BILKENT UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

SENIOR DESIGN PROJECT

Project Analysis Report

Horus

Group Members

Ali İlteriş Tabak
Hakan Kılıç
Ömer Eren
Süleyman Can Özülkü
Yiğitcan Kaya

Supervisor

İbrahim Körpeoğlu

Innovation Expert

Nergis Koyaş

Jury Members

Özgür Ulusoy, Selim Aksoy

1.0 Introduction

The development and adoption of Internet of Things is a critical element of data driven decision making. However, most of the technology to capture and track sensor measurements are developed and installed aren't connected to the internet and can only be seen by measurement panels. Real-time tracking of the data generated by legacy systems is impossible without changing whole system. In this project, we propose an interim technology to track legacy control panels without adding big overhead by using a simple camera and image processing techniques that will bring power of the cloud and big data analytics to help our customers with the analysis of the data they accumulated from their sensor screens.

2.0 Current System

Because of the sensor and gps tracking equipment prices there have not been many startups that focused on sensors prior to 2014. However, there is a startup called **Helium** [1]. Their main use case is to provide an all included sensor pack to place wherever the client wants. However, problem with providing sensor pack is that you have to replace your existing tracking systems with new sensors, and also specific sensors wanted by client may not be provided within the given pack. However, our project ensures that clients do not need to replace their legacy systems and deployment can be made where the existing tracking system lies.

3.0 Proposed System

3.1 Overview

Measurement panels of our interest are quite diverse and used for all kinds of application from dc voltage panels to heat measurement panels used in hospital refrigerators. We are going to provide an end to end solution, from the hardware device that will track the current system to big data analytics tool to make sense of the data, that will connect current measurement system to our cloud and an analytics dashboard for our customer.

The data collection on the legacy systems will be handled by a user-end device, called the appliances that will consist of devices such as **Arduino** [2], **BeagleBone Black** [3] or **Raspberry Pi** [4], one of which will be chosen for the implementation based on the requirements of such work. These devices are very suitable for such application since they are modular, highly customizable, affordable, well-documented, and easy to work on. Since **BeagleBone Black** and **Raspberry Pi** are essentially linux machines which work on fairly strong hardware, even though their power usage may too much for operating on a battery, they can perform complex tasks that our system would require. On the other hand, **Arduino** is essentially a low-power microcontroller on a customizable board, its power usage is very suitable for operating on a battery, but its processing power may not be sufficient for our application. The final decision about the device that will be usage in our project will be made after more excessive research and experiments about

these devices, however currently the **Arduino** board seems unfeasible because of its low processing power.

The appliance contains the necessary camera module to take images of the legacy (analog) sensors, processing power to extract the data from the image using image processing techniques and a communication module to send the sensor data to the remote center (cloud) in which this data is analyzed according to the needs of the customers, and become ready to be reviewed by the customers.

The project involves work in many different domains of computer science. To extract the sensor data from the image a elegant and fault tolerant image processing system that can be executed with the limited processing power of the appliances should be developed, in order to send the data to the cloud wireless communication techniques via GSM or Wi-Fi should be used, additionally to send the data in a safe way, encryption techniques should be deployed. On the cloud side, the tools of data analytics, big-data should be used to make a meaning from the data according to the customers' requirements, and finally in order to present their sensor data and the analytics to the customers, a clear, easy-to-use and understandable user interface should be designed and implemented. Also cloud system should be robust, fault-tolerant and able to handle large throughputs, since, in ideal case, there will be lots of sensors and the data gathered from sensors is sensitive in business aspect.

Image 1: A typical legacy sensor [4]

3.2 Constraints

The main constraints of the system is technical constraints that involve the software and hardware component that will be used in the project, functional constraints that involve the functionality that the system will provide to the customers, and finally the economical constraints that involve the marketability, sustainability and the customers of the project.

3.2.1 Technical & Functional Constraints

The technical and functional constraints will be separated on two parts based on the main two parts of the project, the appliances that will be attached to the sensors to extract the sensor data and the remote center that the sensor data will be collected, analyzed and presented to the customers with an interface.

3.2.1.1 Appliances (Devices)

As the appliances that collect the sensor data and send to the remote center need to work on remote, hard to reach areas:

- The appliances should operate on battery power for at least two weeks in order to be a real automation over the current practises.
- The appliances should be durable against weather conditions and physical conditions as they might need to work under tough circumstances.
- The appliances should work persistently as they might be the only medium of accessing the data of sensors that they read, and they should be safe against failure.
- The appliances should be aware of their failure conditions (low battery, hardware/software failure etc.) and must alert the center under such circumstances in order to ensure that the needed maintenance could be performed on them.
- The appliances should support remote firmware / configuration updates, as they might not be available for a near (through direct access) updates.

- The appliances should be able to operate on minimum resources in order to prolong the battery life.
- The algorithms that will be executed on the appliances to extract the sensor data from visual data should be as efficient and accurate as possible, in order to ensure persistency, correctness and also to save the battery life.

3.2.1.2 Remote Center (Cloud)

As the remote center will manage the appliances, and also be the main interface of the system to the customers:

- Remote center should be able to handle the volume of data that is transmitted from appliances.
- Remote center should provide a clear and understandable interface to the customers, as it is the main medium of access to the sensors that the customer have.
- Remote center should provide necessary data analysis tools to the customers (plotting tools, statistical tools etc.) in order to enable the customers to keep track of their sensor data in a useful and meaningful way.
- Remote center should be in contact with the appliances all the time with a handshake mechanism to ensure the failure safety of the system.
- Remote center should be able to send data to the appliances such as firmware, configuration updates.

3.2.2 Economical Constraints

- As the customers have to rent / buy the appliances of the system to attach their sensors, the appliance hardware must be as affordable as possible.

- The appliances should be modular in order to provide only as much as the customer needs. For example if the battery packs are not needed because of the availability of the electricity, then the appliances will need to be adjusted accordingly.

- In order to be preferred by the customers over their traditional, existing methods to handle the goal (to collect sensor data), the cost of the overall implementation should be less, or it should be justify the additional costs by providing extra utilities and functionality.

3.3 Professional & Ethical Issues

One of the biggest ethical issue in this project is ensuring the data privacy, persistence, correctness and safety for the costumers since this system will be their main medium to access their sensor data which may be a sensitive and confidential data, such as a data from a nuclear plant or a military facility.

Thus in order to handle these ethical issues about data privacy, we will employ encryption on both the appliances and the remote center, also we will ask for guidance from one of our faculty members that can help us about data privacy issues. Also in

order to ensure the data correctness, we will deploy error detection and correction utilities again both in appliances and the remote center, which will help us and our customers to collect the accurate data, persistently.

