Sheffield Hallam University

SMART CITY

CSSD-Subtask 3

Group ID: CSSD-3

Perera P.A.H.E.	27045240	IT16083424
Wijesinghe W.M.I.O.	27045261	IT16051294
Abeyratne K.M.S.R.	27045230	IT16063488

TABLE OF CONTENTS

Chapter 1

Introduction

The main intention of designing this “SMART CITIES” system is to automate the infrastructure based systems of the cities in the world. This system plays a main role in the cities by performing the tasks simply and effectively by reducing the man effort. By digitalizing a system like this may help the City council or Mayor of the city and the people who live in the cities to save their valuable time by accomplishinging or carry out their tasks or fulfill their needs by using a single click. By designing our system using the latest technology, it will improve the sustainability and also helps the development of the economy. It will create a city with a smarter energy infrastructure.

Basically, in the system of SMART CITIES, there are four main roles who work on it namely, the User, Mayor-City Council, Paramedic and the waste collector. Currently, this project provides us with few number of sensors and also planning to extend the system in the future.

The Mayor-City council is the person who is responsible for managing the sensor stations. He can add the sensors to the needed stations. Whenever he selects the location or the station by using the google map, the longitudes and the latitudes (GPS coordinates) related to the selected location will be retrieved from the system. And then he should enter the station ID as well as the sensor station name in order to add the sensor station. Also he can delete the sensor station. The administrator also can view the added sensor stations.

By entering the sensor ID, he can add the particular sensor needed by the particular station after selecting the sensor description (bin sensor, flood sensor, traffic sensor), activity status and the frequency in the stage of adding a sensor, in this situation always the current data field is zero.

User cannot access the administrator’s data. He can access or view the data related to the flood and the traffic sensors. He can choose the sensor ID, station name, sensor description and also the data of the current situation (Current data). This data will be compared with the given frequencies of the sensors. Then the system alerts the people with an appropriate message when the frequency (current data) reached the given frequency level. The frequencies are not the same in every sensor. It differs from one sensor to another.

Paramedic can access or view the information related to the traffic sensors. Then he can decide the best route with the lesser traffic out of all the routes.

Waste collector is the person who empty the bins. He can access the data related to the traffic sensors in order to choose the route with the lesser traffic to reach his destination. And also he can access the data related to the bin sensors.

Chapter 2

UI Evaluation Report

2.1 The approach used to evaluation and justification

Before the finalization of the developed system, an evaluation for the user interfaces was required in order to collect new ideas and to get a feedback regarding the system. The proposed UI evaluation approach for our system was a **Heuristic Evaluation** method as it is fast, cheaper and it provides an outline analysis of the design. We used this approach in order to check the usability of our application. Because of this evaluation method, we can identify the minor and major threats in the design before we develop the final design. There are many proposed concepts or approaches discovered by the people. Out of all the popular approaches such as System Usability Scale(SUS), we choose the most popular and the trendy concept is **Nielsen's heuristics** invented in 1994 by Jacob Nielsen.

2.2 Evaluation Plan

When planning the evaluation for the smart city system, initially we have identified several important factors and questions that needs to be considered for the further development of the system. we used multiple evaluators (4 evaluators) in order to obtain the best results and responses. Because involving a single evaluator is not sufficient enough to identify all the usability problems in the design. Therefore, it is highly recommended to use multiple evaluators for this task as when the number of evaluators increases the number of problems identified will also increase.

After selecting the evaluators, we advised them about what exactly they meant to do in this evaluation. We asked them to inspect our system several times and fill the survey form individually by providing reports, comments or suggestions. We collected the survey forms and summarized the problems and suggestions after they completed the evaluation.

Factors of Nielsen's heuristics evaluation

1. **Visibility of system status**

   Checked whether our system always manages users by informing them about the current status and what is going on in the system at the moment by notifying them with field validations, confirmation messages, etc. within a given reasonable time.

2. **Match between the system and the real world**

   Checked whether the design contains clearly understandable words, phrases and the commands for the users on behalf of using confusing system oriented words and phrases.

