete sub-user
* Actor: Admin
* Preconditions: Admin is logged in, admin has sub-users
* Description: On the ‘Users’ page, the admin selects a user by clicking on it. Admin clicks on the ‘Delete’ button of the selected user, a floating window comes up asking the admin to confirm that he/she wishes to delete the selected user account. If ‘Cancel’ is clicked, the window closes and the admin returns to the ‘Users’ page. If ‘Delete Profile’ is clicked, all the information for the selected user is deleted, and the account is closed; app redirects admin to ‘Users’ page, and the deleted user is no longer listed

**User Story #777:** Multiselect Sub-users

**Description**: As an admin, I want to be able to multi-select users on the Sub-Users page, so that I am able to assign devices to multiple users at the same time

**Acceptance Criteria:**

1. An ‘Actions’ dropdown appears in the header of the Users List table of the Sub-Users page
2. The ‘Actions’ dropdown contains a ‘Multiselect’ checkbox and a button ‘Assign Devices’
3. Checking the ‘Multiselect’ checkbox adds a Selected column (with a checkbox) to the Users table, and hides the ‘Devices per User’ table
4. Clicking on ‘Assign Devices’ brings up the floating window containing the devices assigned to the logged in user. Admin multi-selects devices, and can either click on assign or cancel
5. Selected devices are assigned to all selected sub-users, and changes are saved to the database

**Use Case:**

* Name: Multiselect sub-users
* Actor: Admin
* Preconditions: Admin is logged in, admin as sub-users
* Description: On the ‘Users’ page, the admin clicks on the ‘Actions’ dropdown on the top right corner of the ‘Users List’ table. The admin checks the checkbox ‘Multiselect’, and a new ‘Selected’ column appears on the Users table (with all users unselected by default). The Admin selects multiple sub-users by checking the ‘Selected’ column. Admin clicks on the ‘Actions’ dropdown again, and then clicks on the ‘Assign Devices’ button. A floating windows comes up, showing all devices assigned to the admin. Admin multi selects devices and clicks on ‘Assign’. The floating windows closes after the selected devices are assigned to all the selected sub-users.

**User Story #714:** Devices on Map

**Description**: As an admin/User, I wish to see my devices displayed on a map so I can see where certain devices are and know where to go if I need to go to that device to clean the basin bin.

**Acceptance Criteria:**

1. There is a section that displays a map near my location.
2. The devices that have been assigned to me are marked and shown on the map.
3. Selecting a device will highlight the marker on the map.

### Sprint 5

**User Story #679:** Implement Notifications Page

**Description**: As a User, I want to be able to view all notifications received, so that I am able to check the status of the devices assigned to me

**Acceptance Criteria:**

1. Users are able to reach the notifications page by clicking on the ‘Notifications’ tab on the side panel
2. Notifications page shows all notifications for the devices the user has assigned
3. Allow users to filter by device
4. Allow users to sort notifications

**Use Case:**

* Name: See Notifications
* Actor: User
* Preconditions: User is logged in, user has device(s) assigned
* Description: Anywhere in the app, the user clicks on the “Notifications” tab on the side panel. User is redirected to the Notifications page, which contains a table displaying all notifications generated by the devices assigned to the user. User can click on the “Actions” button on the top right corner of the table, to sort the list by different fields, or to filter the list by one or more devices

**User Story #788:** Set Notifications Settings

**Description**: As a User, I would like to be able to change the settings for my notifications, so that I am able to control what types of notifications I receive

**Acceptance Criteria:**

* The Notifications Settings can be reached from anywhere in the app
* The current settings are displayed to the user
* Users can select ‘SMS’ and ‘Email’ as Notification Type
* Users can set the ‘Fill Level’
* Users can update the existing settings and save/cancel the changes

**Use Case:**

* Name: Set Notifications Settings
* Actor: User
* Preconditions: User is logged in
* Description: Anywhere on the application, the logged in user can click on the dropdown menu that appears on the top right corner. After clicking on “Settings”, the user is redirected to the “Notifications Settings” page, containing a form with the user’s current notifications settings. The form contains a checkbox for email notifications, an additional checkbox for sms notifications, and a spinner to input the fill level that will trigger the notifications. The user can modify the settings and save, or leave the page, in which case it will be notified that any unsaved changes will be lost

**User Story #672:** Email List

**Description**: The user should be able to receive notifications to the their account email whenever the email notification option for a device on one of the thresholds (green to yellow or yellow to red) is set in the notifications settings page.

**Acceptance Criteria:**

1. We are able to set the website notification option for a device on both thresholds and it is recorded.
2. The device is able to send data to the system when particular threshold is met.
3. The system creates a notification to be sent to the user’s account email

### Sprint 6

**User Story #798:** Display Devices Fill Level Graph

**Description**: As a User, I want to be able to see a graphical representation of the fill level data sent by devices assigned to me, so that I am able to interpret the data sent by the sensors

**Acceptance Criteria:**

1. The graph is located on the Main Dashboard page
2. The x-axis contains the devices assigned to the current user
3. The y-axis contains the fill level corresponding to the device

**User Story #799:** Display Devices Battery Level Graph

**Description**: As a User, I want to be able to see a graphical representation of the battery level of devices assigned to me, so that I am able to see when a device battery has drained

**Acceptance Criteria:**

1. The graph is located on the Main Dashboard page
2. X-axis contains all devices assigned to the user
3. Y-axis contains the battery level of the devices

**Use Case:**

* Name: View Battery Level Graph
* Actor: User
* Preconditions: User is logged in
* Description: Anywhere in the app, the user clicks on the ‘Dashboard’ tab on the side panel. User is redirected to the Dashboard page, which contains a bar graph that represents the battery life of all the devices assigned to the user. The x-axis contains the names of all devices, while the y-axis contains the battery level of each device

## 

## 

## Pending User Stories

* **#671:** Website Notifications
* **#673**: SMS (Text) Notifications
* **#730**: Set Sensing Interval

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Project Plan

## Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

Harward:

* Macbook Laptops for development

Software:

* MS Code IDE for writing code
* Atom IDE for writing code
* Chrome browser for testing local development
* Bash shell to run deployment firebase commands
* Google Docs for writing documentation
* Google Slides for presentation
* StarUML 2.8.0 for modeling diagrams.
* Github for version control
* Mingle for backlog and sprint management

## Sprints Plan

### Sprint 1

Sprint 1 was used to contact the product owner in order to gain insights about the project. Team members used this sprint to read the project’s previous version documentation, and to decide on the technologies to be used for the second version of the Smart Stormwater/ Smart City System. Team members and product owner decided to start the project from scratch, and to develop a scalable system by leveraging Google Cloud Platform

### Sprint 2

Attendees: Leandro Gonzalez, Luis Herrnsdorf, Emilio Lopez

Start time: 1/29 - 2:30

End time: 1/29 - 3:30

After discussion, the velocity of the team were estimated to be 60.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story 670 - Account Registration
* User Story 675 - Sign In
* User Story 676 - Account Settings
* User Story 696 - Sub-users List Page
* User Story 713 - Logout

The team members indicated their willingness to work on the following user stories.

* Leandro Gonzalez
* User Story 670 - Account Registration
* User Story 676 - Account Settings
* User Story 696 - Sub-users List Page
* Luis Herrnsdorf
* User Story 675 - Sign In
* User Story 713 - Logout

### Sprint 3

Attendees: Leandro Gonzalez, Luis Herrnsdorf, Emilio Lopez

Start time: 2/12 - 12:15 PM

End time: 2/12 - 1:00 PM

After discussion, the velocity of the team were estimated to be 64.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story 674 - Device Registration
* User Story 697 - Assign devices to sub-user
* User Story 698 - Modify sub-user’s information
* User Story 704 - See device assignments
* User Story 708 - Unassign devices

The team members indicated their willingness to work on the following user stories.

* Leandro Gonzalez
* User Story 697 - Assign devices to sub-user
* User Story 698 - Modify sub-user’s information
* User Story 704 - See device assignments
* User Story 708 - Unassign devices
* Luis Herrnsdorf
* User Story 674 - Device Registration

### 

### Sprint 4

Attendees: Leandro Gonzalez, Luis Herrnsdorf, Emilio Lopez

Start time: 2/26 - 12:15 PM

End time: 2/26 - 1:00 PM

After discussion, the velocity of the team were estimated to be 52.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story 714 - Devices on Map
* User Story 731 - Delete Sub-Users
* User Story 777 - Multiselect Sub-Users

The team members indicated their willingness to work on the following user stories.

* Leandro Gonzalez
* User Story 731 - Delete Sub-Users
* User Story 777 - Multiselect Sub-Users
* Luis Herrnsdorf
* User Story 714 - Devices on Map

### Sprint 5

Attendees: Leandro Gonzalez, Luis Herrnsdorf, Emilio Lopez

Start time: 3/19 - 12:15 PM

End time: 3/19 - 1:00 PM

After discussion, the velocity of the team were estimated to be 60.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story 672 - Email List
* User Story 673 - SMS Notifications
* User Story 679 - Implement Notifications Page
* User Story 788 - Set Notifications Settings

The team members indicated their willingness to work on the following user stories.

* Leandro Gonzalez
* User Story 679 - Implement Notifications Page
* User Story 788 - Set Notifications Settings
* Luis Herrnsdorf
* User Story 672 - Email List
* User Story 673 - SMS Notifications

### Sprint 6

Attendees: Leandro Gonzalez, Luis Herrnsdorf, Emilio Lopez

Start time: 4/2 - 12:15 PM

End time: 4/2 - 1:00 PM

After discussion, the velocity of the team were estimated to be 56.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* 798 - Display Devices Fill Level Graph
* 799 - Display Devices Battery Level Graph

The team members indicated their willingness to work on the following user stories.

* Leandro Gonzalez
* 799 - Display Devices Battery Level Graph
* Luis Herrnsdorf
* 798 - Display Devices Fill Level Graph

### Sprint 7

Attendees: Leandro Gonzalez, Luis Herrnsdorf, Emilio Lopez

Start time: 4/16 - 12:15 PM

End time: 4/16 - 1:00 PM

This Sprint will be used to finish working on:

* Poster
* Slide Presentation
* Final Documentation
* Showcase
* Videos
* Final Deliverable

# 

# 

# System Design

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

This section describes the System Architecture used in the design of the Smart Stormwater/Smart City System 2.0 platform that was based on the needs and requirements of the product owner, as well as what we have foreseen as possible future scenarios. We present an overview of the components (servers, web and database) and their functions in the system.

The proposed Systems Architecture for the platform are Model-View-Controller, Microservices and Serverless or Function as a Service (FaaS). The usage target of the platform is at a city level and thus the scalability requirements are of first order, which is why we have chosen a Serverless Architecture. For the handling of sensor devices, we have chosen IoT Core, a component of Google Cloud Platform that can accommodate a large amount of devices of any kind, making our system completely device-agnostic.

Another important aspect is the fast handling of data at the moment that occur, thus a real-time database like Firebase was chosen. The systems include various well defined components, which makes it easy to divide into their own specific managed services, making them practically maintenance free for the system administrator.

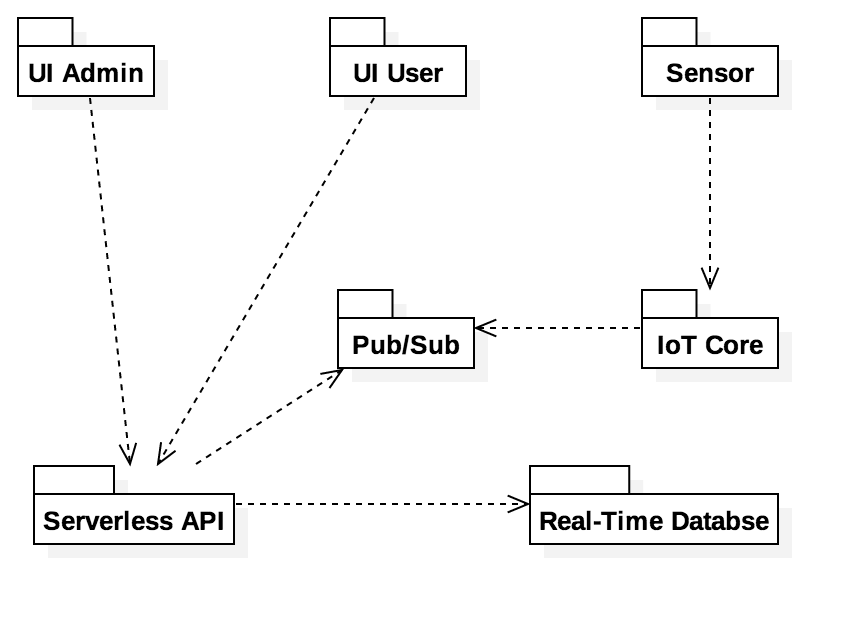


Figure 1 - Shows the platform’s package diagram divided in subsystems.

IU Admin: Presentation layer facing administrators. #698: Modify Sub-user’s Information.

IU User: Presentation layer facing user. #788: Set Notifications Settings.