3.4 Requirements

3.4.1 Functional Requirements

3.4.1.1 Dashboard

- Users will be able to add appliances (end-point clients) to their monitoring system.
- Users will be able to see whether specific appliances are working or not.
- Dashboard will provide statistical data to users.
- Users will be able to specify threshold values for specific sensors.
- Dashboard will provide an alarm in case of a sensor's threshold value is reached.

3.4.1.2 Appliances (Devices)

- Appliances will be able to inform remote center about their status (in terms of battery power status etc.).
- Appliances will be able to get image data from analog sensors and transfer it to the remote center for processing.
- Appliances will be able to collect data from digital sensors deployed in the legacy system.
- Appliances will be able to report data with adjustable frequencies chosen by customers.
- Appliances will be remotely instructed to transfer data from specific sensors.

3.4.2 Non-functional Requirements

- Remote center should contain data securely.
- Remote center should be able to handle requests from appliances within 5 seconds.
- Appliances should be able to run on battery power for at least two weeks.
- Appliances should be able to hold data as long as the transfer to remote center is not finished.
- Remote center should keep redundant copies of its database.
- Remote center should be fault tolerant, the system as a whole should not shutdown in case of specific module failures.

3.4.3 Pseudo Requirements

- Mobile application of Horus will be implemented using Java for Android OS.
- Back-end of Horus will be implemented using Golang.
- Front-end of Horus will be implemented using CSS, Javascript, AngularJS, NodeJS.
- Hardware of device will be BeagleBone Black. HD Camera Cape for BeagleBone Black will be attached to BeagleBone Black. Image Processing will be applied using OpenCV library for Python.
- Backend and the devices will communicate using HTTPS secure channel.

3.5 System Models

3.5.1 Scenarios

Actors: User, Client Admin, Restricted Client

3.5.1.1 Register to the System

Scenario Name: Register to the System

Participating Actor Instances: EDA$ Official:User

Assumptions: User has no account on the system.

1. EDA$ Official opens web page of the system.
2. EDA$ Official clicks register button.
3. EDA$ Official chooses its account type as individual or corporate.
4. EDA$ Official chooses a username and password.
5. EDA$ Official fills the related fields according to its account type.
6. EDA$ Official clicks confirm button and is directed to a page where s/he needs to enter a verification code.
7. System sends a mail that contains a verification code to e-mail address specified by EDA$ Official.
8. EDA$ Official enters the verification code to the system.

3.5.1.2 Add Device

Scenario Name: Add Device

Participating Actor Instances: EDAŞ Official:Client Admin, EDAŞ Laborer:Restricted Client

Assumptions: Client Admin has an account in the system.

1. EDAŞ Laborer places the device in front of a meter that s/he wants to monitor.
2. EDAŞ Laborer logins in to mobile application of the system on his/her smartphone.
3. EDAŞ Laborer clicks adding device.
4. The mobile application ensures EDAŞ Laborer to activate NFC feature of his/her smartphone.
5. The mobile application sends the device information and account information of EDAŞ Laborer adding the device to the system as a request.
6. Adding device request is displayed in “View Devices” page of web application for confirmation from EDAŞ Official.
7. EDAŞ Official confirms the request.

3.5.1.3 Add Alert and Set Its Actions

Scenario Name: Add Alert and Set Its Actions

Participating Actor Instances: EDAŞ Official:Client Admin

Assumptions: A device is already added to the account.

1. EDAŞ Official logins to the system.

2. EDA$ Official clicks “View Devices” button on home page of the web application.
3. EDA$ Official chooses a device to manage from the device table that contains the list of devices.
4. EDA$ Official clicks to “Manage Device” button on device page.
5. EDA$ Official clicks to “Add Alert” button, he/she chooses an alert type, adds contact information of individual responsible and defines threshold and majorant values that will be checked by the system.

