Date: 12/12/2021

University of Birmingham

Building usable software

(Software engineering and professional practice)

Challenge: Living Lab

Project: Smart Energy Management system

Group number: 60

Students' details:

1. ID: 2389182, Email: AXG293@student.bham.ac.uk
2. ID: 1963223, Email: FFK823@student.bham.ac.uk
3. ID: 2378164, Email: CXJ164@student.bham.ac.uk
4. ID: 2248338, Email: NTA038@student.bham.ac.uk
5. ID: 2382544, Email: NXZ144@student.bham.ac.uk
6. ID: 2362320, Email: RXD120@student.bham.ac.uk
7. ID: 2117893, Email: SXA1367@student.bham.ac.uk
8. ID: 2369763, Email: ZXL166@student.bham.ac.uk

Table of Contents

Table of Figures

Part A: Requirement engineering

A1: System Description

A2: Functional and non-functional features

System overview

The smart energy management system will manage energy production, consumption, and storage. It will provide the necessary features to ensure that the campus is aligned with governmental guidelines to achieve net-zero emission by 2050.

The system will monitor the stored energy levels and energy storage conditions to minimise risks and avoid power shortage. It will monitor energy usage and consumption and provide heat maps for visualization. The system will provide a real-time comparison of power consumption between different buildings. The results of these comparisons will be displayed across the campus to educate people about energy.

With the cutting edge IoT sensors implemented across the campus, it will be possible to create a digital twin. The digital twin will help with the research process, anticipate the changes in energy requirements, support early fault detection and simulate possible energy handling scenarios. It will also be used to test modifications, possible updates, and scenarios, and see how they will affect the campus system in real life.

As the data collected by the sensors accumulates in data server. The system will allow researchers from inside and outside the university to access them. This will help advance research with university-specific problems and provide futuristic solutions for them. As access from outside will be required, the system should maintain a certain level of data security to avoid any risk.

Other features like solar panels cleaning, efficiency measurement, and environmental adaption of buildings will be automated using an external AI system that implements sensors to monitor the surrounding environment and other conditions. The platform can be extended to include other features as the system expands.

Finally, the system will be compatible with 2D and 3D interactive display devices. The devices will stay in standby power-saving mode. They will include any required accessibility features and support multiple languages like Arabic and English.

Energy mapping:

Different map types are available using 2D and 3D display devices. The energy monitoring unit will be able to visualize energy consumption and production. Heat maps will show the real-time consumption of each building and allow users to access previous consumption records. It will also show buildings specific systems consumption. It will also map energy production in the university renewable plants to include the real-time production levels, previous energy production records and comparison between different times of the year and for campus plants.

Energy storage management:

Energy storage management unit will monitor energy storage on campus to keep stored energy within the allowed limits. It will consume small amounts of energy surplus using entertainment systems like light shows and music. If dissipation isn’t enough, the unit will transfer the excess into the city grid. The unit will also draw energy from the city grid to cover any shortage in power.

Finally, the unit will compare energy requirements with the predicted production levels to decide if there will be a need to dissipate, transfer or request energy from the city grid.

Environmental adaptation:

The building management system will use sensors and cameras to monitor and collect information about the surrounding environment and will use the AI system to identify room activity and optimise energy usage. It depends on the outside weather, humidity, time of day and number of people in the room to decide room setting. It selects the suitable light source and when to adjust the room temperature. This will enhance the system performance and reduce the overall power consumed by the university.

Early faults detection:

Fault monitoring unit will communicate with the AI system to identify and locate potential faults that might lead to grid failure. It will be able to determine the risk level of each fault and the affected systems. It collects data and monitors the electric grid to provide real-time observation of faults. When a fault is detected, the unit uses AI to determine its type and the time required for maintenance. It will send an immediate alert to the maintenance department with these details. It will also suggest actions to prevent damage until the fault is fixed. Finally, the system will track the maintenance process and contact the head of maintenance if the fault was not fixed in time. In time, the system can provide data about the possibility of each fault and suitable actions to prevent it. It will also send an alert for repetitive faults and unusual energy consumption.

Assumptions:

- This is an independent system that has its own subsystems and components like display devices and monitoring units.
- The system is integrated with other external systems across the campus like the AI system.
- System sensors are always connected and not affected by noise and weather.
- The campus depends on several on-grid solar powered renewable energy plants.
- Different types of 2D and 3D display devices are installed around the campus and integrated with the system.