Sensor: Devices that sends location data to the system. #714: Devices on Map.

IoT Core: Management of sensor devices. #798: Display Devices Fill Level Graph

Pub/Sub: Publishing message system sends data to the API. #799: Display Devices Battery Level Graph.

Serverless API: Backend logic API running services. #672: Email List.

Real-Time Database: Save system data and synchronize sensor data with front-end. #697: Assign Devices to Sub-user.

## Architectural Patterns

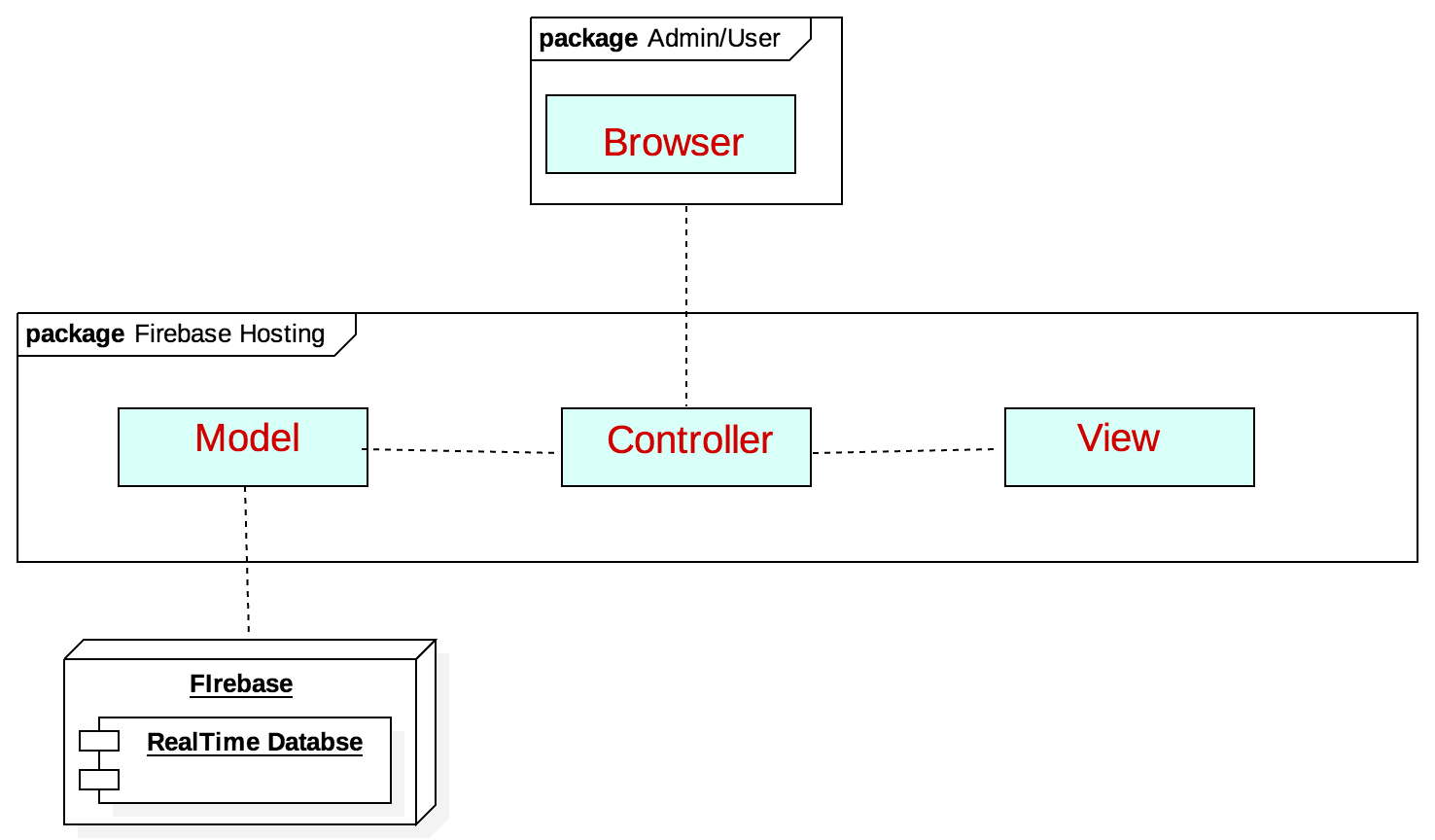


Figure 2 - Shows the MVC (Model-View-Controller) architectural pattern used in the system.

## 

## 

## 

## 

## 

## 

## Subsystem Decomposition

IU Admin: Web app for administrators that contain the “*Modify User*” button, allowing the administrators to modify user’s information. #698: Modify Sub-user’s Information.

IU User: Web app for users that contain the “*Settings*” button, that allow them so set when to receive email/sms notification, if ever. #788: Set Notifications Settings.

Sensor: External devices that sends location data and other information to the system. The system then display the sensor’s location and other data on a map. #714: Devices on Map.

IoT Core: System that list all sensors, and handles what sensors are authorized to post data into the system. #798: Display Devices Fill Level Graph

Pub/Sub: Publishing and subscribing message system that receives data from IoT Core and sends it to the API for further processing. #799: Display Devices Battery Level Graph.

Serverless API: NodeJs backend functions acting as an API and handles B2B calls. #672: Email List.

Real-Time Database: Database where all data from the system is saved, like users, devices, association of user-devices. Data from sensors are synchronized in real-time with front-end. #697: Assign Devices to Sub-user.

## 

## 

## 

## 

## Deployment Diagram

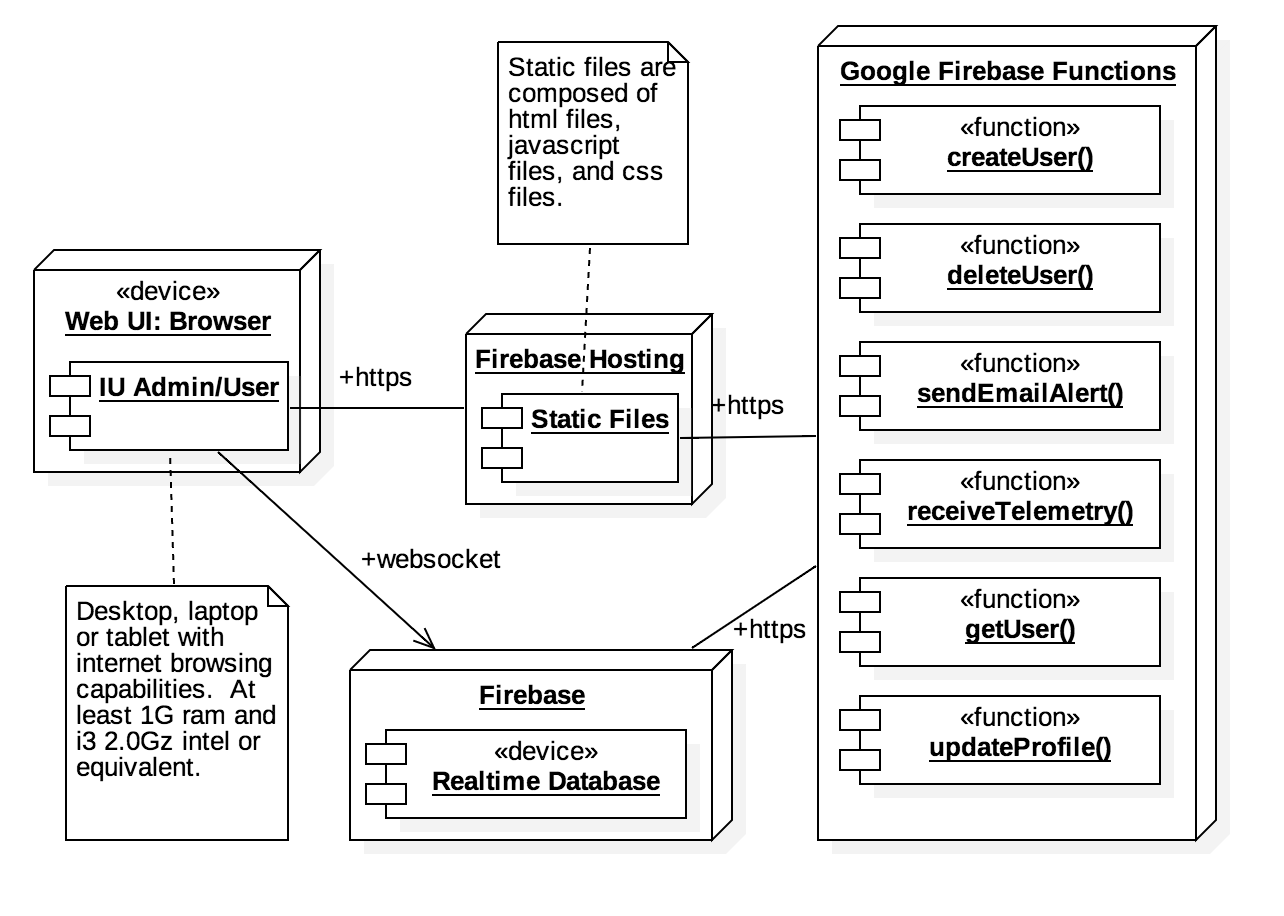


Figure 3 - Shows the details of the deployment diagram.

## 

## 

## 

## 

# System Validation

# #670 Account Registration:

* Test case ID: AccountRegistration001
* Description/Summary of Test: Admin creates a new user
* Pre-condition: Email, password, role, and department fields are provided
* Expected Results: New user is created, admin is redirected to sub-users page, new user is listed
* Actual Result: New user was created, admin was redirected to sub-users page, new user was listed
* Status (Fail/Pass): Pass

# #676 Account Settings:

* Test case ID: AccountSettings001
* Description/Summary of Test: User updates his/her account information (first name, last name, phone number), and changes his/her password
* Pre-condition: User is logged in; first and last names are provided (these two fields are mandatory)
* Expected Results: The changes will be persisted
* Actual Result: User was able to change his/her first name, last name, phone number, and password
* Status (Fail/Pass): Pass

# #696 Sub-Users List Page:

* Test case ID: SubUsers001
* Description/Summary of Test: Admin clicks on Users tab to access a list of all of his/her sub-users, from there he/she can add new users, or click a button to modify them
* Pre-condition: Admin is logged in
* Expected Results: The admin sees a list containing all of his/her sub-users
* Actual Result: The admin sees a list containing all of his/her sub-users
* Status (Fail/Pass): Pass

**#697 Assign devices to sub-user:**

* Test case ID: AssignDevice001
* Description/Summary of Test: Admin assigns devices to a sub-user
* Pre-condition: Admin is logged in, a user is selected in the ‘Users List’ table of the Users page
* Expected Results: Selected devices are assigned to the selected user
* Actual Result: The selected devices were assigned to the selected user
* Status (Fail/Pass): Pass

**#698 Modify Sub-user’s information:**

* Test case ID: ModifySubuser001
* Description/Summary of Test: Admin modifies a sub-user’s profile information
* Pre-condition: Admin is logged in, admin has sub-users
* Expected Results: Sub-user’s profile information is updated
* Actual Result: Sub-user’s profile information was updated
* Status (Fail/Pass): Pass

**#704 See Device Assignments:**

* Test case ID: SeeDeviceAssignment001
* Description/Summary of Test: Admin sees users assigned to a specific device
* Pre-condition: Admin is logged in. Admin has devices assigned. Admin has sub-users
* Expected Results: Admin can see the users that have a device assigned to them
* Actual Result: Admin sees the users that have a device assigned to them
* Status (Fail/Pass): Pass

**#708 Unassign Devices:**

* Test case ID: UnassignDevice001
* Description/Summary of Test: Admin removes device from list of devices assigned to a sub-user
* Pre-condition: Admin is logged in
* Expected Results: The removed device will be removed from the list of devices assigned to the selected sub-user
* Actual Result: The removed device no longer appears as assigned to the selected user
* Status (Fail/Pass): Pass

**#731 Delete Sub-Users:**

* Test case ID: DeleteSubUser001
* Description/Summary of Test: Admin deletes a sub-user’s account
* Pre-condition: Admin is logged in, admin has sub-users
* Expected Results: Sub-user’s account is deleted
* Actual Result: Sub-user’s account was deleted
* Status (Fail/Pass): Pass

**#777 Multiselect Sub-Users:**

* Test case ID: MultiselectSubUsers001
* Description/Summary of Test: Admin selects multiple sub-users and assigns devices to them
* Pre-condition: Admin is logged in, admin has sub-users
* Expected Results: Admin assigns devices to multiple sub-users at once
* Actual Result: Devices are assigned to all the selected users.
* Status (Fail/Pass): Pass

**#679 Implement Notifications Page:**

* Test case ID: CheckNotifications001
* Description/Summary of Test: User checks table containing his/her notifications, and sorts/filters the list
* Pre-condition: User is logged in, user has received notifications
* Expected Results: User is able to see all notifications, and to sort and filter
* Actual Result: User can check all his/her notifications and filter/sort the list
* Status (Fail/Pass): Pass