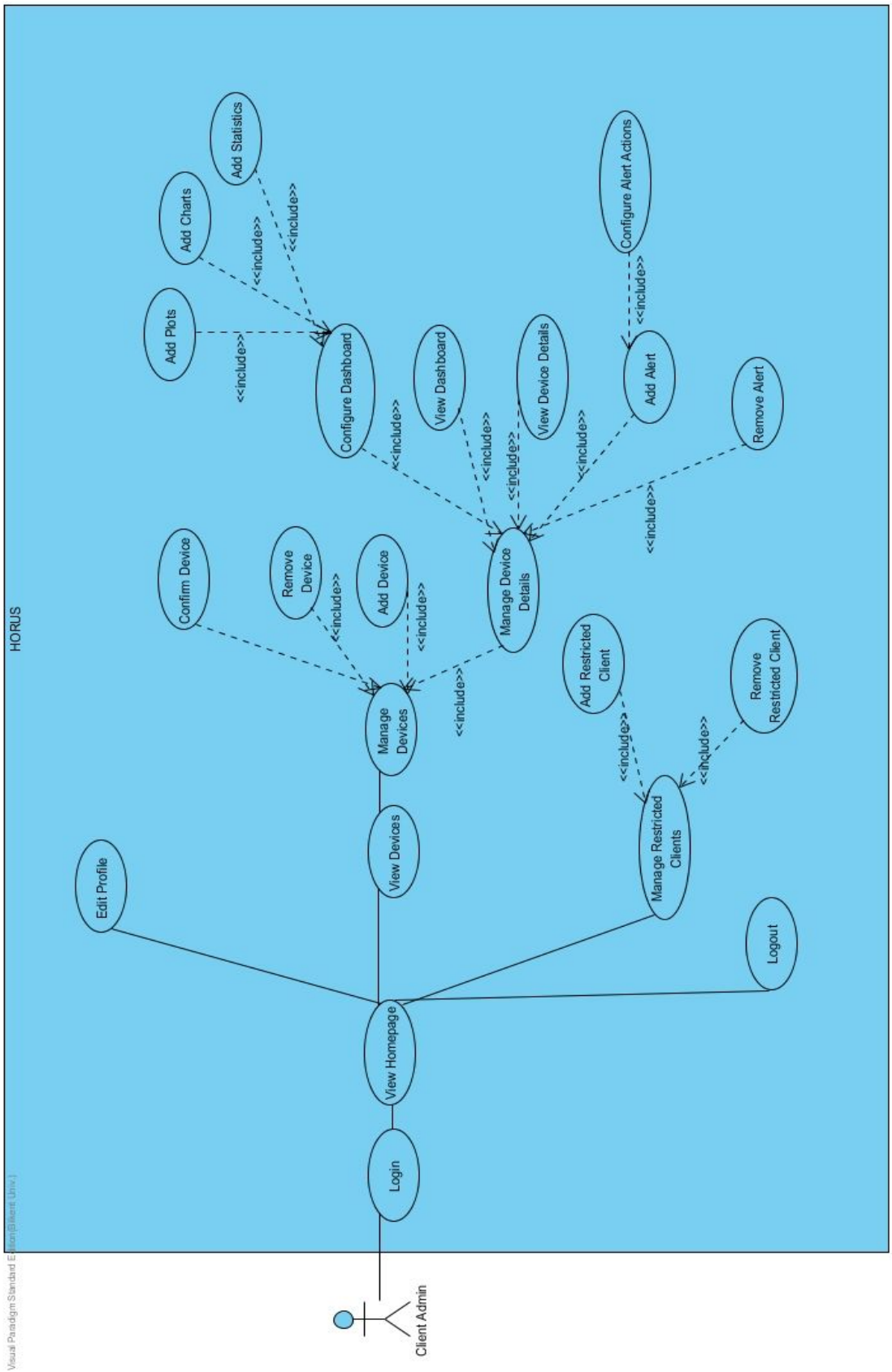
3.5.2 Use Case Model

In the system there are two types of users, the first one is the Client Admin user, which is the main account of the clients that is responsible of managing the devices, configuring the dashboards of the devices, editing client settings, managing restricted client accounts, and so on (adding and removing). The second type of user is the restricted client accounts that are tied with the main (admin) client account and also mainly managed by this client account. These accounts have view-only access to the system, and also they are able to add devices to the system with the scenario “Add Device”. These restricted client accounts are for distributing the responsibility of monitoring the devices to multiple users without giving any of them more access right than needed.

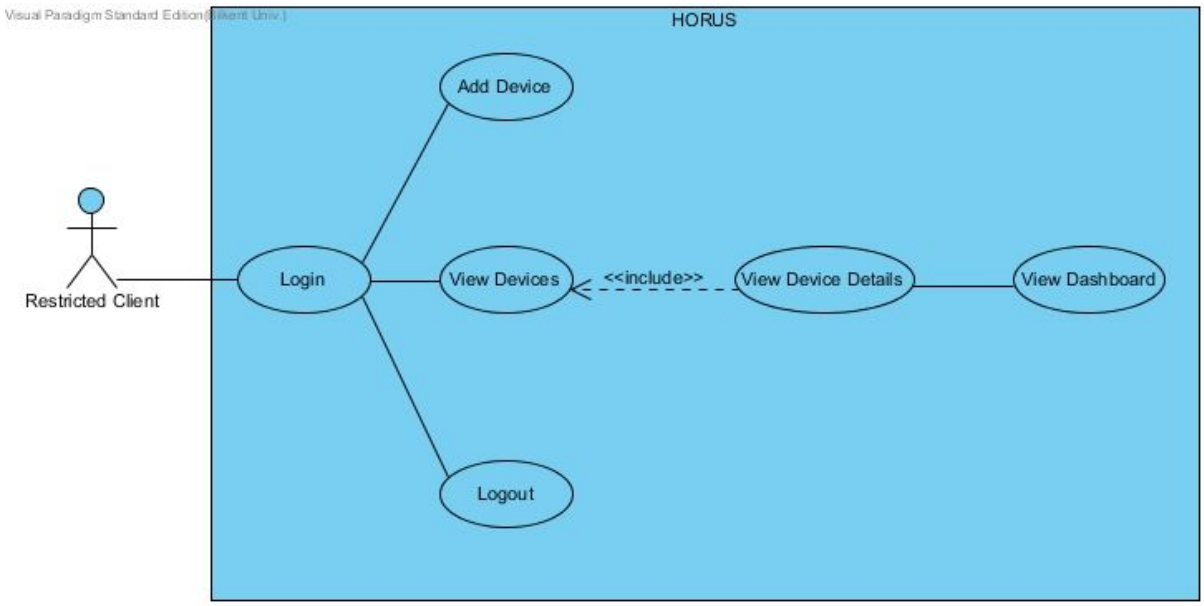
Since each of these have different use-cases on the system, two different use-case diagrams will be presented.

3.5.2.1 Use Case Diagrams

3.5.2.1.1 Client Admin Use Case Diagram



3.5.2.1.2 Restricted Client Use Case Diagram



3.5.2.2 Use Cases

3.5.2.2.1 Add Device

Use Case Name: Add Device

Primary Actor: Restricted Client

Stakeholders and Interests:

1. Restricted users (Restricted Client) should add devices.

Pre-condition:

1. User (Restricted Client) must be logged in to the android app from his/hers smart phone.
2. Restricted client must be within the reach of NFC area of appliance.

Post-condition: -

Entry Condition:

1. Restricted client clicks to “Add Device” button on the android app.

Exit Condition: Back-end server sends confirmation message to the android app.

Event Flow:

1. Restricted client should hold his smartphone near the appliance until the connection is established between smartphone and appliance.
2. When client sees the confirmation message regarding to the status of NFC module connection client should clicks “approve” button.
3. Android app sends the device information and the account information of the restricted client who is trying to add the device to the back-end server.

3.5.2.2.2 Confirm Device

Use Case Name: Confirm Device

Primary Actor: Client Admin

Stakeholders and Interests:

1. Privileged users (Client Admin) should confirm devices that are waiting to be added to the system.

Pre-condition:

1. At least one device must be added to the system by a client (Restricted client).
2. User (Client Admin) must be logged in to the web application.

Post-condition: -

Entry Condition: Client admin should click to a device that is on the “request list”, and clicks to “Manage Device” button.

Exit Condition: -

Event Flow:

1. Client admin will view the details of the “waiting” device, and the information regarding to the account who made the “add request”.
2. Client admin should click to “Confirm Device” button.
3. Back-end server will change the device’s status from “waiting” to “confirmed”.
4. Back-end server will add the device to “Device list”, so that it will be available for viewing later from “View Devices” page.

3.5.2.2.3 Add Restricted Clients

Use Case Name: Add Restricted Clients

Primary Actor: Client Admin

Stakeholders and Interests:

1. Privileged users (Client Admin) that should be able to add and remove restricted client accounts.
2. View only users (Restricted Clients) that only have read only access.

Pre-condition: Client Admin must be logged in.

Post-condition: A new restricted client has been added to the system.

Entry Condition: From homepage, client admin clicks the “Manage Restricted Clients” option, and opens the menu to add or remove restricted clients.

Exit Condition: Client admin clicks “Done” to finalize the operation.

Event Flow:

1. Client admin selects “Add Restricted Client” option from the menu.
2. Client admin enters the e-mail address of the desired restricted client that will be tied to current client account to give read only access to the system.
3. Client admin clicks “Done” button.

4. System generates a unique sign in link that will be sent to the given email address, which will be used by restricted client to sign in the system.
5. System sends the email with link to the email address.

3.5.2.2.4 Remove Restricted Clients

Use Case Name: Remove Restricted Clients

Primary Actor: Client Admin

Stakeholders and Interests:

1. Privileged users (Client Admin) that should be able to add and remove restricted client accounts.
2. View only users (Restricted Clients) that only have read only access.