3. **User control and freedom**

   Checked whether when a user mistakenly clicks on in case of an emergency situation in the system, he can easily go back without any confusions through an emergency exit.

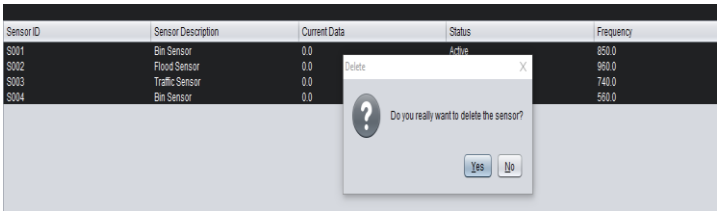
4. Consistency and standards

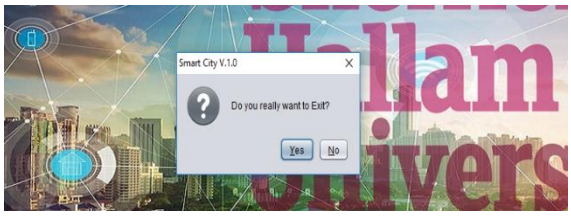
Checked whether the user interfaces are in a proper standard, familiarity of the words applied in interfaces, understandability of the situation and actions.

5. Error prevention

Checked whether the system minimizes the possibility of occurring the potential errors in by using only the legal commands.

Such as When working with the designed system, users may come up with several mistakes. They often choose system functions by accident. When such a thing happens, the system prompts confirmation messages and error messages. So the user can cancel the command given by him accidentally. This helps to avoid errors in the system at the first place.





6. Recognition rather than recall

Checked whether the users in the system do not have to remember the previous part's information while working on another part of the system. All the information is easily retrievable and visible. Such as when a user needs to add a sensor to a sensor station it is difficult for him to remember the data relevant to the particular station (station ID-As sensor station ID is difficult to remember). Therefor the system needs to display more attributes or data of all the stations available in the system when a user needs to do a selection. (station ID and sensor station name)

7. Flexibility and efficiency of use

Checked whether the system allows not only the experts, but also the inexperienced people to use our system without any difficulties. (In an easy and efficient manner to use by any user).

8. Aesthetic and minimalist design

Checked whether the system comprises only the important or less information on screens and presents the data in a manner that is very neat and clear. A lot of information on a screen will confuse the users and it will be difficult for them to understand the procedure of the system.

9. Help users recognize and recover from errors

Checked whether the system includes the error messages which can easily understand by all the users as we have used a very user friendly language and also the system will provide the necessary suggestions and solutions for the errors as well.

10. Help and documentation

Checked the system comprises a proper help and documentation services. Because a system without any documentation is more appropriate in the real world. But it may be necessary to use documentation as a guidance to the end users. So we have included a simple documentation for our

system in order to provide more information regarding the tasks and the list of steps in the system for the users.

2.3 Ethical Considerations

1. Confidentiality and privacy

Confidentiality is one of the important factors in ethical considerations. This means safeguarding the information in the system or protect the system's data from unauthorized access. The information of the users in our system cannot view by the external parties. The test results or comments obtained from the evaluators are anonymous. Therefore, the other parties cannot identify the identity of the particular evaluator.

2. Informed Consent

We have provided the evaluators with a brief knowledge of the concept and the procedure of our system and also informed about the survey or the evaluation conducted by us in order to test the design. The evaluation or the test was only conducted after the evaluators voluntarily gave their assent. They were allowed to leave the evaluation if they are not interested on the system at the beginning.

2.4 Report on Evaluation Conducted

Problem	Description	Heuristics	Proposed solutions for the problem
Lack of previous tabs	In the system there are some important places where the previous tabs are not enabled. (making accessible)	**3**	Enable to go to the previous tab to view sensor station UI, station management UI.
No help and documentation services	In the system, there are some difficulties when using the system as there is no help and documentation.	**10**	Add a simple help documentation.
Font sizes	Implemented font size is too small	**4**	Resize the fonts in visible sizes and maintain the consistency of the font sizes in all the interfaces.