**#788 Set Notifications Settings:**

* Test case ID: ChangeSettings001
* Description/Summary of Test: User changes his/her notifications settings
* Pre-condition: User is logged in
* Expected Results: Changes are saved
* Actual Result: Changes made to the notifications settings have been updated
* Status (Fail/Pass): Pass

**#799 Display Devices Battery Level Graph:**

* Test case ID: ViewBatteryLevelGraph
* Description/Summary of Test: User navigates to dashboard page to view graph containing battery level of all his/her devices
* Pre-condition: User is logged in
* Expected Results: Battery Level Graph is accurate
* Actual Result: Battery Level Graph displays the correct battery level of all the user’s devices
* Status (Fail/Pass): Pass

# Glossary

User: Regular user of the system. Able to modify his/her profile, settings, and see device data

Admin: Super User of the system. Able to add devices, create new users, and assign devices to users

# 

# 

# 

# 

# 

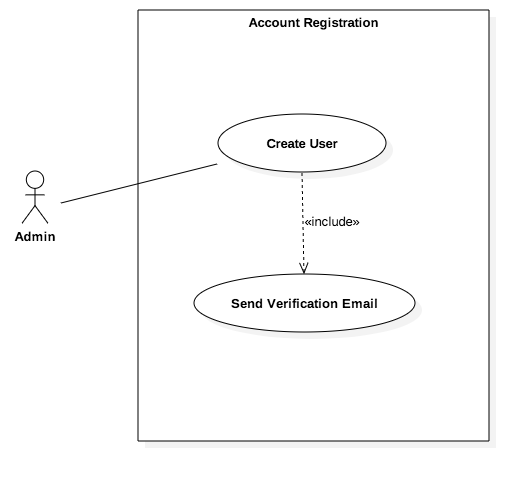
# 

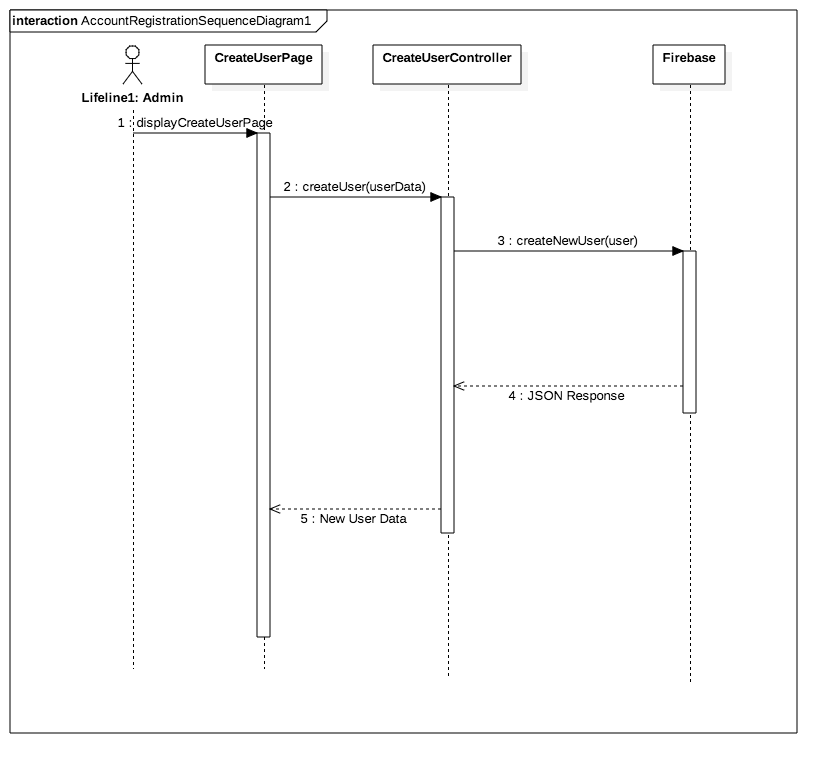
# 

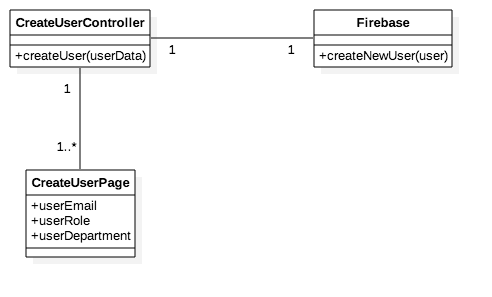
# Appendix

## Appendix A - UML Diagrams

# #670 Account Registration:







# 

# 

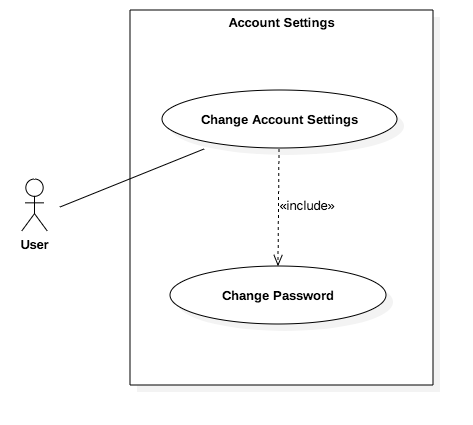
# 

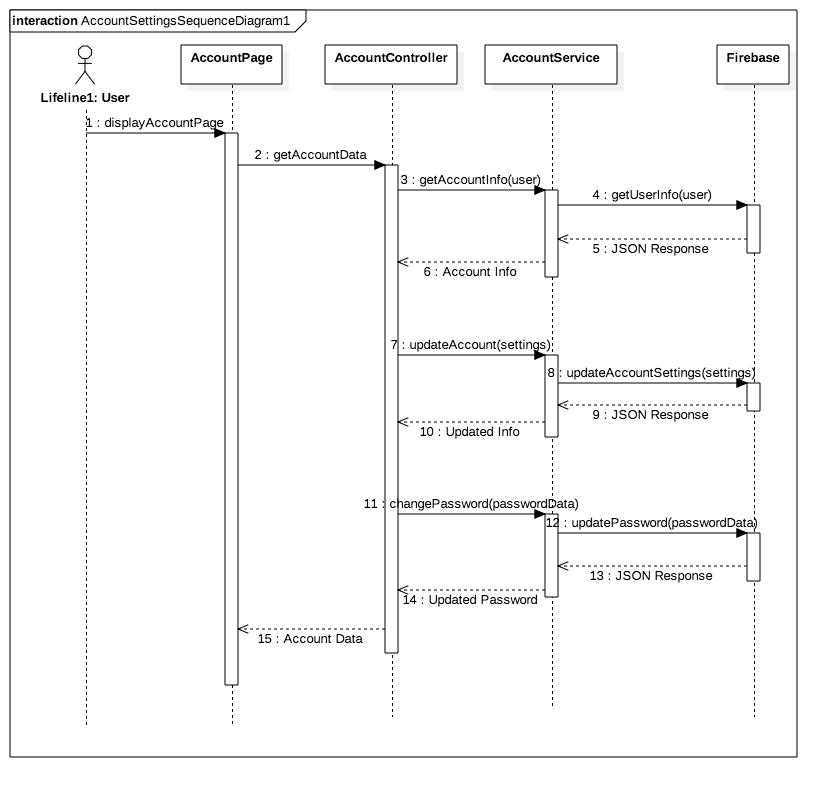
# 

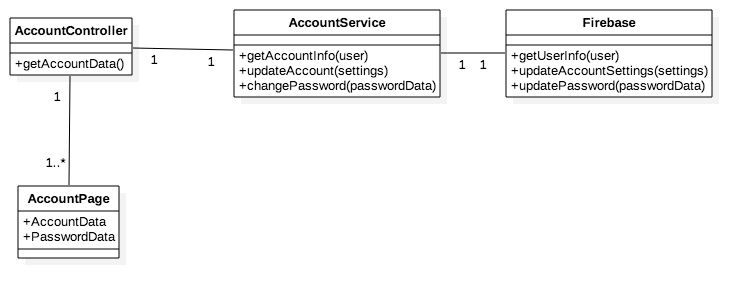
# 

# 

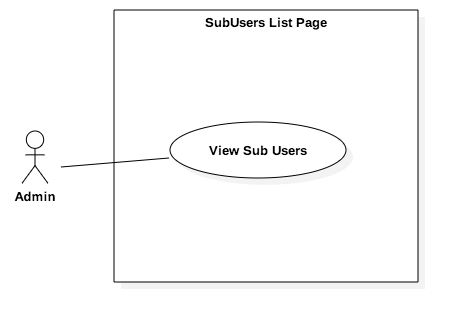
# #676 Account Settings:

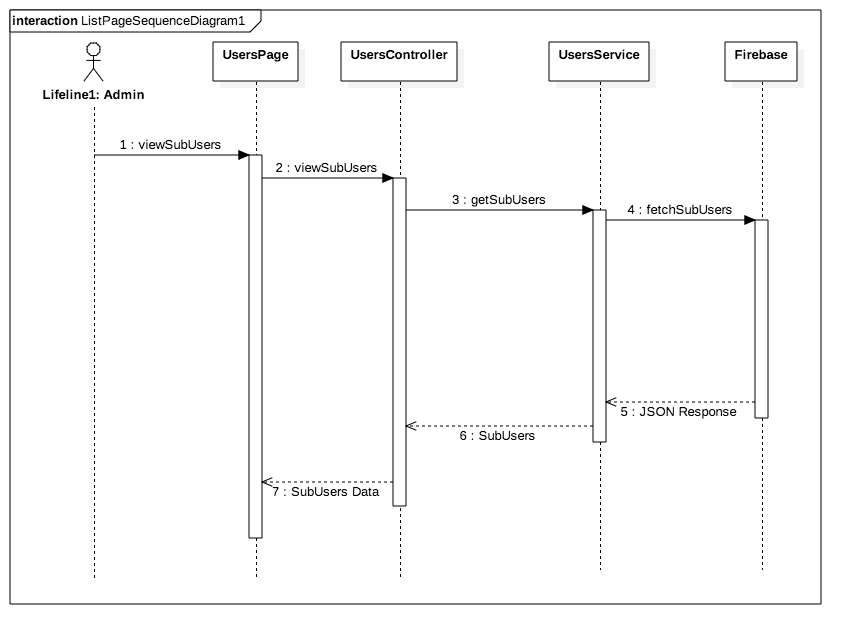


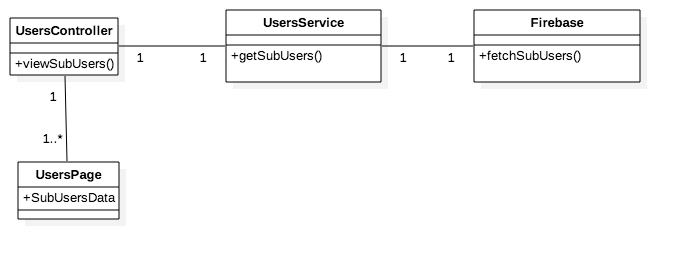




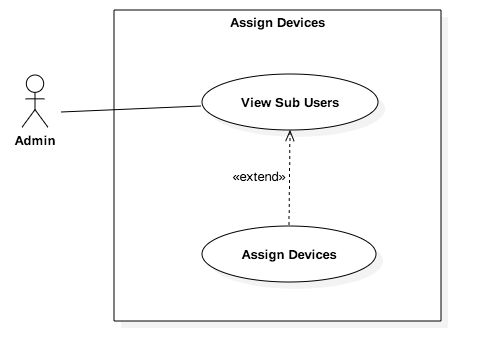
# #696 Sub-Users List Page:

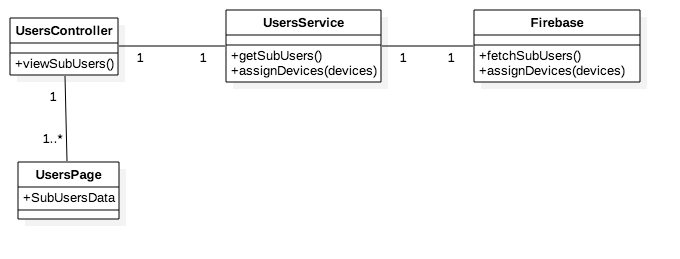
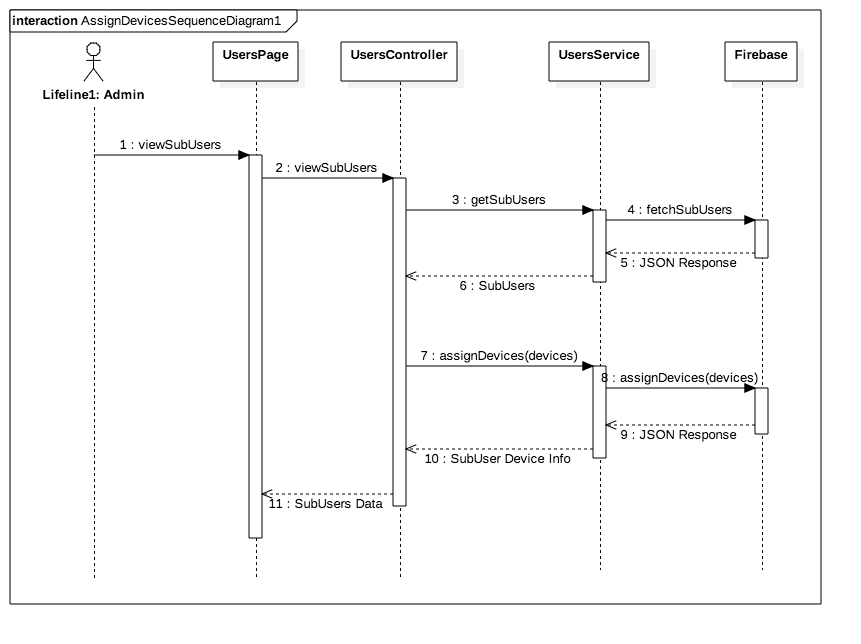




