Deliverable 1

Quantum Quants

Housefy

Team 04

Deliverable 1

Team members

Member 1: Wenyuan Yu, Student ID: n01403697

Member 2: Kyrylo Lvov, Student ID: n01414058

Member 3: Artem Tsurkan, Student ID: n01414146

Table of Contents

1. Team Contract

Team Name: Quantum QuantsTeam

Number: Team 04

Project Name: Smart Home Environment and Energy Monitoring System (Housefy)

Please negotiate, sign, scan and include as the first section in your Deliverable 1.

Please note that if cheating is discovered in a group assignment each member will be charged with a cheating offense regardless of their involvement in the offense. Each member will receive the appropriate sanction based on their individual academic honesty history.

Please ensure that you understand the importance of academic honesty. Each member of the group is responsible to ensure the academic integrity of all of the submitted work, not just their own part. Placing your name on a submission indicates that you take responsibility for its content.

For further information read Academic Honesty Policy on https://humber.ca/legal-and-risk-management/policies/search-by-students.html.

Team Member Names (Please Print)	Signatures	Student ID
Project Leader: Artem Tsurkan	Artem Tsurkan	n01414146
Wenyuan Yu	Wenyuan Yu	n01403697
Kyrylo Lvov	Kyrylo Lvov	n01414058

By signing this contract, we acknowledge having read the Humber Academic Honesty Policy as per the link below.

https://academic-regulations.humber.ca/2018-2019/17.0-ACADEMIC-MISCONDUCT

Responsibilities of the Project Leader include:

- Assigning tasks to other team members, including self, in a fair and equitable manner.
- Ensuring work is completed with accuracy, completeness and timeliness.
- Planning for task completion to ensure timelines are met
- Any other duties as deemed necessary for project completion

What we will do if . . .

Remark: Symbol "X" is used to indicate selection

Deliverable 1

Scenario	**Accepted initials**	**We agree to do the following**
Team member does not deliver component on time due to severe illness or extreme personal problem	Artem Tsurkan Wenyuan Yu Kyrylo Lvov	a) Team absorbs workload temporarily **X** d) Other:
Team member cannot deliver component on time due to lack of ability	Artem Tsurkan Wenyuan Yu Kyrylo Lvov	a) Team reassigns component **X** b) Team helps member __ b) Team "fires" team member by not permitting his/her name on submission __ d) Other:
Team member does not deliver component on time due to lack of effort	Artem Tsurkan Wenyuan Yu Kyrylo Lvov	a) Team absorbs workload **X** b) Team "fires" team member by not permitting his/her name on submission __ c) Other:
Team member does not attend team meeting	Artem Tsurkan Wenyuan Yu Kyrylo Lvov	a) Team proceeds without him/her and will assign work to the absent member **X** b) Team doesn't proceed and records team member's absence __ c) Team proceeds for that meeting but "fires" member after __ occurrences __
An unforeseen constraint occurs after the deliverable has been allocated and scheduled (a surprise test or assignment)	Artem Tsurkan Wenyuan Yu Kyrylo Lvov	a) Team meets and reschedules deliverable __ b) Team will cope with constraint **X** c) Other:
Team cannot achieve consensus leaving one member feeling	Artem Tsurkan	a) Team agrees to abide by majority vote **X**

Scenario	Accepted initials	We agree to do the following
"railroaded", "ignored", or "frustrated" with a decision which affects all parties	Wenyuan Yu Kyrylo Lvov	b) Team flips coin __ c) Other:
Team members do not share expectations for grade desired	Artem Tsurkan Wenyuan Yu Kyrylo Lvov	a) Team will elect one person as "standards-bearer" who has the right to ask that work be redone **X** b) Team votes on each submission's quality __ c) Team will ask for individual marking and will identify sections by author __ d) Other:
Team member behaves in an unprofessional manner by being rude or uncooperative	Artem Tsurkan Wenyuan Yu Kyrylo Lvov	a) Team attempts to resolve the issue by airing the problem at team meeting **X** b) Team ignores behaviour __ c) Team agrees to avoid use of all vocabulary inappropriate to the business setting __ d) Team fires the team member.
Team member assumes or requests that his/her name be signed to a submission but has not participated in production of the deliverable	Artem Tsurkan Wenyuan Yu Kyrylo Lvov	a) Team agrees that this is cheating and is unethical __ b) Friends are friends and should help each other __ c) That person name will not be put on the