Pre-condition: Client Admin must be logged in.

Post-condition: The existed restricted client has been removed from the system.

Entry Condition: From homepage, client admin clicks the “Manage Restricted Clients” option, and opens the menu to add or remove restricted clients.

Exit Condition: Client admin clicks “Done” to finalize the operation.

Event Flow:

1. Client admin selects “Remove Restricted Client” option from the menu.
2. From the list of previously added restricted clients, client admin selects the desired restricted client account.
3. Client admin clicks “Remove” button.
4. System removes the selected restricted client account.

3.5.2.2.5 Add Alert

Use Case Name: Add Alert

Primary Actor: Client Admin

Stakeholders and Interests:

1. Privileged users (Client Admin) that should be able to add alerts and configure alert actions.

Pre-condition: Client Admin must be logged in and must choose a device from device list to manage device details .

Post-condition: An alert for a specific device is created with threshold and majorant values.

Entry Condition: From view devices page, client admin clicks the “Manage Device Details” option.

Exit Condition: Client admin clicks “Done” to finalize the operation.

Event Flow:

1. Client admin selects “Add Alert” option from the Manage Device Details.
2. Client admin enters a threshold value and a majorant value where the device does not generate an alert.
3. Client admin chooses an alert type which determines the way of notifying the individual responsible when an alert generated by the device.
4. Client admin enters contact informations of individual responsible who must be notified about the emergency situation by the system.
5. Client admin clicks “Done” button.

3.5.2.2.6 Remove Alert

Use Case Name: Remove Alert

Primary Actor: Client Admin

Stakeholders and Interests:

1. Privileged users (Client Admin) that should be able to remove alerts.

Pre-condition: Client Admin must be logged in and must choose a device from device list to manage device details.

Post-condition: An alert for a specific device is removed.

Entry Condition: From view devices page, client admin clicks the “Manage Device Details” option.

Exit Condition: Client admin clicks “Done” to finalize the operation.

Event Flow:

1. Client admin selects “Remove Alert” option from the Manage Device Details.
2. From the list of previously added device alerts, client admin selects the desired device alert.
3. Client admin clicks “Remove” button.
4. System removes the selected device alert.

3.5.2.2.7 Configure Dashboard

Use Case Name: Configure Dashboard

Primary Actor: Client Admin

Stakeholders and Interests:

1. Privileged users (Client Admin) that should be able to add and remove restricted client accounts.
2. View only users (Restricted Clients) that only have read only access.

Pre-condition: Client Admin must be logged in.

Post-condition: A dashboard to monitor the selected device is created.

Entry Condition: Client admin is in “Manage Devices” menu.

Exit Condition: Client admin clicks “Done” to finalize the operation.

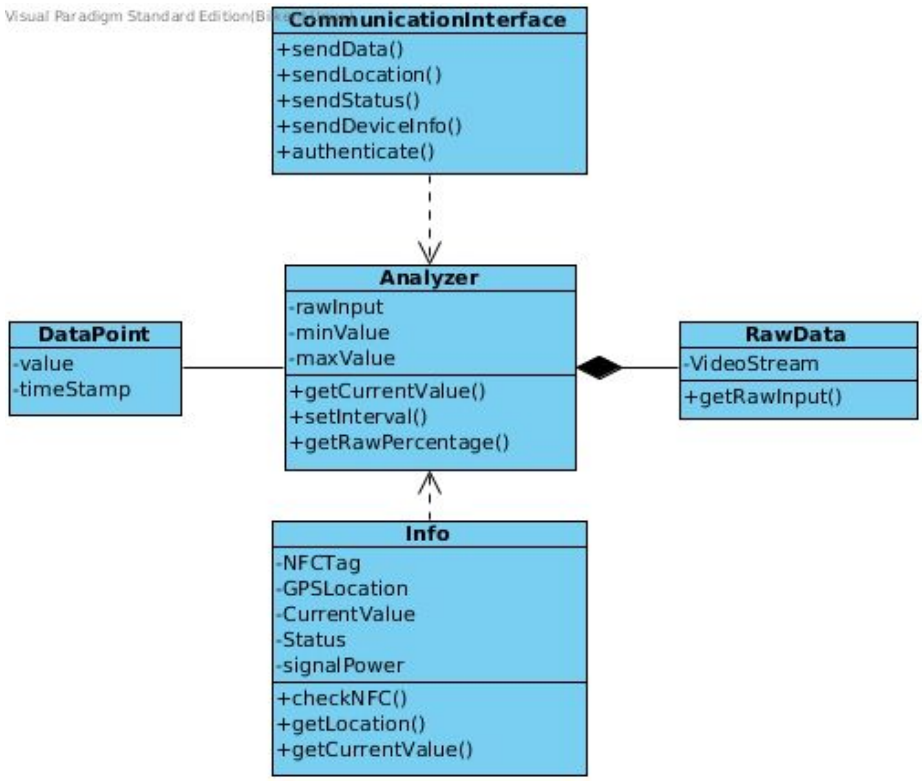
Event Flow:

1. Client admin selects the desired device.
2. Client admin selects the “Configure Dashboard” option, and opens the menu to configure the dashboard for the selected device.
3. An interactive menu is opened for client admin.
4. Client admin selects the desired dashboard elements and drags them to the desired location in the dashboard.
5. Client admin modifies the parameters and features for the added dashboard elements.
6. When client admin finished the configuration, clicks “Done” button to finalize the dashboard for the selected device.