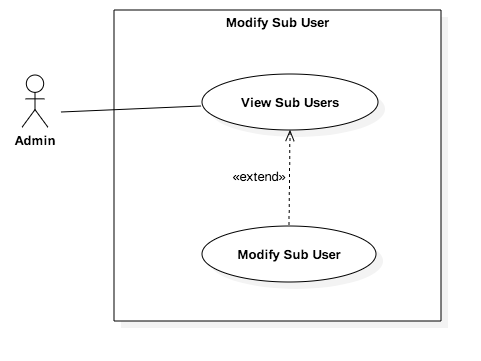


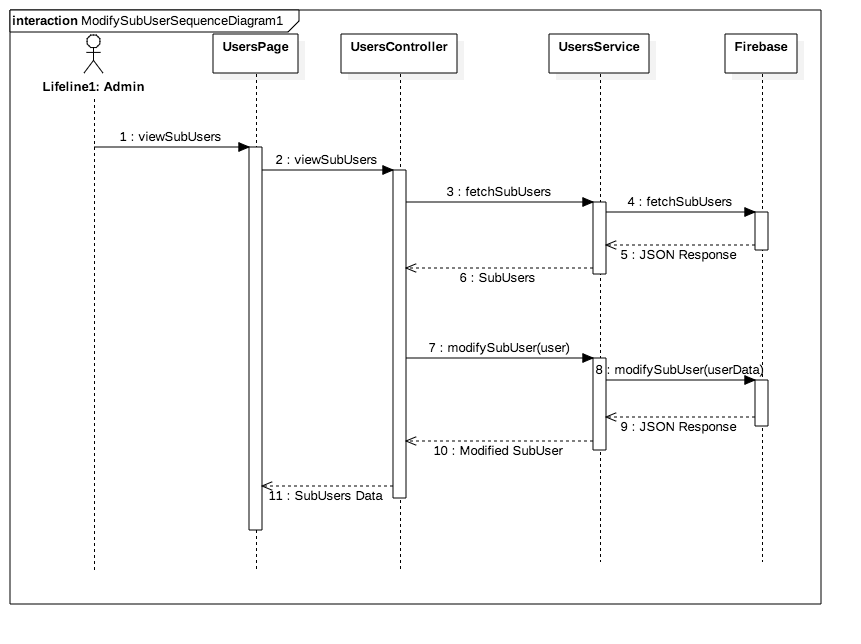
**#697 Assign devices to sub-user:**

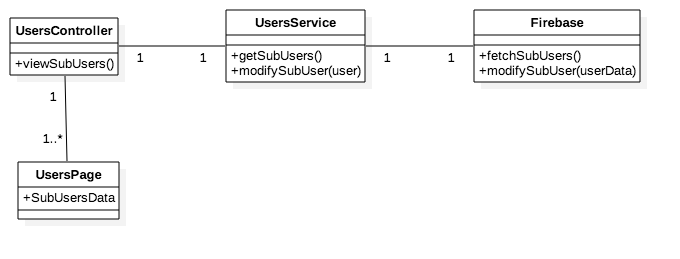




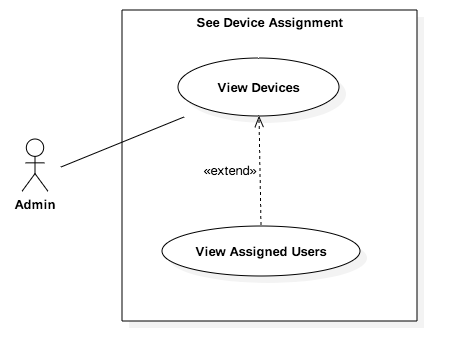
**#698 Modify Sub-user’s information:**

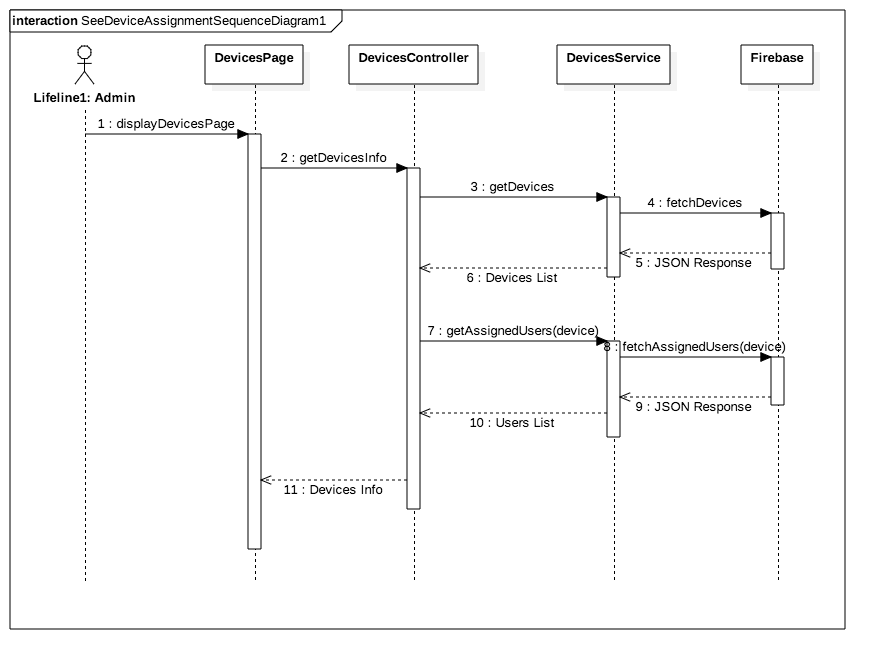


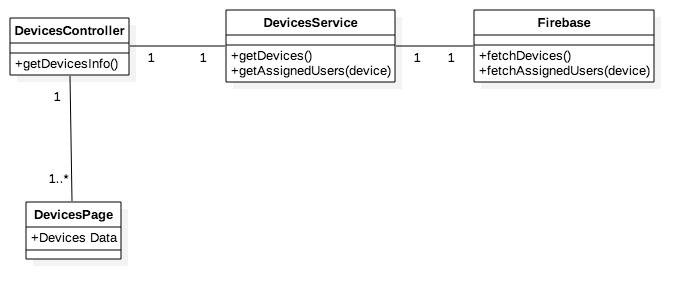




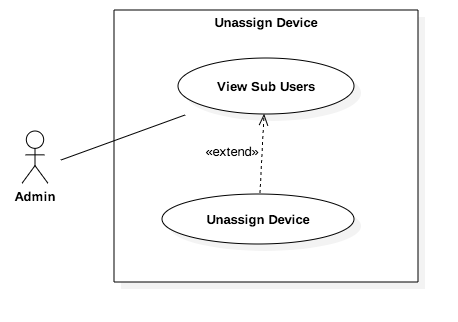
**#704 See Device Assignments:**

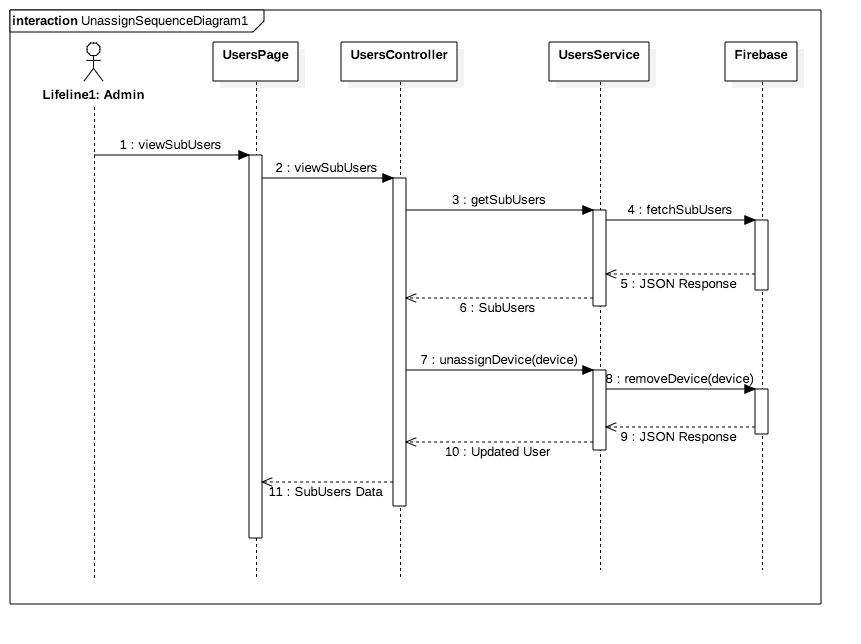
****

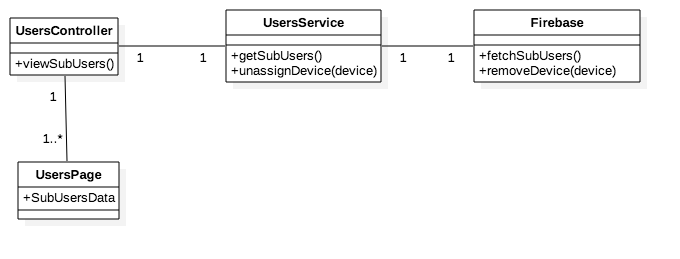
****

****

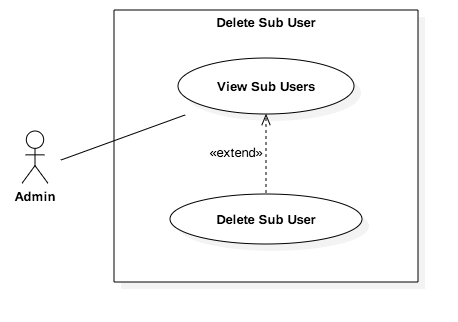
**#708 Unassign Devices:**

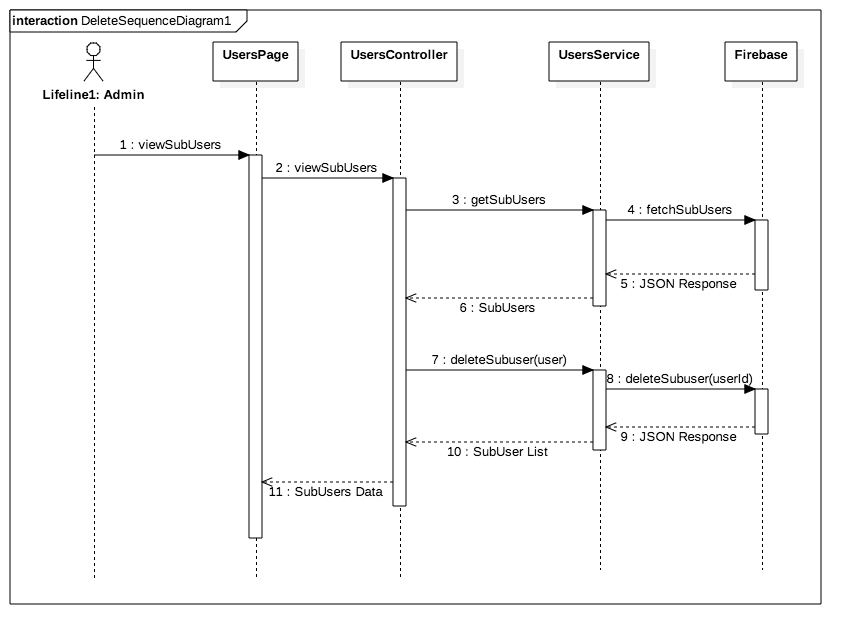
****

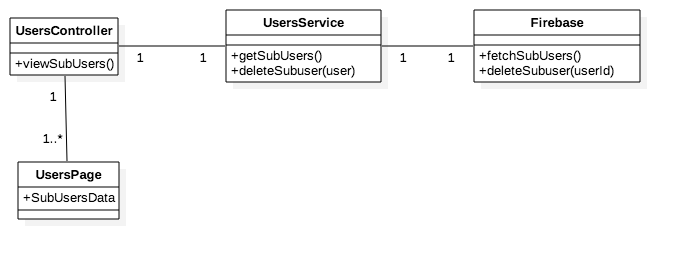




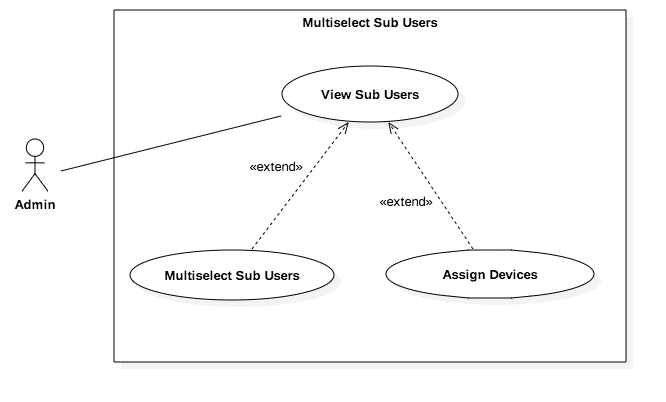
**#731 Delete Sub-Users:**

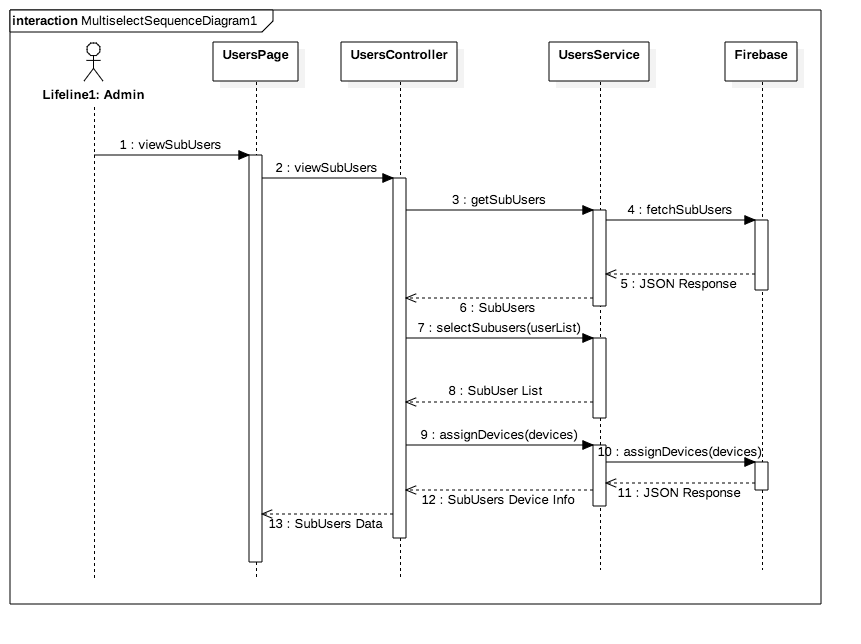


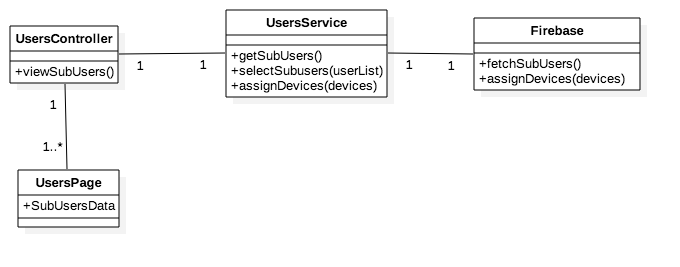




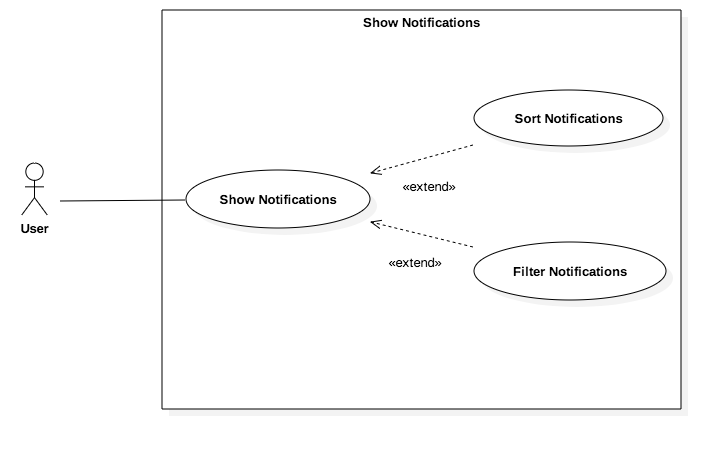
**#777 Multiselect Sub-Users:**

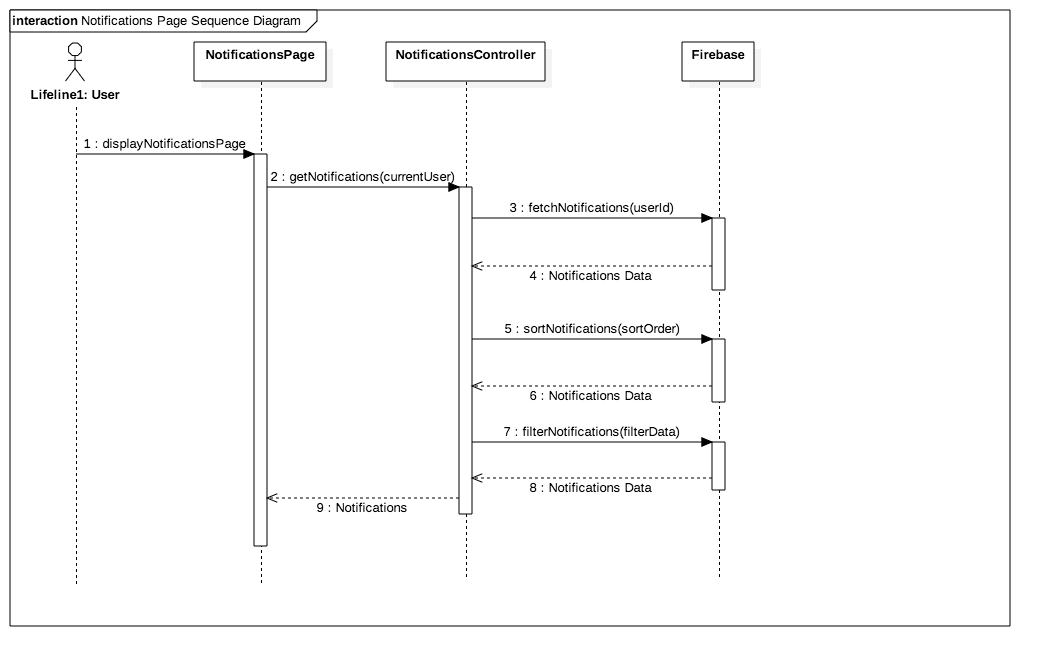
****

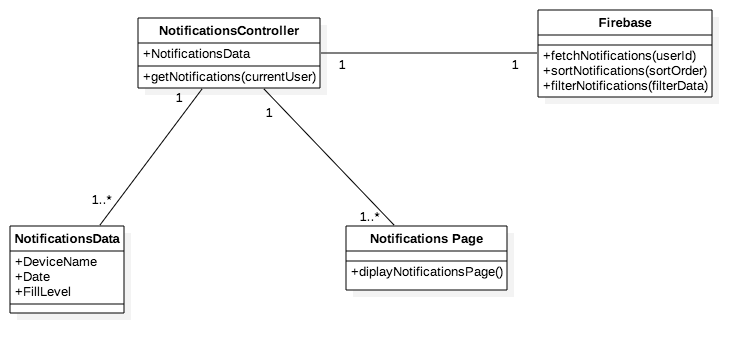
****

****

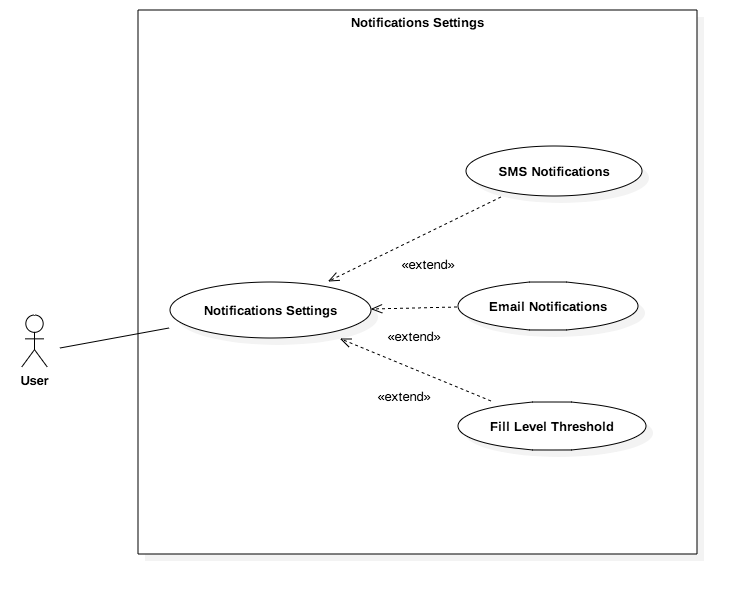
**#679 Implement Notifications Page:**

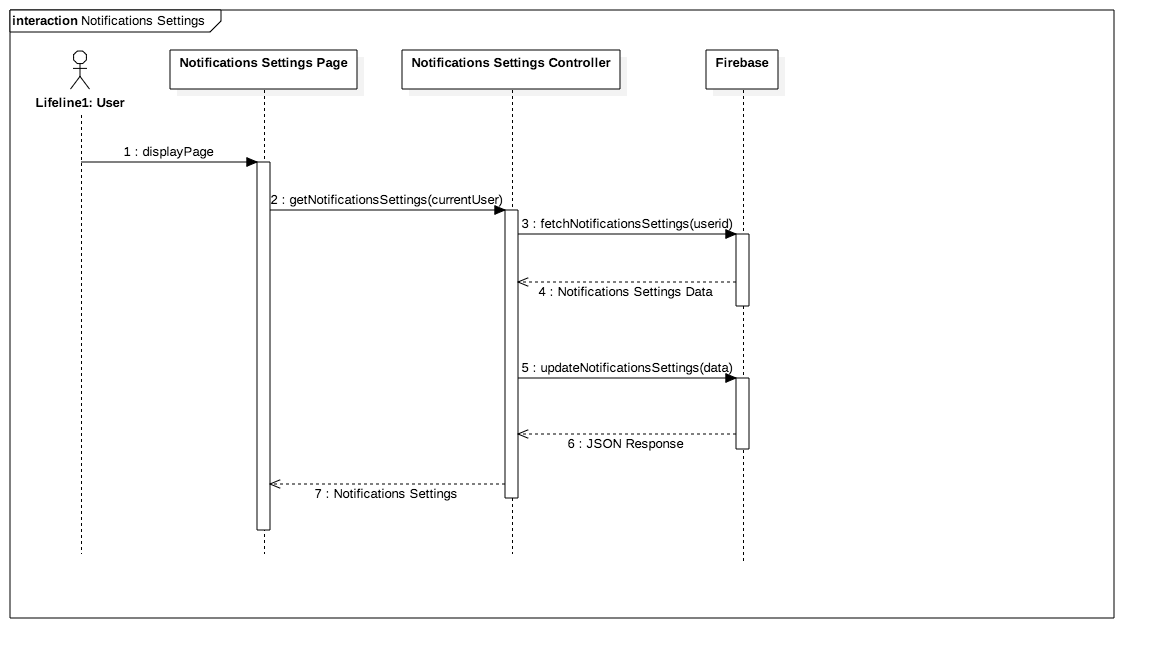
****

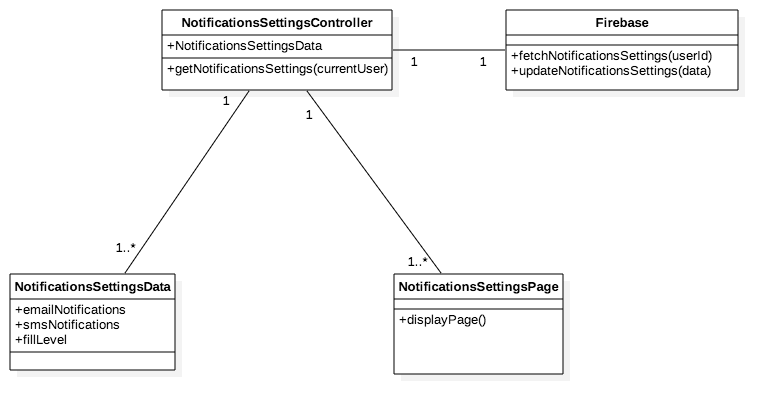
****



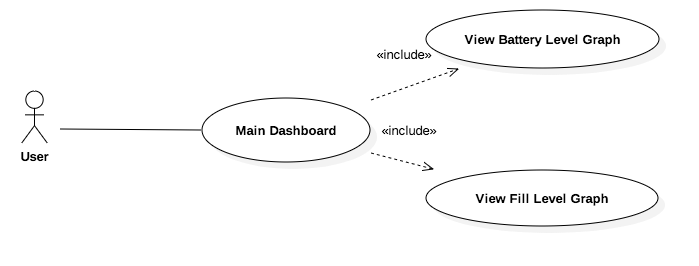
**#788 Set Notifications Settings:**

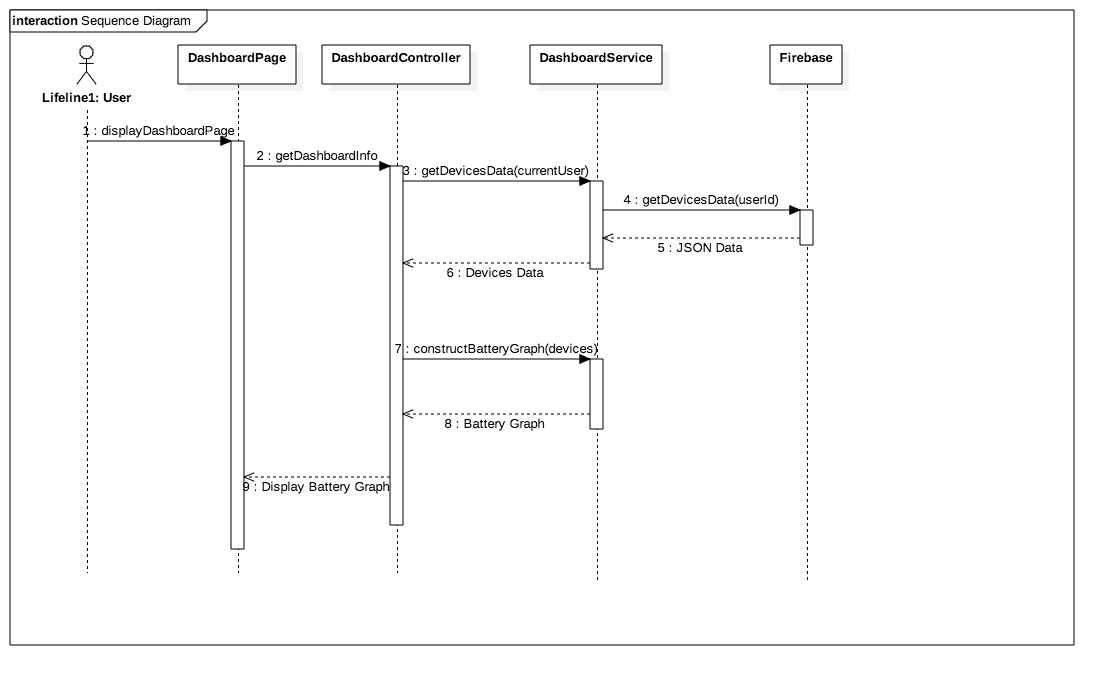


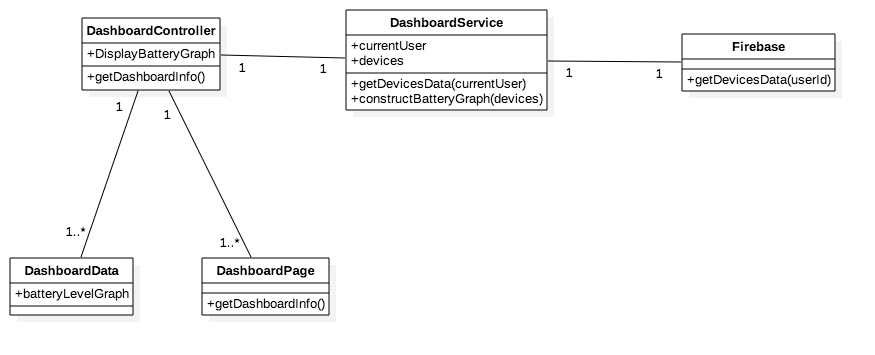




**#799 Display Devices Battery Level Graph:**







## 

## 

## 

## 

## 

## 

## 

## 

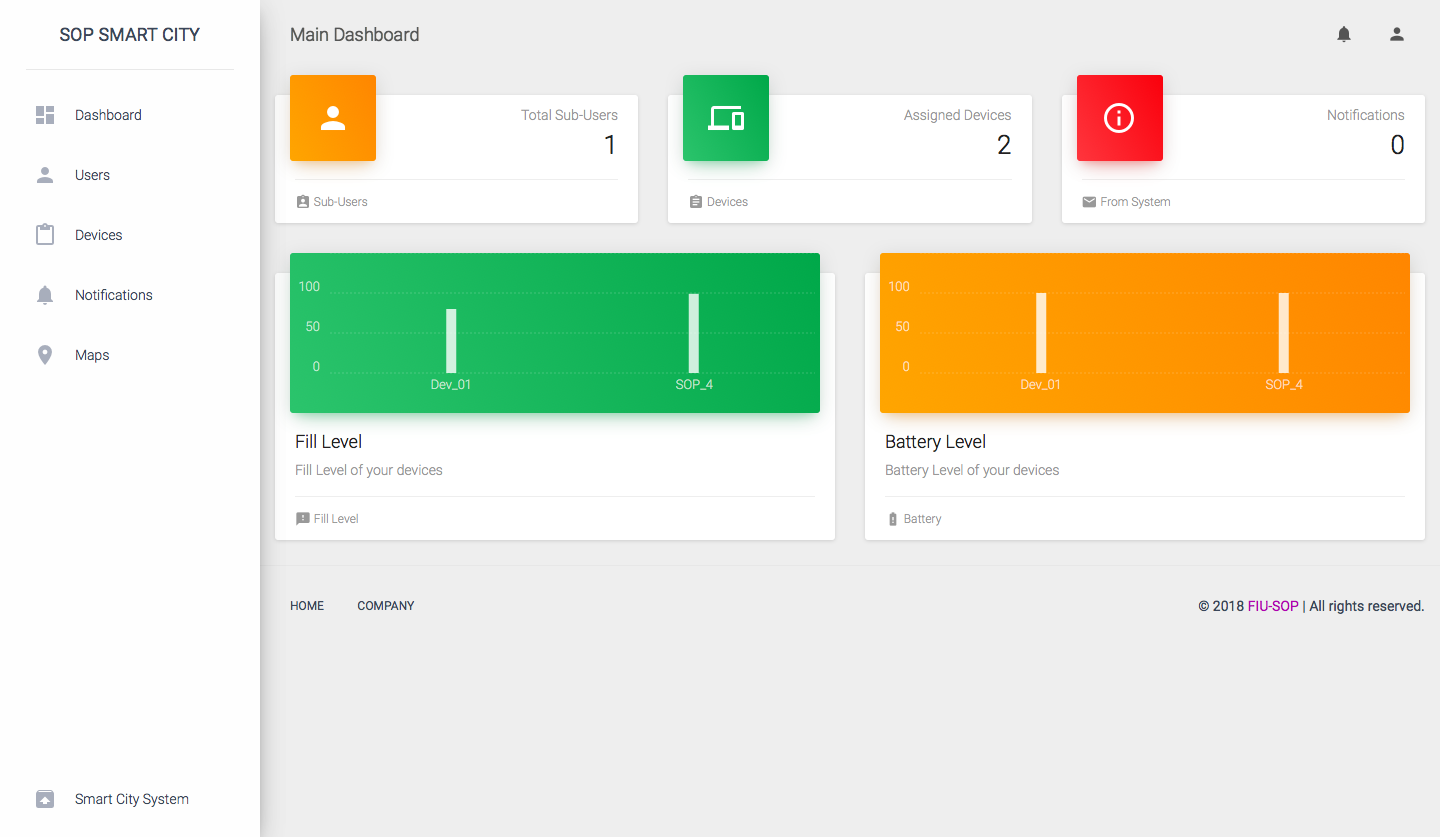
## 

## 

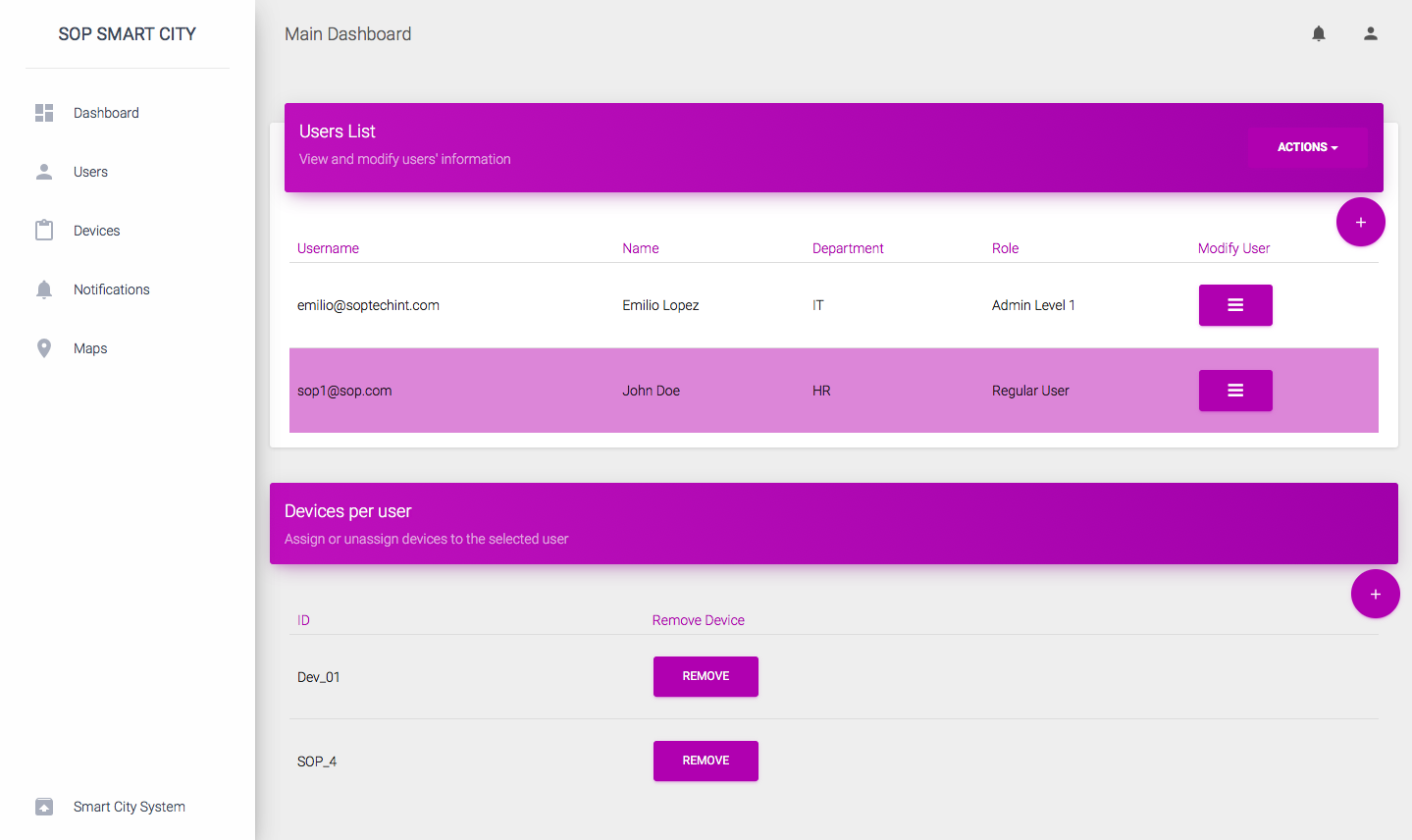
## 

## Appendix B - User Interface Design

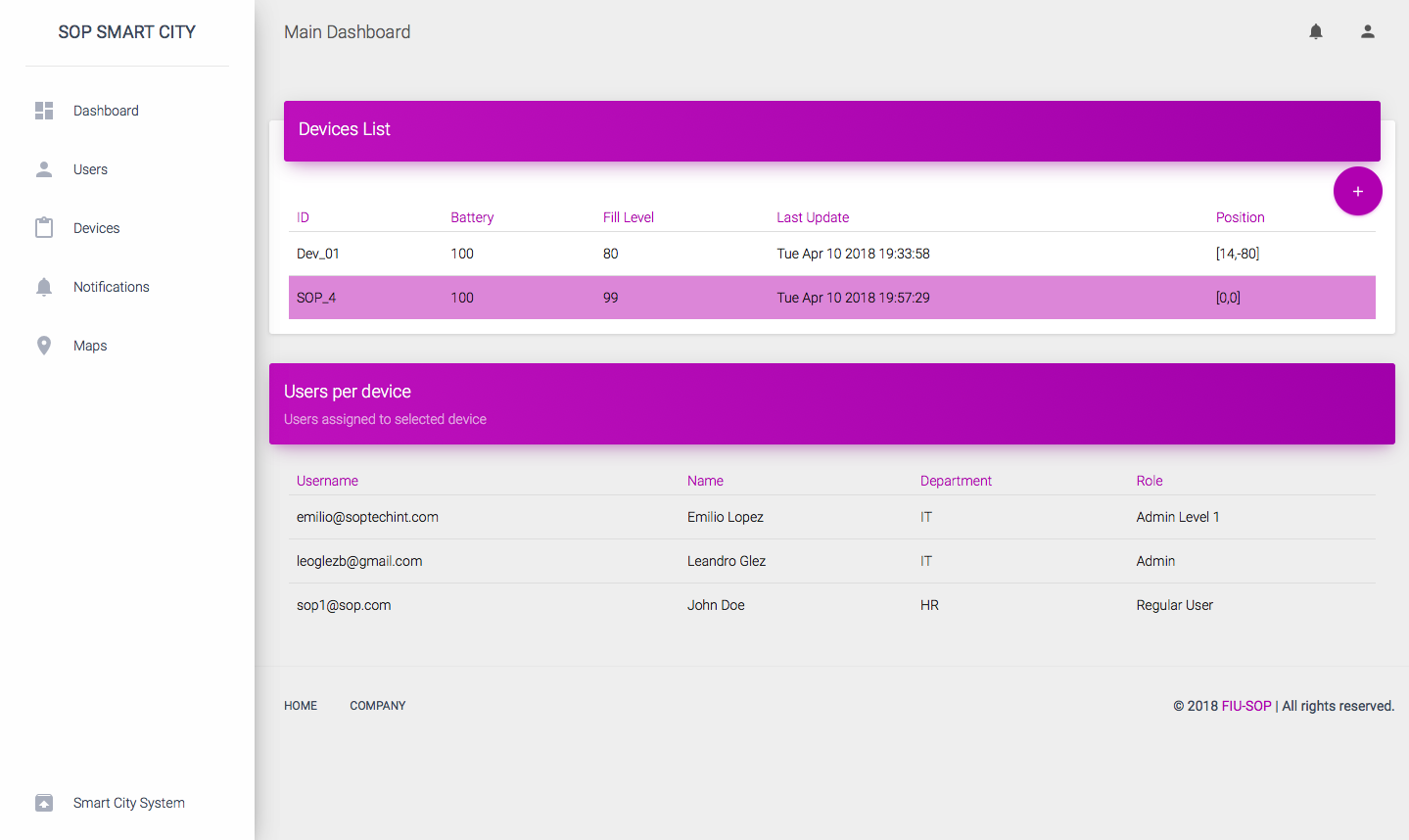
**Main Dashboard Page**



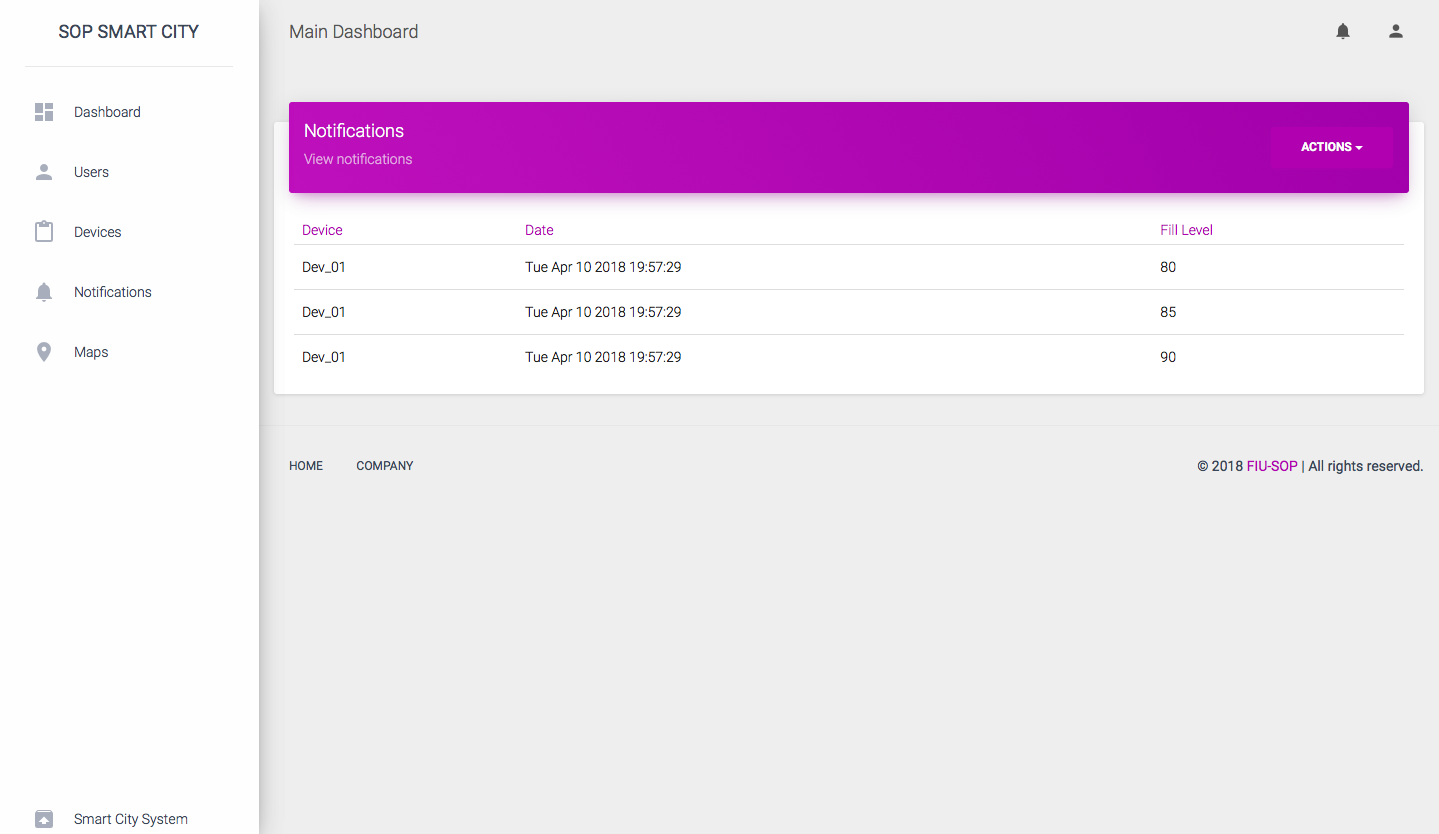
**Users Page**

****

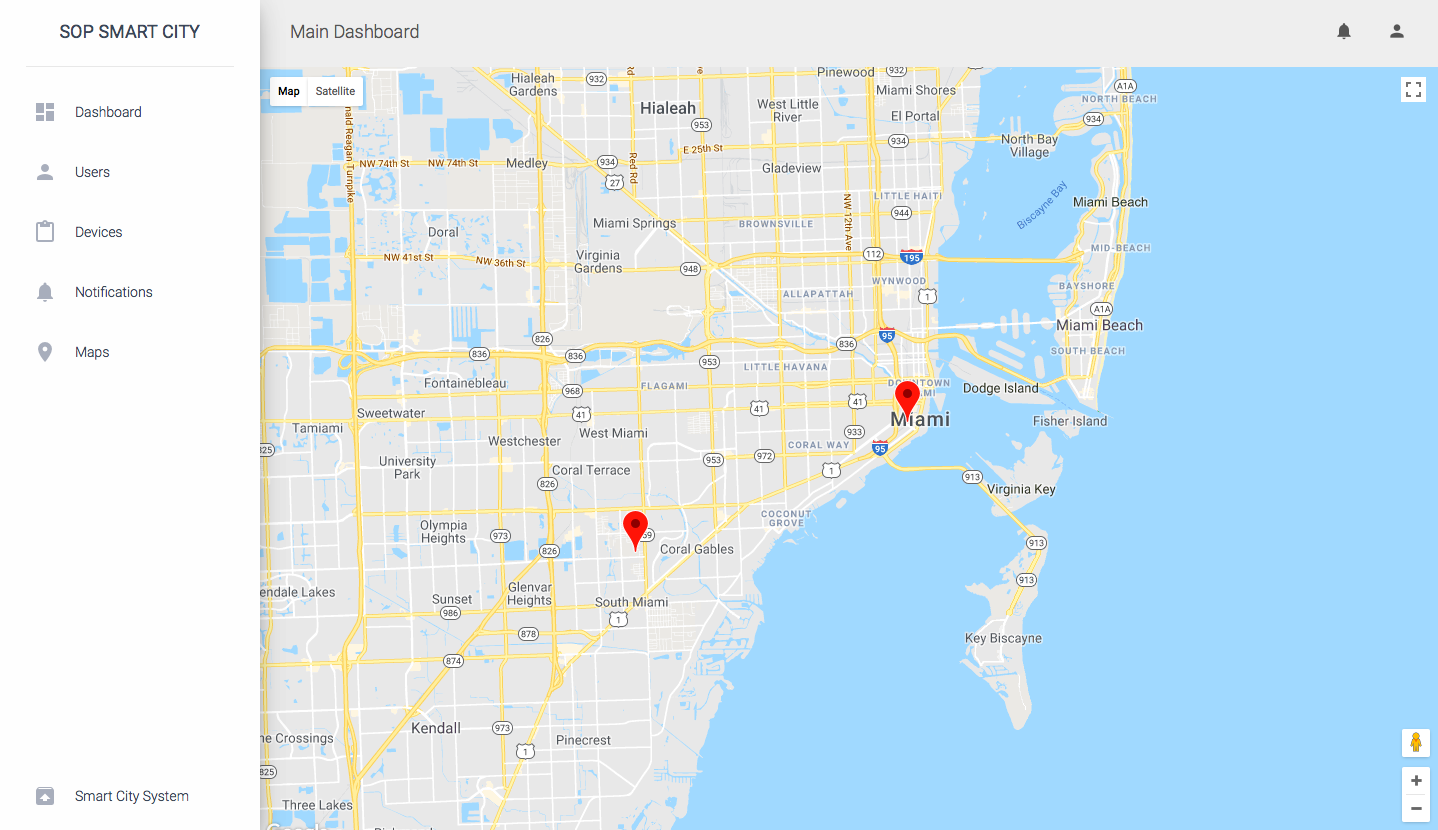
**Devices Page**



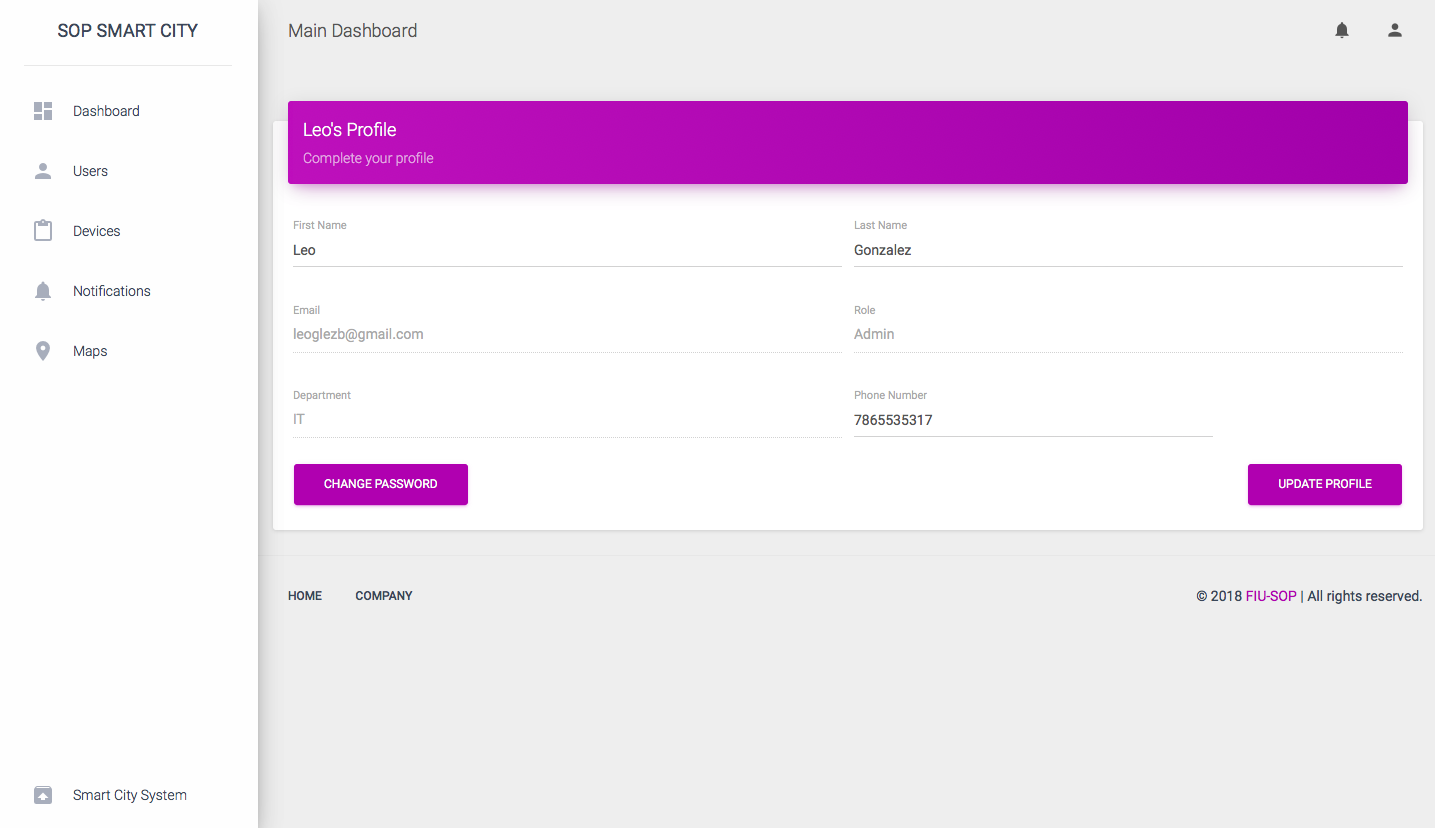
**Notifications Page**

****

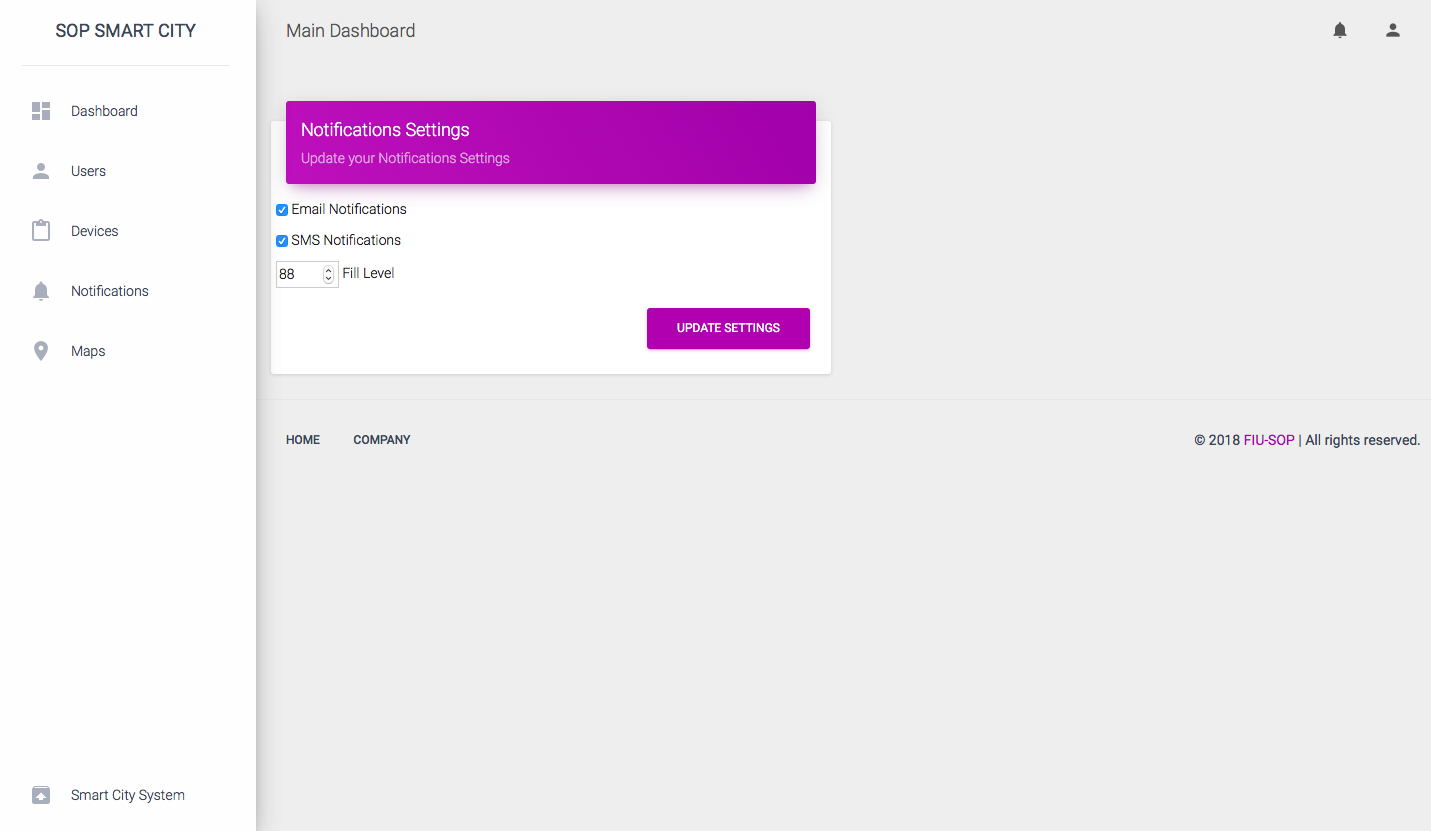
**Map Page**

****

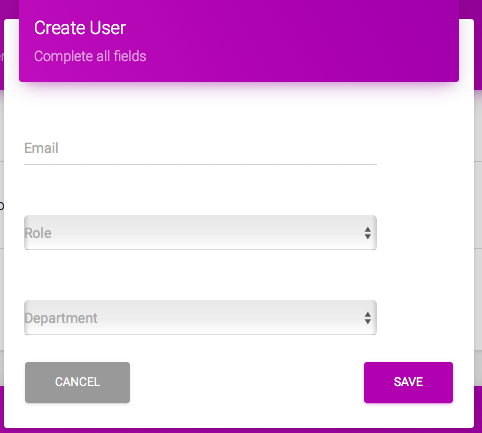
**Account Page**



**Notifications Settings Page**



**Create User Page**



**Assign Devices Page**

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## Appendix C - Sprint Review Reports