Scenario	**Accepted initials**	**We agree to do the following**
		submission. **X**
There is a dominant team member who is content to make all decisions on the team's behalf leaving some team members feeling like subordinates rather than equal members	Artem Tsurkan Wenyuan Yu Kyrylo Lvov	a) Team will actively solicit consensus on all decisions which affect project direction by asking for each member's decision and vote **X** b) Team will express subordination feelings and attempt to resolve issue __ c) Other:
Team has a member who refuses to participate in decision making but complains to others that s/he wasn't consulted	Artem Tsurkan Wenyuan Yu Kyrylo Lvov	a) Team forces decision sharing by routinely voting on all issues __ b) Team routinely checks with each other about perceived roles __ c) Team discusses the matter at team meeting **X**

2. Participant signatures

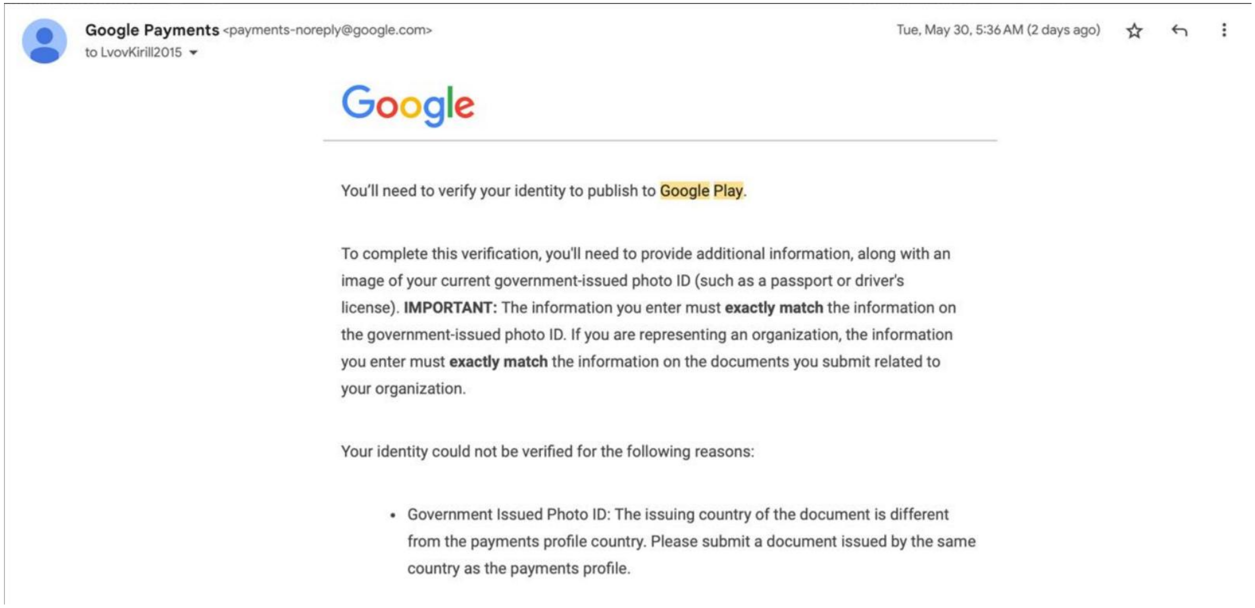
Artem Tsurkan

Wenyuan Yu

Kyrylo Lvov

3. Google Play Developer Account

One of our team member's got rejected:



Google Payments <payments-noreply@google.com>
to LvovKirill2015
Tue, May 30, 5:36 AM (2 days ago)