2.5 Results and Re-Design Suggestions

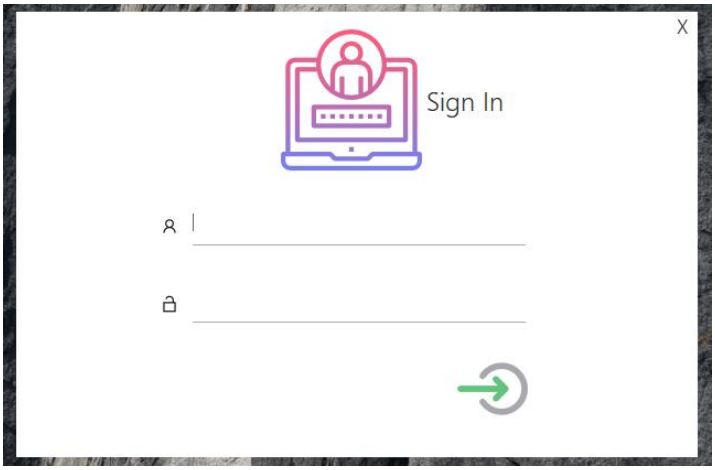
1. **Avoid the problem of “Lack of previous tab options” by enabling previous tabs options.**

According to the results obtained through the evaluation report, we planned to add more previous tabs options as it will be easier for the end users to work with the system. So we suggest to implement previous tabs options for some of the interfaces. We hope that it will save the time of the users.

2. **Avoid the problem of "no help and documentation" by adding a user guidance to the system.**
   This will help the new users or the inexperienced users to work with the system in a proper and easy manner with the provided guidance regarding the system.

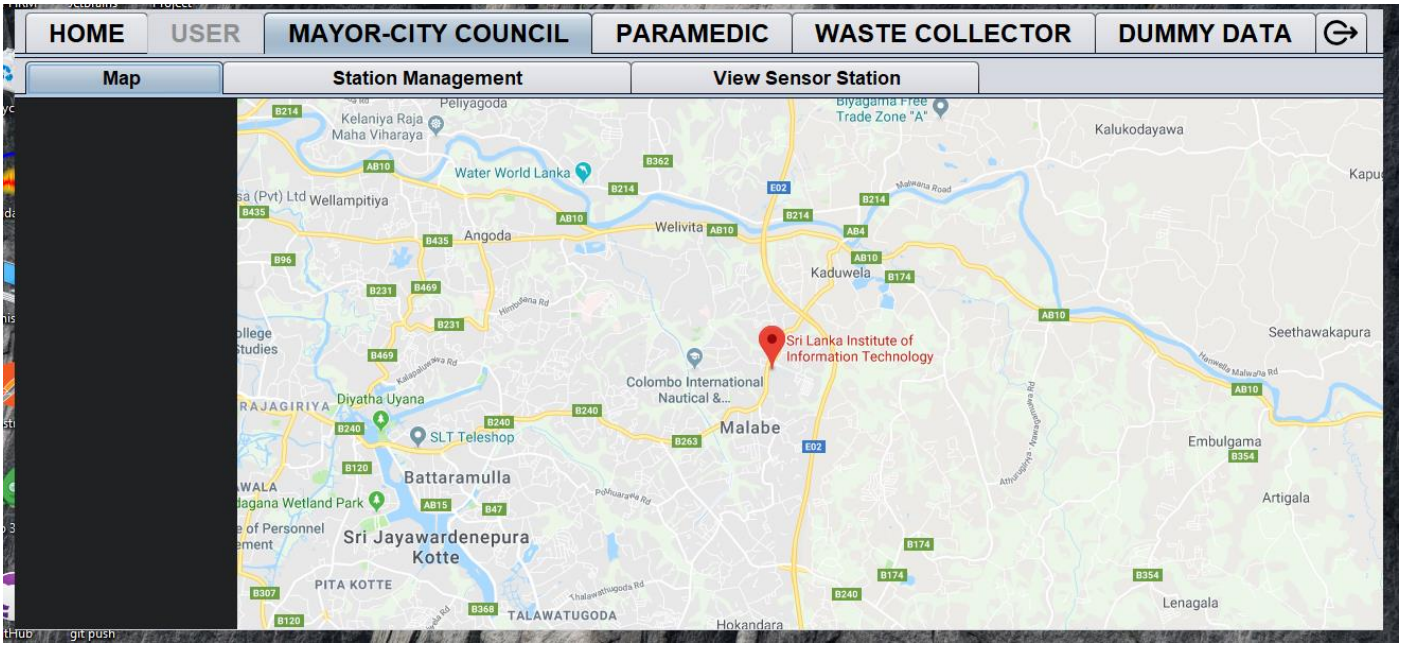
3. **Avoid the problem of "font size" by changing the font sizes.**
   This will allow users to view the content in the interfaces clearly.