**Sprint 2 Review Meeting**

Attendees: Leandro Gonzalez, Luis Herrnsdorf, Emilio Lopez

Start time: 2/12 - 11:30 AM

End time: 2/12 - 12:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* User Story 670 - Account Registration
* User Story 675 - Sign In
* User Story 676 - Account Settings
* User Story 696 - Sub-Users List Page
* User Story 713 - Logout

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

* No user stories were rejected

**Sprint 2 Retrospective Meeting**

Attendees: Leandro Gonzalez, Luis Herrnsdorf

Start time: 2/12 - 12:00 PM

End time: 2/12 - 12:15 PM

What went wrong?

* Did we do a good job estimating our team's velocity?
  + The team’s velocity was estimated appropriately
* Did we do a good job estimating the points (time required) for each user story?
  + The points for each user story were estimated appropriately
* Did each team member work as scheduled?
  + Each team member worked as scheduled

What went right?

* All the user stories selected for the sprint were completed
* Great communication between team members

How to address the issues in the next sprint?

* How to improve the process?
  + Create UML, Sequence, and Class diagrams during implementation of each user story
* How to improve the product?
  + Work on the user stories to be selected on the Sprint Planning
  + Keep working on look and feel of the application

**Sprint 3 Review Meeting**

Attendees: Leandro Gonzalez, Luis Herrnsdorf, Emilio Lopez

Start time: 2/26 - 11:30 AM

End time: 2/26 - 12:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* User Story 674 - Device Registration
* User Story 697 - Assign devices to sub-user
* User Story 698 - Modify sub-user’s information
* User Story 704 - See device assignments
* User Story 708 - Unassign devices

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

* No user stories were rejected

**Sprint 3 Retrospective Meeting**

Attendees: Leandro Gonzalez, Luis Herrnsdorf

Start time: 2/26 - 12:00 PM

End time: 2/26 - 12:15 PM

What went wrong?

* Did we do a good job estimating our team's velocity?
  + The team’s velocity was estimated appropriately (64 hours)