Google

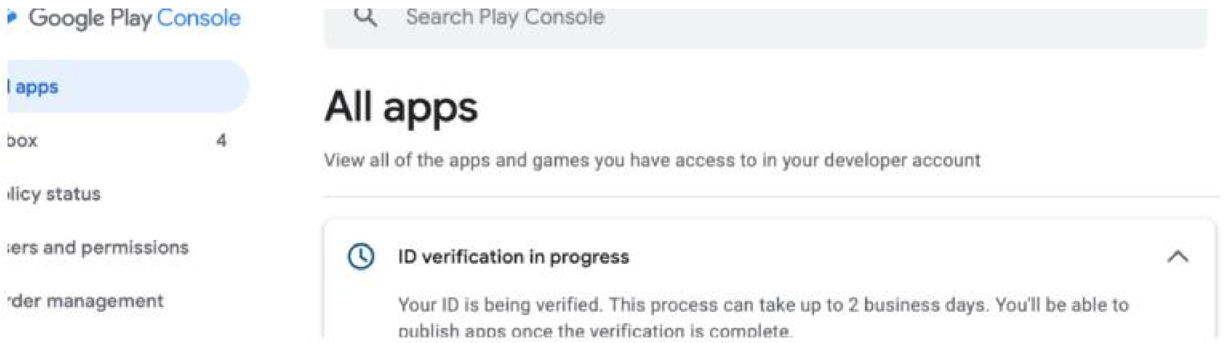
You'll need to verify your identity to publish to Google Play.

To complete this verification, you'll need to provide additional information, along with an image of your current government-issued photo ID (such as a passport or driver's license). **IMPORTANT:** The information you enter must **exactly match** the information on the government-issued photo ID. If you are representing an organization, the information you enter must **exactly match** the information on the documents you submit related to your organization.

Your identity could not be verified for the following reasons:

- Government Issued Photo ID: The issuing country of the document is different from the payments profile country. Please submit a document issued by the same country as the payments profile.

Due to this reason, we submitted another application for another team member and are still waiting for approval:



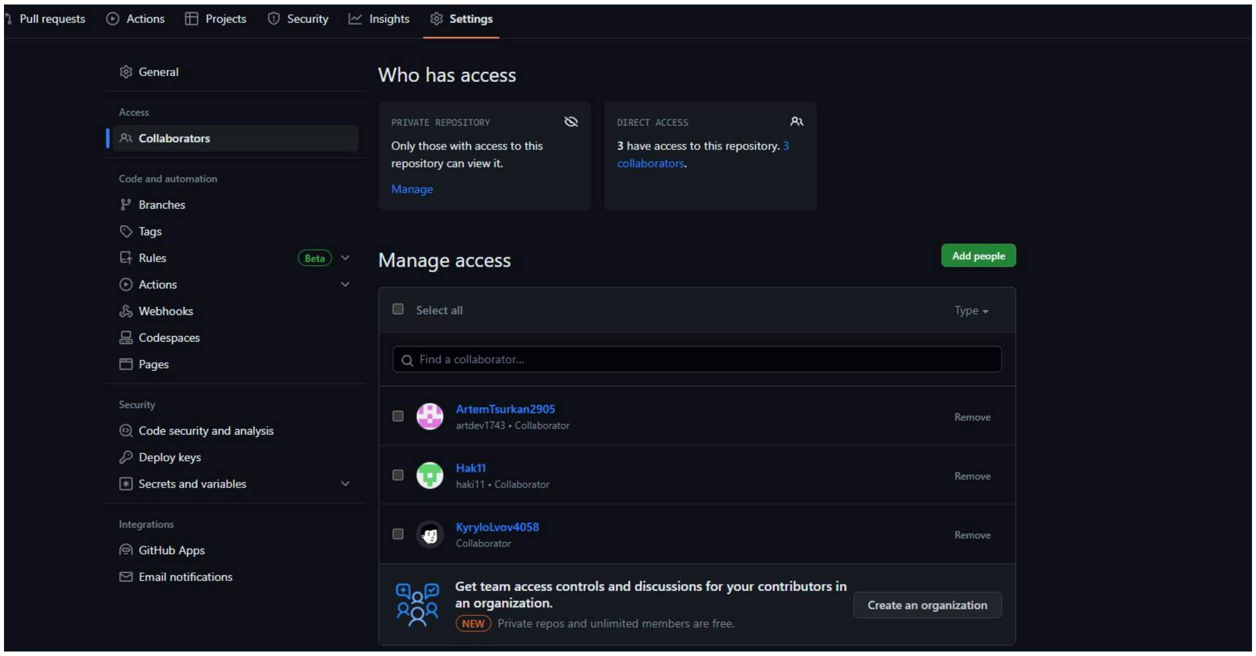
All apps

View all of the apps and games you have access to in your developer account

ID verification in progress

Your ID is being verified. This process can take up to 2 business days. You'll be able to publish apps once the verification is complete.

4. GitHub Invitation



Who has access

PRIVATE REPOSITORY
Only those with access to this repository can view it.
Manage

DIRECT ACCESS
3 have access to this repository. 3 collaborators.

Manage access

ArtemTsurkan2905
artdev1743 • Collaborator

Hak11
haki11 • Collaborator

KyryloLvov4058
Collaborator

Get team access controls and discussions for your contributors in an organization.
NEW Private repos and unlimited members are free.

Git Hub repo link:

https://github.com/WenyuanYu3697/housefy

5. Project Background and Description

5.1. Project Goals and Final Version:

The main goal of the project is to develop an Internet of Things (IoT) setup comprising of an Android application and a Raspberry Pi system, equipped with environmental sensors and devices for improving user comfort and providing insights on energy consumption. The system should be able to:

1. Monitor the Environment: This includes measuring air quality, temperature, humidity, energy consumption, and light levels, using the corresponding sensors.
2. Control Devices Based on Sensor Data: Based on the data received from the sensors, the system should automatically control connected devices like the LED lamp and the motor acting as an AC. The LED lamp's brightness should adjust according to the ambient light levels, and the motor should activate when the temperature crosses a certain threshold.

3. Provide User Controls: Through the Android application, users should be able to manually control the devices (LED lamp and motor). This includes adjusting the LED lamp brightness and turning the motor on/off.
4. Send Alerts: The Android application should alert users when the sensor values cross certain pre-set thresholds.

The final vision of the project is to have a comprehensive IoT system that not only collects and displays valuable environmental data but also automatically responds to this data to improve user comfort and efficiency. The Android application connected to this system should be user-friendly, allowing users to monitor their environment and control the devices easily.

Specifically, the application should:

1. Display real-time and historical sensor data in a user-friendly manner.
2. Notify users about significant changes in their environment, such as a decrease in air quality or increase in temperature beyond set limits.
3. Allow users to control the brightness of an LED lamp and the operation of a motor.
4. Adjust the brightness of an LED lamp and activate the motor automatically based on sensor data.

5.2. Software Aspect and Hardware:

5.2.1. Software Aspect:

1. **Python Script on Raspberry Pi**: This script will continuously collect data from the attached sensors and control the connected devices. It will also send this data to the InfluxDB time series database. It should handle exceptions and errors gracefully, and ideally run as a service so it starts up automatically.
2. **InfluxDB Database**: This time series database will store sensor data for analysis and visualization. It's designed for handling time-stamped data like ours, making it a good choice for this project.
3. **Java Android Application**: This app will fetch data from the InfluxDB database and display it in real-time to the user. It will have different screens for each sensor's data and for controlling the LED lamp and motor. It will also send user input data back to the Python script on the Raspberry Pi to control the connected devices.