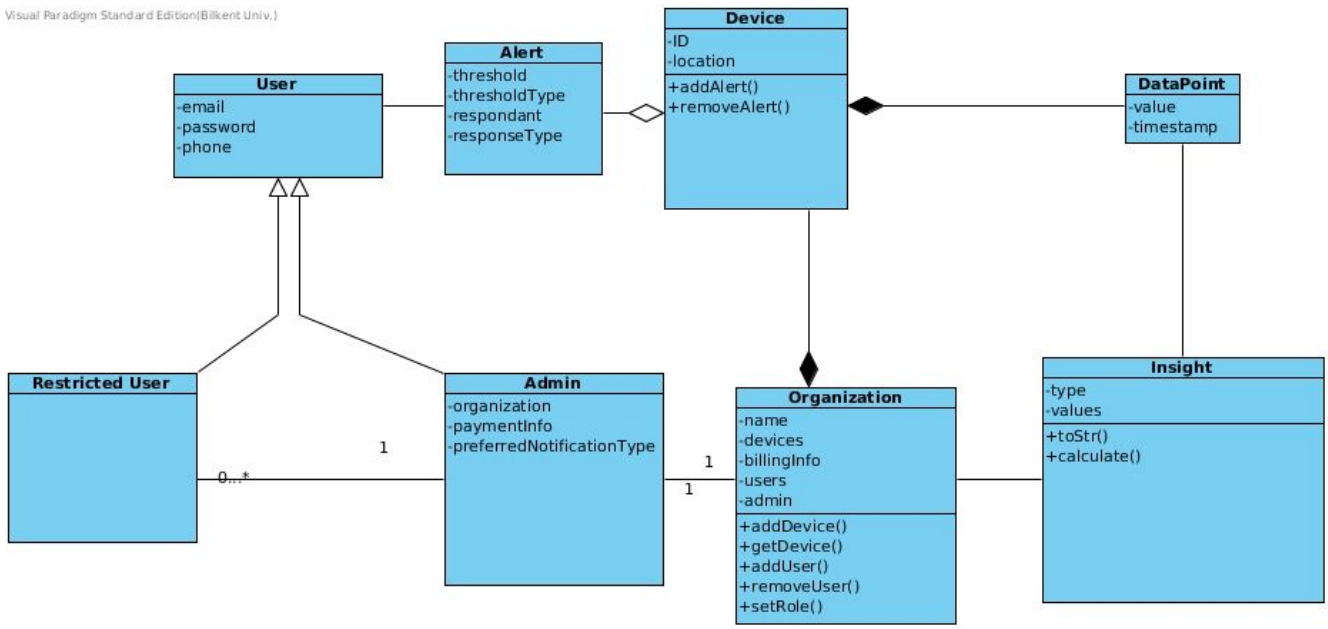
3.5.3 Object and Class Model

The object model will consist of three different object model diagrams. The first one is for the appliances (devices), and their high-level class implementation, the second one is for the back-end and its high-level class implementation and the third one is the small android application that will read the NFC tags in the appliances in order to tie them to the account of the client. (mobile authenticator)

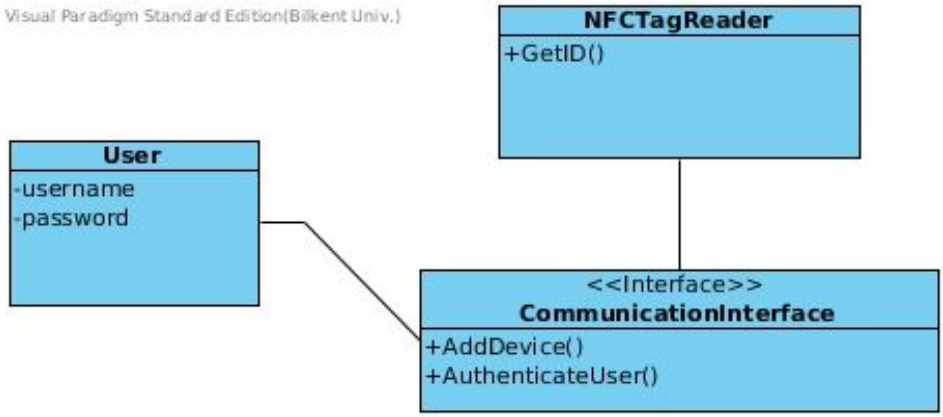
3.5.3.1 Appliances (Devices)