Some UI illustrations-

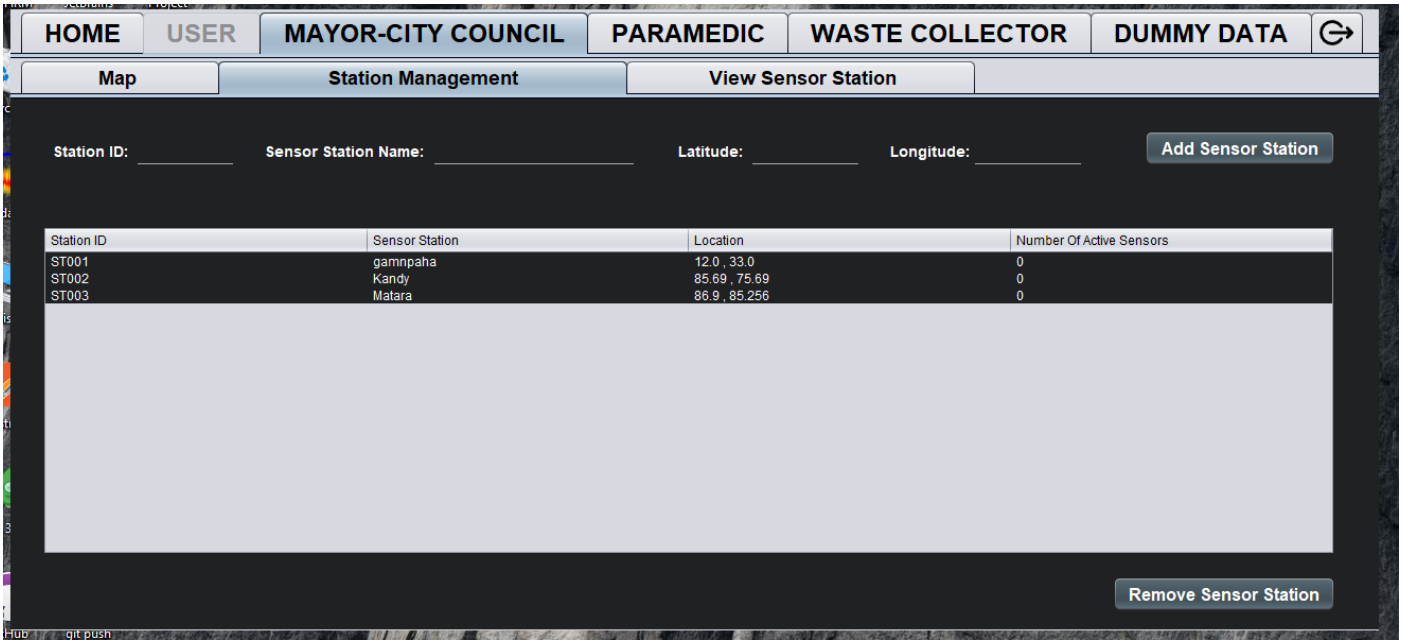


-System login Ui-