5.2.2. Hardware Aspect**:**

1. **Raspberry Pi 4 Model B**: This is the microcontroller that will interface with the sensors and devices. It will run the Python script that gathers sensor data and controls the devices.
2. **CCS811 Air Quality Sensor**: This sensor measures eCO2 (equivalent calculated carbon-dioxide) concentration within a range of 400 to 8192 parts per million (ppm), and TVOC (Total Volatile Organic Compounds) concentration within a range of 0 to 1187 ppb (parts per billion).
3. **DHT22/AM2302 Temperature and Humidity Sensor**: This sensor can measure temperatures from -40 to 80 degrees Celsius and humidity from 0 to 100%.
4. **PZEM-004T Smart Energy Meter: This sensor is used for measuring the voltage, current, power, and energy in a circuit. It interfaces with the Raspberry Pi through an RS485 to UART converter.**
5. **TSL2561 Digital Light Sensor**: This sensor measures visible and infrared light to accurately calculate the luminosity (in lux) of the environment.
6. **LED Lamp**: This will be controlled by the GPIO pins of the Raspberry Pi, and its brightness will adjust according to the light sensor's data.
7. **DC Motor**: This will be controlled by the GPIO pins of the Raspberry Pi and will activate when the temperature sensor reads a certain threshold.
8. **Power Supply**: A power supply for the Raspberry Pi and another power supply for the DC motor.
9. **Motor Driver/Relay and Transistor/MOSFET**: These are needed to control the DC motor from the Raspberry Pi's GPIO pins. The exact components depend on the motor's specifications.
10. **RS485 to UART converter**: This is necessary for interfacing the PZEM-004T with the Raspberry Pi.

5.3. Screen Flows:

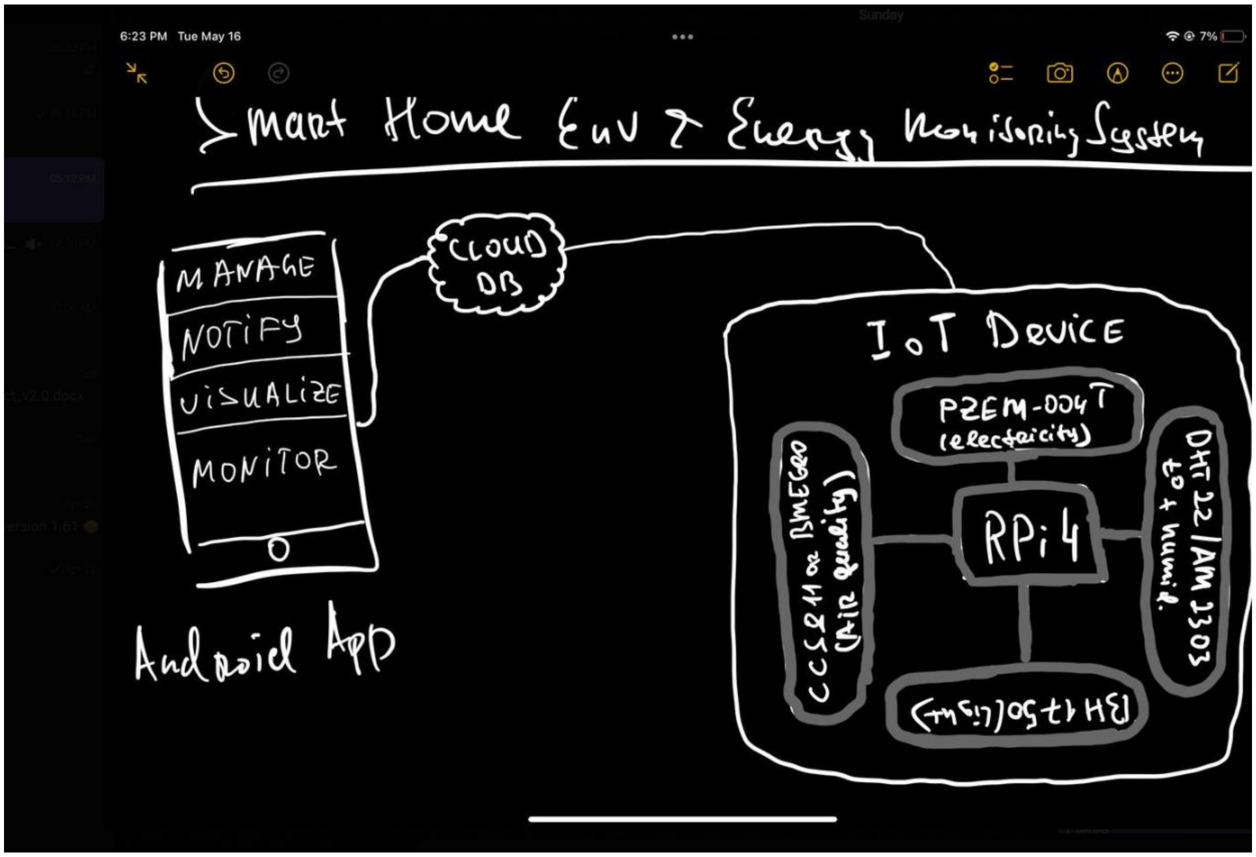
Navigation between these screens will be handled by a navigation drawer.