3.5.3.2 Back-end



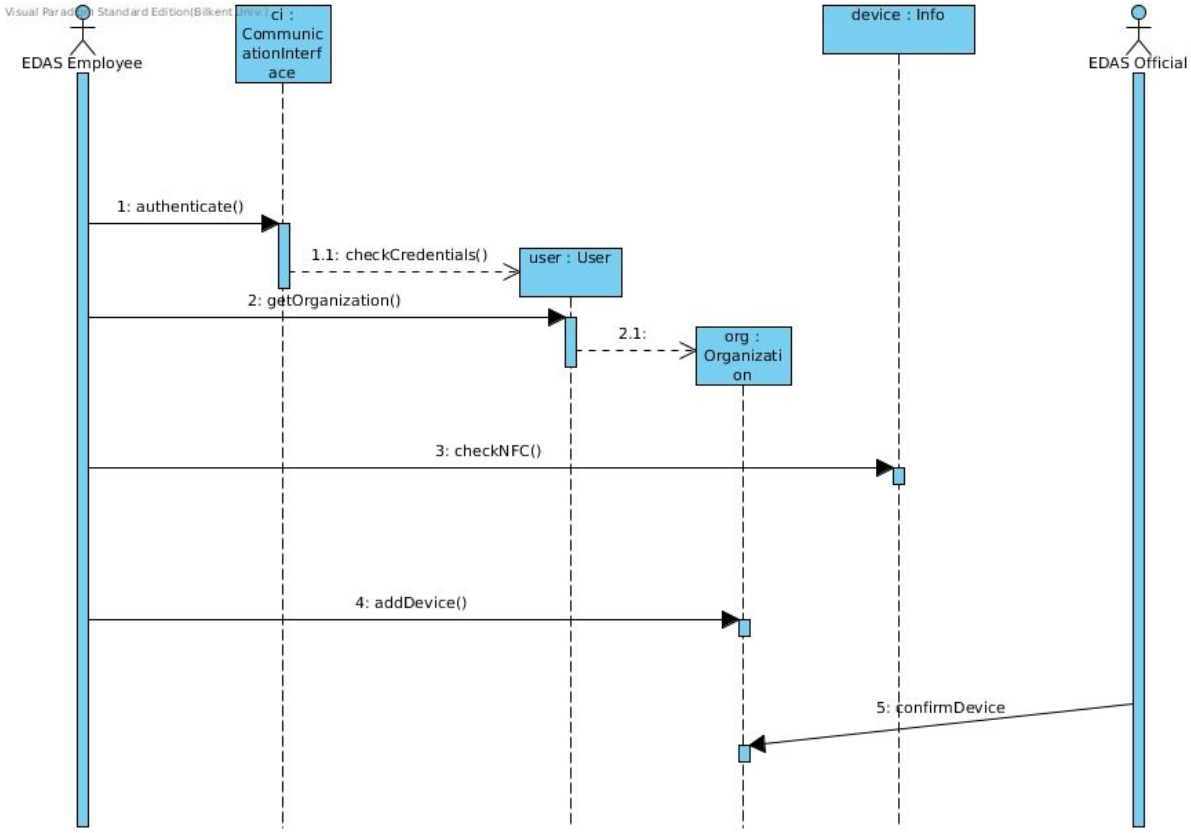
3.5.3.3 Mobile Authenticator



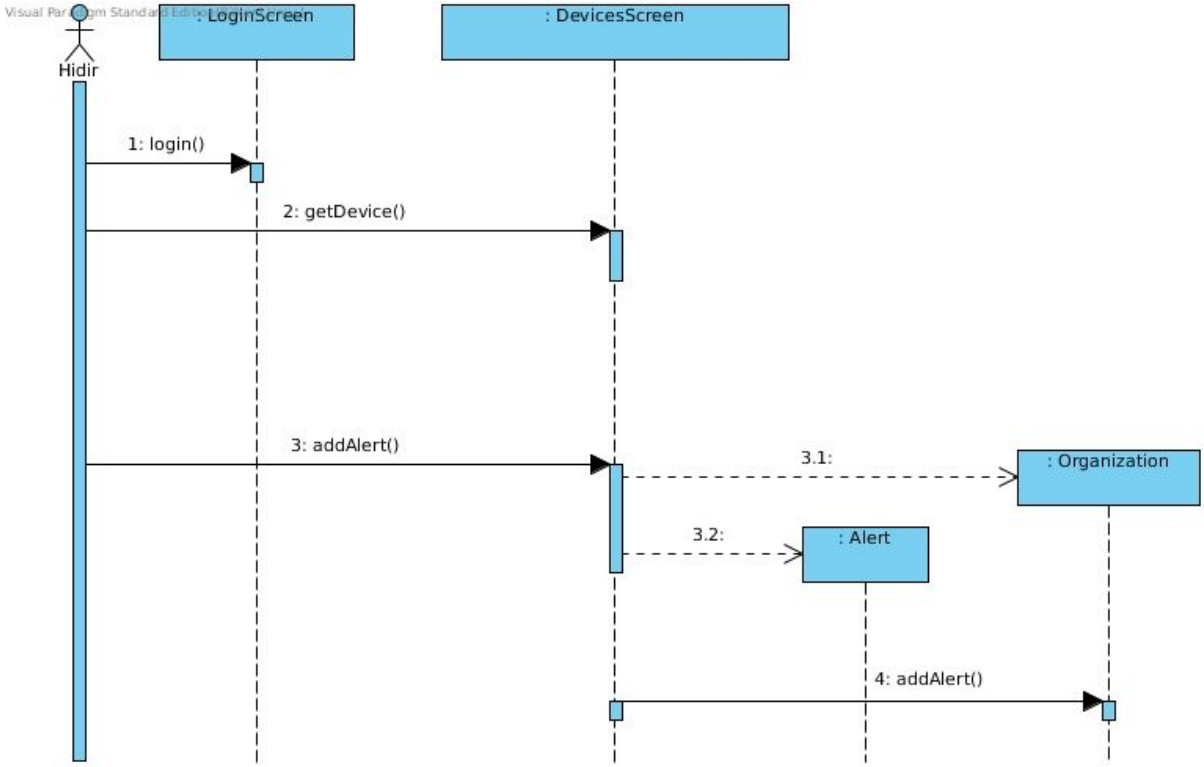
3.5.4 Dynamic Models

3.5.4.1 Sequence Diagrams

3.5.4.1.1 Add Device

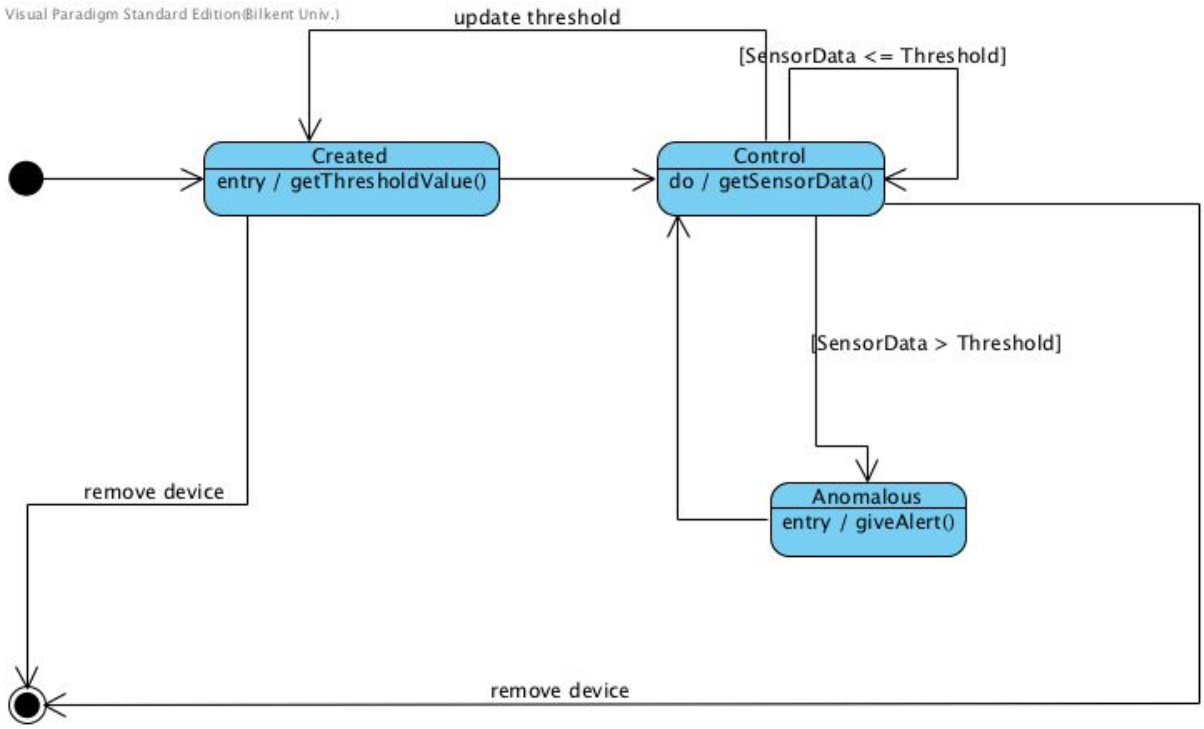


3.5.4.1.2 Add Alert



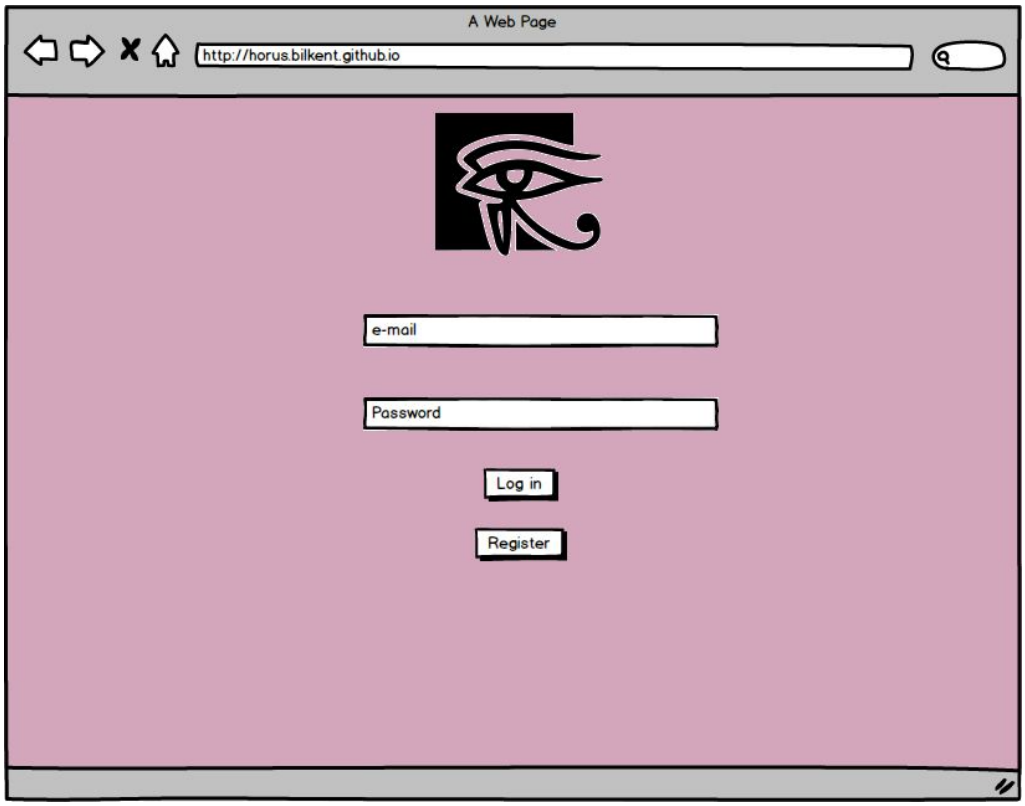