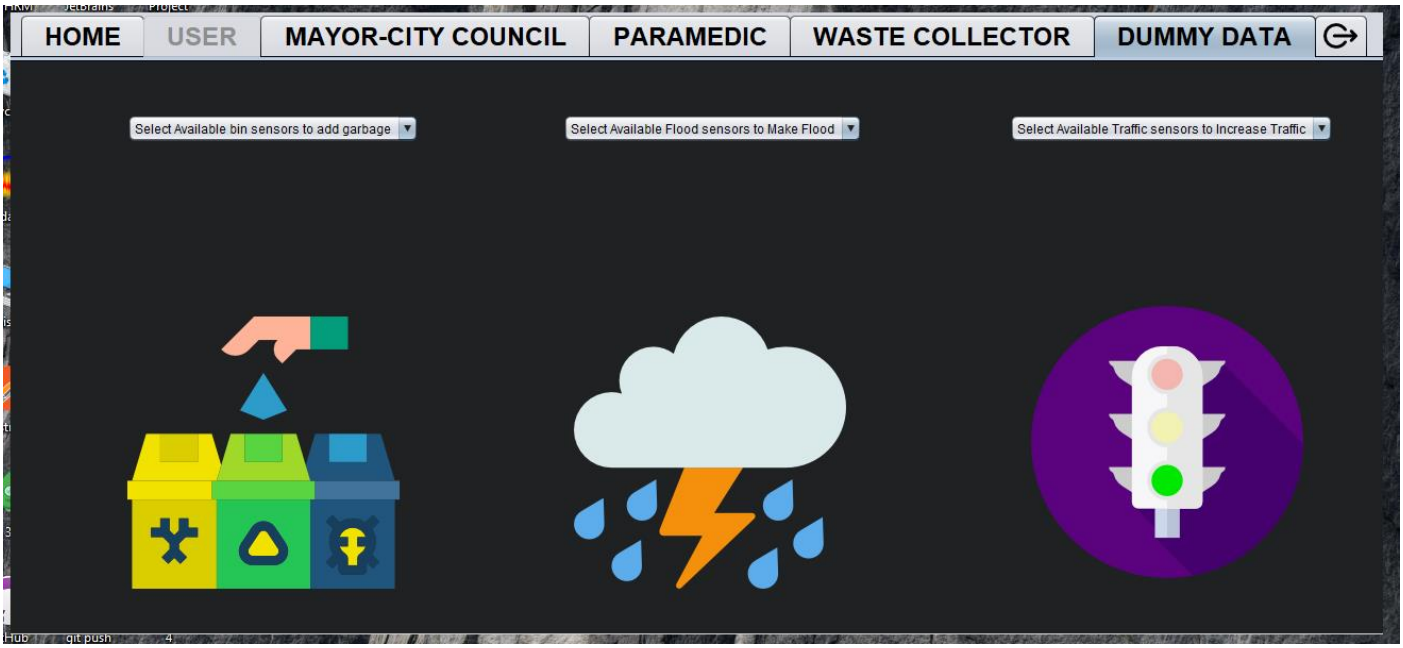
-Main Tab(Dashboard) UI that contain Different User Roles-