* Did we do a good job estimating the points (time required) for each user story?
  + The points for each user story were estimated appropriately
* Did each team member work as scheduled?
  + Each team member worked as scheduled

What went right?

* Great communication between product owner and team members
* Team members researched about BigQuery, which will be used in future sprints
* All user stories were completed appropriately

How to address the issues in the next sprint?

* How to improve the process?
  + Diagrams must be created during the development of each story and not at the end of the sprint
* How to improve the product?
  + Research about Google IoT Core
  + Continue to test to find any bugs that may have been introduced to the application along the newly implemented features

**Sprint 4 Review Meeting**

Attendees: Leandro Gonzalez, Luis Herrnsdorf, Emilio Lopez

Start time: 3/19 - 11:30 AM

End time: 3/19 - 12:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* User Story 714 - Devices on Map
* User Story 731 - Delete Sub-Users
* User Story 777 - Multiselect Sub-Users

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

* No user stories were rejected

**Sprint 4 Retrospective Meeting**

Attendees: Leandro Gonzalez, Luis Herrnsdorf

Start time: 3/19 - 12:00 PM

End time: 3/19 - 12:15 PM

What went wrong?

* Did we do a good job estimating our team's velocity?
  + The team’s velocity was correctly estimated to be 52 hours
* Did we do a good job estimating the points (time required) for each user story?
  + The points for each user story were estimated appropriately
* Did each team member work as scheduled?
  + Each team member worked as scheduled

What went right?

* Great communication
* All user stories were completed appropriately
* Meeting with company that provides the sensors

How to address the issues in the next sprint?

* How to improve the process?
  + Create UML diagrams during the development of each story and not at the end of the sprint
* How to improve the product?
  + Research about Google Data Studio
  + Continue to test to find any bugs that may have been introduced to the application along the newly implemented features