1. **Splash Screen**: This is the first screen the user will see when they open the app. It displays the app logo and app name, and it stays visible while the app loads.
2. **Home/Dashboard Screen**: After the splash screen, the user will be navigated directly to the home screen. This screen displays an overview of all sensor data and device status. For example, urgent notifications or warnings (like temperature too high, or air quality too poor).
3. **Temperature and Humidity Screen**: Users can navigate to this screen to view more detailed information about the temperature and humidity data, like historical data charts. There will be an option to manually control the motor acting as AC.
4. **Air Quality Screen**: This screen will show the detailed eCO2 and TVOC readings from the air quality sensor.
5. **Energy Usage Screen**: This screen will display detailed energy usage statistics gathered from the PZEM-004T sensor.

6. **Light Sensor Screen**: This screen will show detailed light intensity data. Also, there will be an option for users to manually control the LED lamp's brightness.
7. **Settings Screen**: This screen can be used to adjust application settings or sensor configuration.
8. **About Screen**: This screen basically shows the way of using this app.

5.4. How we incorporated the feedback provided through the interview:

Throughout the interview, our group presented the following diagram and detailed explanation of the main components of our IoT system. Sensor types were scrutinized in terms of their neccessity for the project's purpose. Similar applications that are already available on the market were discussed. The interview confirmed that it is feasible for our group to complete the scope of our project in the expected timeframe.



5.5. The way of planning to satisfy to read/write from the DB is hosted on the cloud:

We will use influxDB, which will follow the following approach:

1. **Database Setup**: We would first need to set up an InfluxDB instance on the cloud. This could be done using cloud providers like AWS, Google Cloud, or Azure, which offer managed database services.
2. **Database Configuration**: After the database is set up, we will need to ensure that it is properly configured to allow connections from our devices. This usually involves setting up security groups or firewall rules to allow incoming connections on the necessary ports.
3. **Sensor Data Collection**: The Python script running on the Raspberry Pi collects data from the sensors. This script uses the InfluxDB Python client library to write data to the database.
4. **Database Reading**: For reading data from the database, the Java Android application will also use an InfluxDB client library (like the one available for Java) to connect to the database and perform queries.

6. Project Scope:

The technical scope of the Android application within our IoT project involves developing an interface for the users to interact with the data generated by the IoT system, and to control certain aspects of the system. Here's a detailed overview of the plan:

1. **Design and Develop User Interface**: We'll begin by designing the screens for the Android application. These screens include a main dashboard for displaying sensor data in real-time, and individual screens for more detailed data related to each sensor. The app will also include controls to manually adjust the LED brightness and activate the motor.
2. **Database Integration**: The next phase involves writing code to fetch the data from the InfluxDB cloud database. This will be done using the InfluxDB Java client and will involve setting up asynchronous tasks to periodically fetch and update data on the app's interface.
3. **Implement Control Features**: The application will also need to send commands back to the Raspberry Pi system to control the LED and motor. This will be achieved by writing to specific locations in the database that the Raspberry Pi will monitor for changes.
4. **Testing**: Each feature of the app will be rigorously tested. This includes checking the real-time update of sensor data, verifying the accuracy of displayed data, and testing the control features to ensure they properly affect the hardware components.

5. **Enhancements and Polishing**: Based on the feedback received during testing, enhancements will be made to the app's user interface and functionality. This may include visual tweaks, performance improvements, and user experience enhancements.