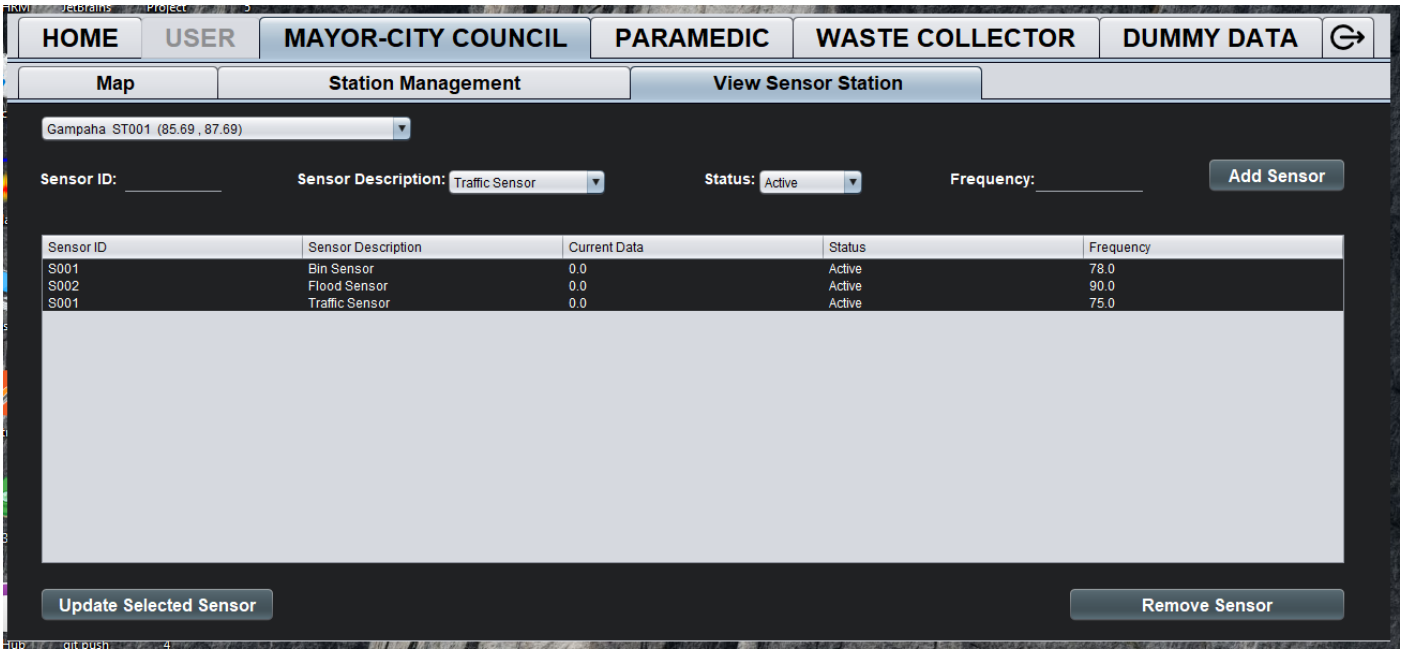
-Mayor Can See the map to place sensor stations-



-Mayor can Add Sensor Stations-



-For the testing Purpose we created dummy Data-



-Mayor Can Add Sensors to Selected Locations-

Acceptance Testing

Requirement	Test Case	Description	Objective
System login	1.1	Direct to dashboard by giving username and password	Verify whether the user can log into the system successfully.(verify username and password)
Add a sensor station	2.1	Decide a location and add a sensor station	Verify whether the system can add the sensor station successfully.
Remove a sensor station	2.2	Decide a sensor station and delete	Verify whether the system can delete the sensor station successfully.
Add a Sensor	3.1	Place a senor on an appropriate location	Verify the operation of placing a Sensor on required location successfully.
Remove a sensor	3.2	Decide a sensor and delete	Verify whether the system can delete the sensor successfully
Empty the bins	4.1	Decide which bin is needed to empty	Verify the operation of cleaning bins.

Test Case ID	1.1			
Pre-Conditions				
Action	Inputs	Expected Output	Actual Output	Test Result
1.Enter Username 2.Enter Password 3.Click login button	Username, Password	User should be logged in to the application	User is Successfully logged in to the application	Passed

Test Case ID	2.1			
Pre-Conditions				
Action	Inputs	Expected Output	Actual Output	Test Result
1.Enter station ID 2.Enter Sensor Station name 3.Enter latitude and longitude 4.Click add sensor station button.	stationID, Sensor Station name, Latitude, longitude	Should display the newly added sensor station on the grid.	Sensor station is successfully added to the application	Passed