Functional requirements: (Written in MoSCoW format)

1. The building management system (BMS) must monitor all possible changes inside and outside buildings.

    1.1 The BMS must be able to identify the number of people in the room at any time.
    1.2 The BMS must use the AI system to identify the activity held inside the room at any time.
    1.3 The BMS must be able to monitor the temperature inside and outside rooms.
    1.4 The BMS must be able to monitor humidity inside and outside rooms.
    1.5 The BMS must be able to monitor environmental changes like rain, fog and sandstorms.
    1.6 The BMS should allow authorised users like the building manager to bypass the automated environmental adaption system and determine the room settings manually.
    1.7 The BMS must always keep track of date and time.
    1.8 The BMS should be able to identify the weather condition at any time.

2. The BMS should be able to modify room settings.

    2.1 The BMS should modify the buildings temperature using AI according to the outside weather and the number of people inside.
    2.2 The BMS should modify the lighting of the building using AI according to time, outside weather and the activity inside the classroom.
    2.3 The BMS should be able to open and close the curtains of the classroom.
    2.4 The BMS should be able to open and close the windows of the classroom.
    2.5 The BMS should prevent people in the room from opening windows manually when the AC's is running.

3. The display unit should allow users to view and interact with energy data in different ways.

    3.1 The display unit should allow users to view energy production or energy consumption heat maps.
    3.2 The 2D display units should display data in 2D using charts and graphs.
    3.3 The 3D display units should display data in 3D using virtual reality and holograms.
    3.4 The display unit should allow users to interact with heat maps by zooming at specific building to get the energy consumption details of each exact systems of each building.
    3.5 The display unit should allow users to interact with heat maps by zooming at specific renewable energy plants to get their energy production levels and the production efficiency records for each panel.

4. The energy monitoring unit (EMU) must allow users to look up the required energy production and consumption data and access previous records.

    4.1 The EMU must allow users to look up the real-time energy consumption levels for campus buildings.
    4.2 The EMU must allow users to look up real-time energy production levels of campus renewable energy plants.
    4.3 The EMU should allow users to access previous energy consumption levels for campus buildings in a daily, monthly and yearly using data server.
    4.4 The EMU should allow users to look up the previous energy production records for campus renewable energy plants in a daily, monthly and yearly form using data server.
    4.5 The EMU could allow users to look up the production efficiency of each panel in the solar energy plants.
    4.6 The EMU must track energy production from each plant in terms of kWh.

4.7 The EMU must track energy consumption of each building as a whole and the consumption of individual systems of the building in terms of Wh.
4.8 The EMU should be able to compare the energy consumption for the same building across different times and among other buildings.
4.9 The EMU must allow the energy management unit supervisor to view the energy records of the system.

5. The fault monitoring unit (FMU) must detect faults, determine their affects, decide what to do and contact required users.

5.1 The FMU must track any changes in the electric grid that may indicate a possible fault like changes in frequency and current.
5.2 The FMU should categories the risk of each fault into low, medium or high risk.
5.3 The FMU could alert the maintenance team if a certain fault is repeated, how many times and in which parts.
5.4 The FMU should suggest possible actions to apply to prevent any further damage until the fault is resolved.
5.5 The FMU should use AI to set the time required to manage the fault.
5.6 The FMU could use AI to determine the systems affected by the fault.
5.7 The FMU should follow up with the maintenance team and alert management if the fault was not resolved in time.
5.8 The FMU could send a confirmation message to the management when the fault is resolved.
5.9 The FMU could send an alert for the energy manager when an unusual consumption is detected in any building.

6. The energy storage management unit (ESMU) must handle the storage, dissipation, transferring and drawing of energy.

6.1 The ESMU should be able to dissipate part of the excess energy produced through entertainment systems like lights and music.
6.2 The ESMU must be able to transfer excess energy into the city grid when energy production levels exceed storage abilities.
6.3 The ESMU must be able to draw energy from the city grid, to maintain services when the produced energy is insufficient to run the campus.
6.4 The ESMU should be able to predict the energy amount required for each building using AI.
6.5 The ESMU could compare the required energy with the produced amounts to minimise energy drawing from the city grid.

7. The system must be able to record, present and use the data collected from the entire campus systems using the data server.

7.1 The system must store the data collected by the BMS and make it accessible for users and researchers using data server.
7.2 The system must store the data collected by the energy monitoring unit and make it accessible for users and researchers using data server.
7.3 The system should Allow the AI system to access the data collected to create a digital twin.