3.5.4.2 State Diagram

3.5.4.2.1 Device Object

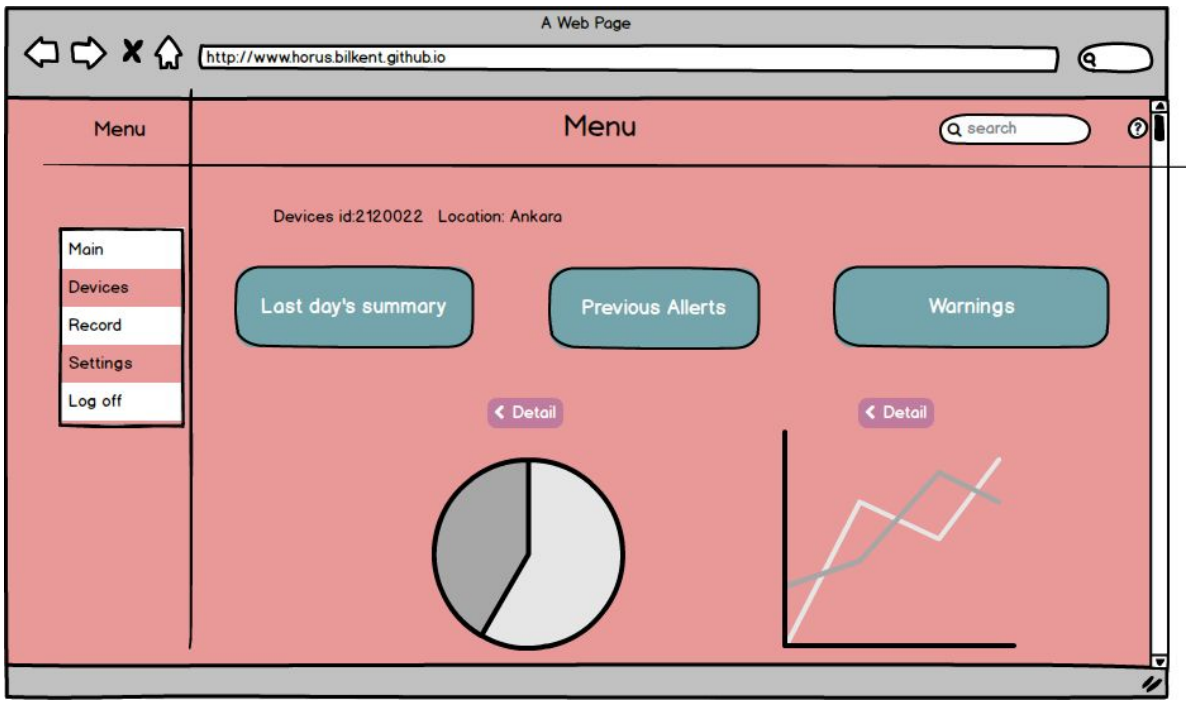


3.5.5 User Interface

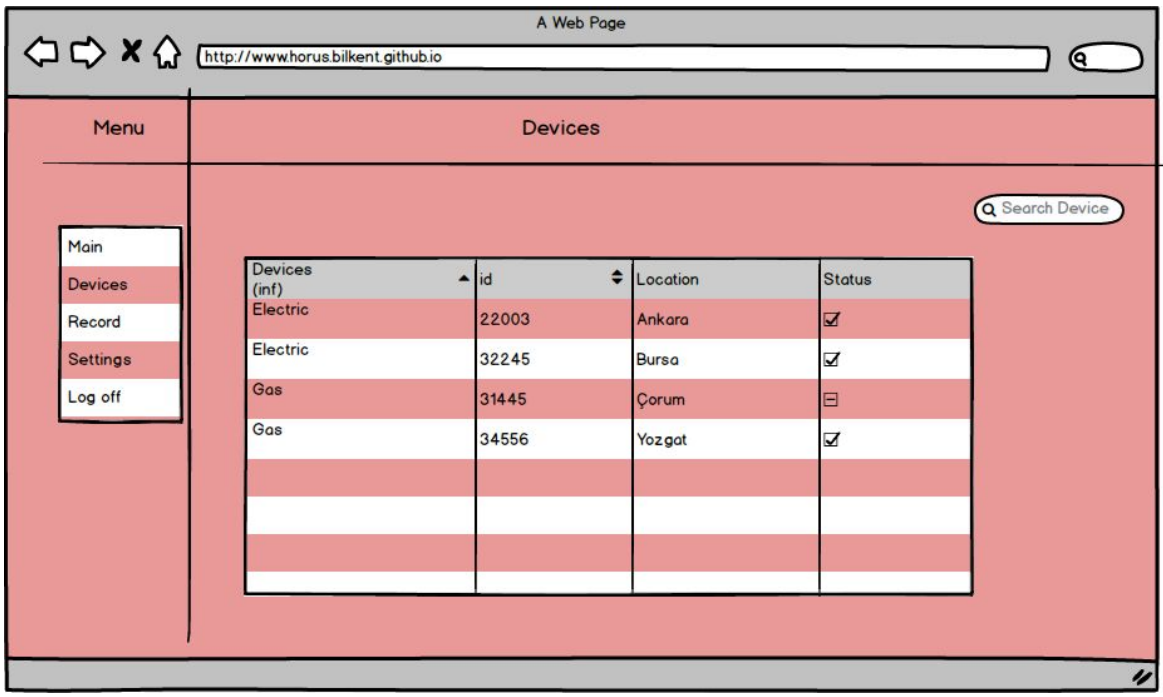
3.5.5.1 Login Screen



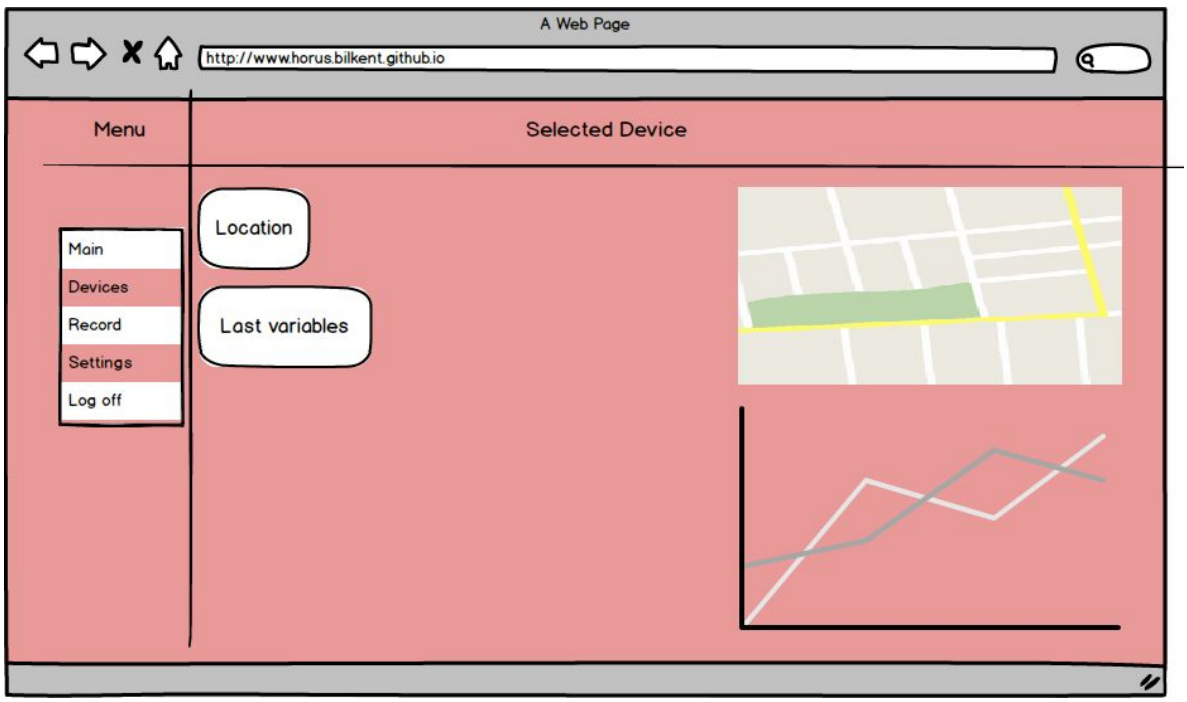
3.5.5.2 Homepage



3.5.5.3 Devices Screen

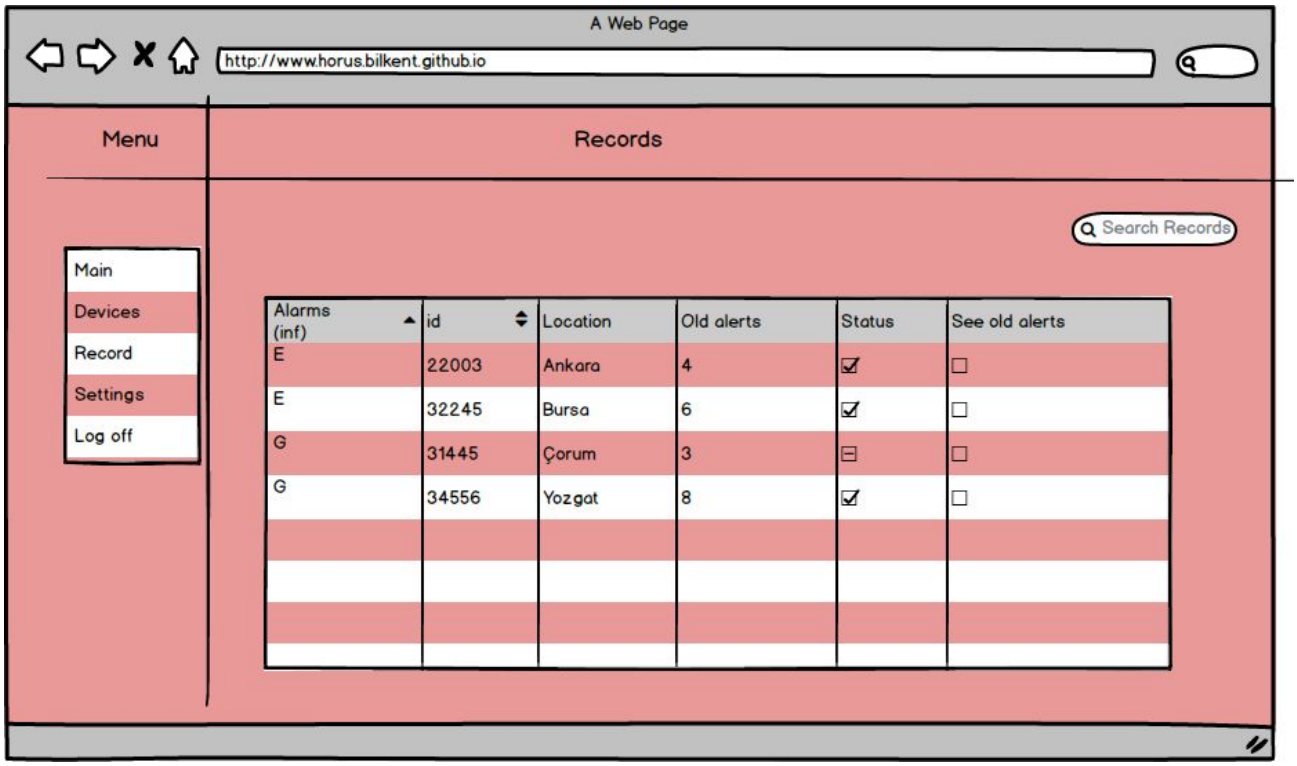


3.5.5.4 Device Details Screen



3.5.5.5 Device Settings Screen

3.5.5.6 Alarms Screen



4.0 References

[1]: www.helium.com
[2]: https://www.arduino.cc/en/Main/Products
[3]: https://beagleboard.org/black
[4]: https://www.raspberrypi.org/help/what-is-a-raspberry-pi/
[5]: http://thumbs.dreamstime.com/z/sensor-5293924.jpg