Test Case ID	**2.2**			
Pre-Conditions	**Should have previously added sensor stations on the grid**			
Action	Inputs	Expected Output	Actual Output	Test Result
1.Click on the relevant row on the grid. 2.Click remove sensor station button.		**Should delete the selected sensor station on the grid.**	**Sensor station is successfully deleted from the application**	**Passed**

Test Case ID	**3.1**			
Pre-Conditions	**Should select a sensor station first**			
Action	Inputs	Expected Output	Actual Output	Test Result
1.Select the sensor station from combo box 2.Enter Sensor ID 3.Select Sensor description 4.Chose the status 5.Enter the Sensor frequency 3.Click Add Sensor button	**SensorID, SensorDescription, Status, Frequency**	**Should display the newly added sensor on the grid.**	**Sensor is successfully added to the sensor station**	**Passed**

Test Case ID	**3.2**			
Pre-Conditions	**Should have previously added sensor on the grid**			
Action	Inputs	Expected Output	Actual Output	Test Result
1.Click on the relevant row on the grid. 2.Click remove sensor button.		**Should delete the selected sensor on the grid.**	**Sensor is successfully deleted from the sensor station.**	**Passed**

Test Case ID	**4.1**			
Pre-Conditions	**Should select a sensor first**			
Action	Inputs	Expected Output	Actual Output	Test Result
1.Click on the relevant row on the grid. 2.Click Empty Bin button.		**Should empty the selected sensor data.**	**Sensor data is successfully removed from the sensor.**	**Passed**

Critical Reflection Report

We have implemented the entire class structure as per class diagram and also all the functionalities of sequence designs. Beyond that we made some changes to the implementation of the system.

- **Implemented a background thread.**
  Each SensorMonitor gathered data time to time from the particular sensor. For that purpose, we set a wake up time. Each SensorMonitors are observers to the Clock.java class. Therefor there should be a proper way to notify the observers to take readings during the lifetime of the system. To overcome that requirement we used the background thread in Clock.java class. Because of using the background thread, it makes the prototype more realistic as changes (increases and decreases) of the sensor data will be displayed time to time.

- **Applied the MVC architecture.**
  To separate the models, views and controllers we have implemented the application using the MVC architecture. For that purpose, moreover we added controller classes to the existing design. It makes the development process faster.

- **Implemented a common subject and observer interface for observe patterns.**
  All the concrete subjects override register, unregister, notify methods from subject interface. All the observers override update method from observer interface. Therefor it allows sending data to other objects effectively without any changes.

- **Made some classes singleton.**
  We applied the singleton pattern for some controller classes which we have created in order to provide global point of access to the object.

- Implement Validator Class
  To validate user inputs, we create a class that contain regular expression

- We used all the attributes in the given class diagram. Rather than that we added few extra attributes and methods to develop the system more accurately.

We remastered the user interface without affecting the existing design by increasing the usability and user friendliness. Few changes we made are mentioned below.

- Implemented the whole application in one form by using tab panes rather than using separate forms as in the given wireframes. It allows the users to work with the system appropriately.

- As the scenario is regarding a smart city and it should be in an attractive way to the user, we used animated icons and appropriate images.

- We designed an interface for the waste collector which is not there in the given wireframes. Using this waste collector interface, he can view the notifications of the bins which are full of garbage. Also he can view the traffic monitor.

- For the purpose of demonstration of the smart city prototype, we designed a separate tab called dummy data which allow add test data to the particular sensor.

- We adjusted the font sizes to make the appearance clear.