8. The panels monitoring unit could automatically clean the solar panels when their efficiency drops because of dirt.

9. Panels monitoring unit could use AI system to determine if solar panels are covered with dirt using thermal imaging.

10. The system’s digital twin should be able to simulate the real-time processes of the campus.

Non-Functional Requirements: (Written in MoSCoW format)

Availability:

1. The system must be able to function 24/7 with up to 99.5% responsiveness.
2. Communication between both campuses should be available 24/7 with a delay less than 5 seconds.
3. The system should have an automatic weekly time for maintenance and updates for less than 45 minutes scheduled every Friday between 2 Am and 3 AM.
4. The FMU must monitor the electrical grid 24/7 for possible faults.
5. The BMS should have access to time, date and weather condition 24/7.
6. The local data base of each building should be able save the BMS data 24/7.

Reliability:

1. The EMU should handle the campus energy production and consumption data and generate the requested maps and graphs within 1 second.
2. Energy consumption maps and graphs must include campus ID, building ID and the amount of consumption for the required time.
3. Energy production maps and graphs must include campus ID, renewable power plant ID and the amount of production for the required time.
4. The FMU should make the alert within 5 seconds of fault detection.
5. An alternative fault monitoring system could be online within 1 hour, in case of a system breakdown.
6. The energy management system should be able to restart in less than 120 seconds.
7. The data server should generate a report with the data required by the user within 5 seconds from the request time.
8. The system must be able to process user requests within 5 seconds.

Accessibility and compatibility:

1. The system should make the data accessible by researchers from inside and outside the campus.
2. The system should be compatible with different devices like PC’s and mobile phones.
3. The systems’ user interface should have accessibility features such as text to speech and high contrast options.
4. The user interface of the display devices must be compatible with multiple languages like English and Arabic.
5. The energy mapping system should be accessible on users’ smart phones and devices using campus application.

Security:

1. The system must maintain an end-to-end encryption for the campus data.
2. The system must prevent any unauthorised users from accessing the data.
3. The system firewall must prevent any suspicious activity like rerouted connections and DDOS attacks.
4. The system should require users to provide an ID and password to access data.
5. The communication between the systems of both campuses should be protected with end-to-end encryption.

Modifiability:

1. The system should accept adding plugins and APIs to extend new functions when needed.
2. The BMS should be able to integrate new types of sensors and updates.
3. The system must allow annual upgrading of its internal software.
4. Any modification applied to one campus should be applicable at the second one.

Scalability:

1. The system should be able to connect with up to 2000 user devices concurrently.
2. The system should allow data access to up to 100 researchers concurrently.
3. The BMS would be able to monitor up to 25000 sensors concurrently.

Safety:

1. The system must have a backup data server service to prevent failure.
2. The system should have an alternative fault monitoring system in case of failure.

Privacy:

1. All data related to activities in the rooms like lectures and exams are considered as private and must not be shared by any party, including researchers.
2. All data related to people on campus like their photos, usernames, emails and activities are considered as private and must not be shared by any party, including researchers.

External requirements:

1. The EMS should be aligned with DEWA and UK regulations and IEC 61000-6 regarding electricity transferring depending on campus location.
2. The system sensors must be aligned with SEIMENS QFM81 series standards.
3. The system must be aligned with UAE and UK health and safety standards depending on campus location.
4. The structure of the renewable energy plants frame must be aligned with IeC/TS 62548 standards.
5. The renewable energy system electrical connections must be aligned with IEC 60364 standards.

Part B: Software design with UML

B1: Use case diagram

B2: Documentation of two use cases

B3: Description of two scenarios for two use cases

B4: Activity diagram

B5: Class analysis:

i. Noun-verb analysis

ii. CRC cards

iii. First-cut class diagram

iv. Detailed class diagram

B6: Object diagram

B7: Two sequence diagrams

B8: Two state machine diagrams

B1: Use case diagram*

The following diagram represents the use case diagram of the smart energy management system. As the use case diagram reflects a simplified version of the system with the main functionalities, some of the relations in the system are not included in this diagram.

System actors and their tasks:

- Building manager:
  The person in charge of monitoring the BMS activity. He can bypass the BMS and set rooms settings manually when needed.

- Renewable energy manager:
  The person in charge of the renewable energy on the campus. He monitors the solar panels, their efficiency and renewable energy production.

- Head of maintenance:
  The person in charge of maintenance. He receives all the system alerts and monitors the early fault detection system.

- Head of energy department:
  The person in charge of the energy storage management unit and the energy monitoring unit.

- Data server:
  An external cloud server that stores the data of the energy management system.

- AI system:
  An external system that communicates with different parts of the system to improve the decision-making process.