The Android application project will be considered complete when:

1. The application successfully displays real-time data from the InfluxDB cloud database.
2. Users can control the LED brightness and motor activation from within the app, and these changes reflect correctly in the system.
3. The user interface is intuitive, visually pleasing, and responsive.
4. The application does not crash or hang during regular operation, and there are no significant bugs affecting the functionality or user experience.
5. The application has been tested on multiple Android devices, ensuring compatibility with various screen sizes and Android versions.

7. Project Layout

We will use navigation drawer as our project layout.

For navigation drawer, each navigation drawer item can correspond to the different sections of the app - Home, Sensor pages, About, and Setting.

For **implementation**, each of these components will be defined in our layout XML files, and their behavior will be implemented in the corresponding Java classes. In each case, we will set an **OnNavigationItemSelectedListener** to respond to selection events.

8. Project Themes, Epics, User Stories, and Tasks

8.1. Theme 1: App Usability and Aesthetic

8.1.1. Epic 1: Navigation System

- **Story 1: Implement navigation drawer.**
    - Task 1: Design layout with navigation drawer.
    - Task 2: Connect navigation to corresponding pages (Home, Sensor pages, Settings, About).
    - Task 3: Test navigation between pages for a seamless user experience.
- **Story 2: Design and implement Home and About pages.**
    - Task 1: Design the Home and About pages.
    - Task 2: Implement the designs in Java.
    - Task 3: Test the appearance and functionality of these pages.

- **Story 3: Dynamic navigation icons**
    - Task 1: Design different icons for each navigation item.
    - Task 2: Implement dynamic changes to icons based on page focus.
    - Task 3: Test navigation icons for visibility and intuitive use.

8.1.2. Epic 2: User Interface Improvement

- **Story 1: Design appealing sensor pages**
    - Task 1: Brainstorm and sketch design ideas for sensor pages.
    - Task 2: Implement design in Java.
    - Task 3: Test sensor pages for user-friendliness and aesthetic appeal.
- **Story 2: Implement responsive design.**
    - Task 1: Define different layouts for different screen sizes.
    - Task 2: Implement the different layouts in Java.
    - Task 3: Test the responsiveness of the app on various devices.
- **Story 3: Dark mode implementation**
    - Task 1: Design a dark theme for the application.
    - Task 2: Implement the dark theme in the application settings
    - Task 3: Test the application in both light and dark modes to ensure it works properly

8.2. Theme 2: Sensor Monitoring and Interaction

8.2.1. Epic 1: Sensor Data Display

- **Story 1: Display real-time sensor data on sensor pages.**
    - Task 1: Design and implement four sensor pages.
    - Task 2: Establish data communication between the app and sensors.
    - Task 3: Update sensor pages with real-time data.
- **Story 2: Refresh function for sensor data**
    - Task 1: Implement a 'refresh' button on sensor pages.
    - Task 2: Define the function to fetch the latest sensor data on refresh.
    - Task 3: Test the refresh function to ensure data updates correctly.
- **Story 3: Sensor data logging**
    - Task 1: Design a data structure to store historical sensor data.
    - Task 2: Implement data logging in the application backend.

    - Task 3: Display historical sensor data in a user-friendly way on sensor pages.

8.2.2. Epic 2: Interactive Control of Devices Based on Sensor Data

- **Story 1: Control LED light based on light sensor data.**
    - Task 1: Develop a function to analyze light sensor data.
    - Task 2: Implement a button on the light sensor page to adjust the LED brightness.
    - Task 3: Test to ensure LED responds correctly to button and light sensor data.
- **Story 2: Control AC (electric fan) based on temperature and humidity sensor data.**
    - Task 1: Develop a function to analyze temperature and humidity sensor data.
    - Task 2: Implement a button on the temperature & humidity sensor page to manually control the AC.
    - Task 3: Test to ensure AC responds correctly to button and sensor data.
- **Story 3: Customizable settings for automatic device control**
    - Task 1: Implement a settings page where users can set thresholds for device control.
    - Task 2: Use these thresholds in the functions controlling the LED light and AC.
    - Task 3: Test to ensure settings correctly affect device control.