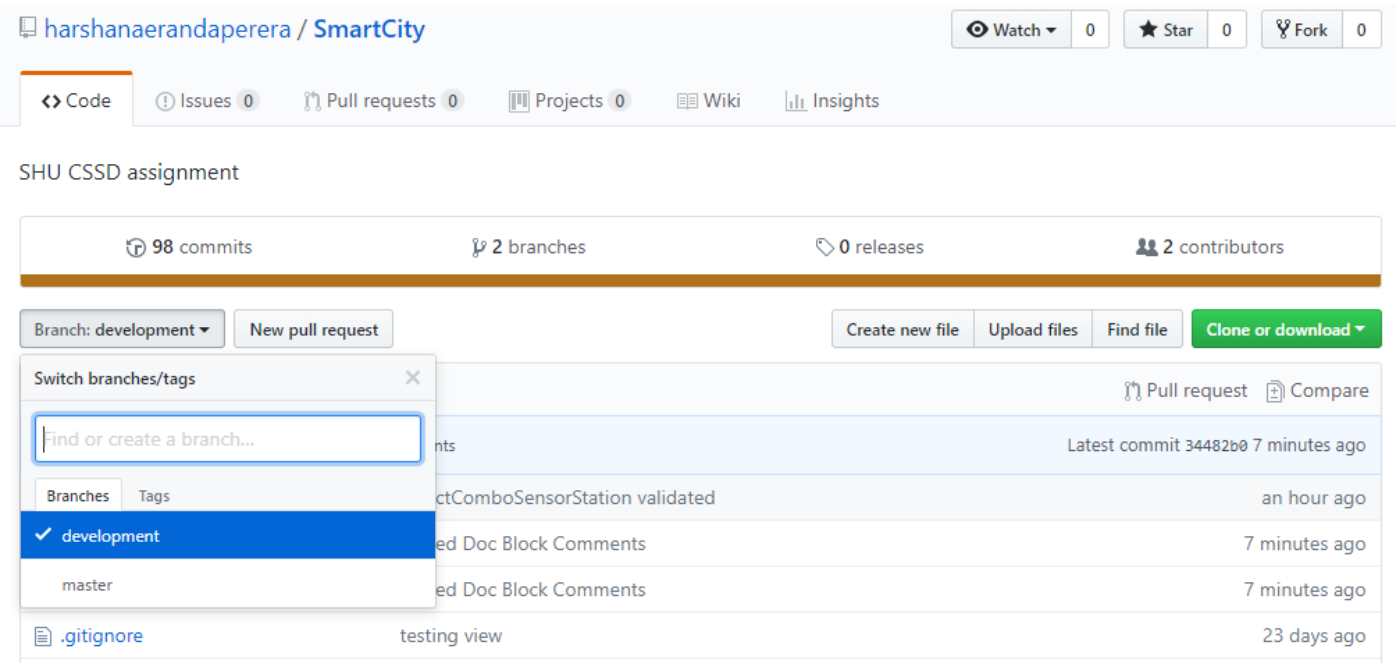
Selecting a project management strategy and useful tools were very critical for the project. This is mainly due to the experimental nature of the project. The below are the main tools that helped to carry out the project reducing frictions of many angles.

- GitHub
- Google Drive
- NetBeans, TODO plugin
- GitKraken

Since the project was a group work, we choose trello to manage the project task. For the communication purposes among group members (code issues, bug fixes, to view announcements) we used discord.

Also we maintained a private git repository on GitHub for the purpose of team coding.

Since our project is a three membered group, we used agile methodology.



Appendices

As our system is a desktop application, the most suitable method to conduct the survey was the paper-based method. We have created a survey form and distributed it among the selected evaluators after they completed the inspection on our design. Then they gave their individual responses and suggestions through the survey form.

Survey Form

Nielsen's heuristics evaluation on Smart City System

Please put a ✓ n the selected box and if you have any comments or suggestions regarding our system please write them down under the comments and suggestions columns.

1- Excellent
2- Very good
3- Good
4- Fair
5- Poor

HEURISTICS	1	2	3	4	5	COMMENTS	SUGGESTIONS
i. Visibility of system status							
ii. Match between the system and the real world							
iii. User control and freedom							
iv. Consistency and standards							
v. Error Prevention							
vi. Recognition rather than recall							
vii. Flexibility and efficiency of use							
viii. Aesthetic and minimalist design							
ix. Help users recognize and recover from errors							
x. Help and documentation							