**Sprint 5 Review Meeting**

Attendees: Leandro Gonzalez, Luis Herrnsdorf, Emilio Lopez

Start time: 4/2 - 11:30 AM

End time: 4/2 - 12:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* User Story 672 - Email List
* User Story 679 - Implement Notifications Page
* User Story 788 - Set Notifications Settings

The following ones were not completed and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

* User Story 673 - SMS Notifications

**Sprint 5 Retrospective Meeting**

Attendees: Leandro Gonzalez, Luis Herrnsdorf

Start time: 4/2 - 12:00 PM

End time: 4/2 - 12:15 PM

What went wrong?

* Did we do a good job estimating our team's velocity?
  + The team’s velocity was correctly estimated to be 52 hours; however, since SMS Notifications rely on third party services, implementation could not be completed due to costs
* Did we do a good job estimating the points (time required) for each user story?
  + The points for each user story were estimated appropriately
* Did each team member work as scheduled?
  + Each team member worked as scheduled

What went right?

* Great communication
* Email notifications implementation went as expected
* Notifications settings implementation went as expected

How to address the issues in the next sprint?

* Conduct more research about third party services that we plan on using in order to avoid not being able to integrate with our system
* How to improve the process?
  + Speak to third party providers beforehand
* How to improve the product?
  + Continue to test to find any bugs that may have been introduced to the application along the newly implemented features

**Sprint 6 Review Meeting**

Attendees: Leandro Gonzalez, Luis Herrnsdorf, Emilio Lopez

Start time: 4/16 - 11:30 AM

End time: 4/16 - 12:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* User Story 798 - Display Devices Fill Level Graph
* User Story 799 - Display Devices Battery Level Graph

The following ones were not completed and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

* No user stories were rejected

**Sprint 6 Retrospective Meeting**

Attendees: Leandro Gonzalez, Luis Herrnsdorf

Start time: 4/16 - 12:00 PM

End time: 4/16 - 12:15 PM

What went wrong?

* Did we do a good job estimating our team's velocity?
  + The team’s velocity was correctly estimated
* Did we do a good job estimating the points (time required) for each user story?
  + The points for each user story were estimated appropriately
* Did each team member work as scheduled?
  + Each team member worked as scheduled

What went right?

* Great communication
* The chartist.js library was of great help for creating the graphs
* Started working on the poster
* Started working on the final documentation

How to address the issues in the next sprint?

* How to improve the process?
  + Get ready for the Showcase
* How to improve the product?
  + Conduct one more test to guarantee everything works as expected

## Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

## User Manuals

Create a project in google cloud platform:

* <https://cloud.google.com/resource-manager/docs/creating-managing-projects>

Setup Firebase database for web usage:

* <https://firebase.google.com/docs/database/web/start>

Setup Firebase hosting:

* <https://firebase.google.com/docs/hosting/>

Setup Firebase Functions:

* <https://firebase.google.com/docs/functions/>

Get started with IoT Core:

* <https://cloud.google.com/iot/docs/>

## 

## Installation/Maintenance

* Download and install npm from npmjs.com
* Install firebase cli from your terminal or command prompt
  + $ npm install -g firebase-tools
* In the in Smart-Stormwater-Smart-City-System-2.0/functions folder, run
  + $ npm install
* In the in Smart-Stormwater-Smart-City-System-2.0 folder, run
  + $ firebase init
* Then to deploy the webapp files and the functions files, run
  + $ firebase deploy
* To deploy only web app files, run
  + $ firebase deploy --only hosting
* and to deploy only functions files, run
  + $ firebase deploy --only functions

## Shortcomings/Wishlist:

* Add SMS notifications
* Floating label in Update SubUser form overlaps with input

# References

* https://www.firebase.com/docs/web/libraries/angular/api.html