- City grid:
  An external energy grid that transfers energy from and to the campus as needed.

- User:
  A General class that includes all the general system users like students, lecturers and campus visitors.

- System manager:
  A type of user that can access the campus data to help with monitoring and to solve any management issue.

- Researchers:
  A type of user that can access the campus data to help with the research process.

*A PDF version of the use case diagram can be found in the supporting documents folder in a higher resolution.

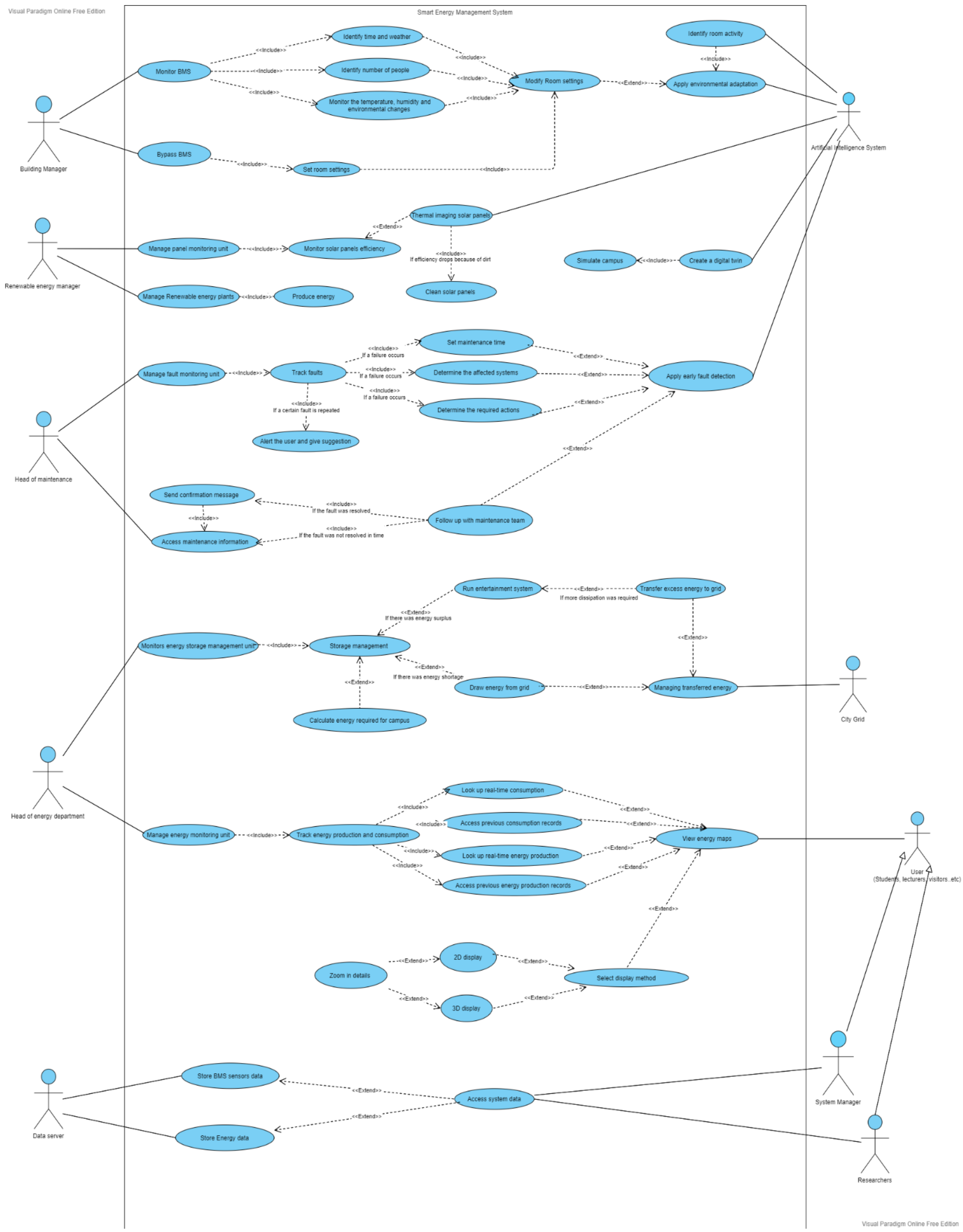

Figure 1: Use case diagram

B2: Documentation of a use case (1)

Name: Apply environmental adaption

Description: Automating the process of changing room settings using the AI system and the BMS.

Actors:

- Artificial intelligence (AI) system
- Building manager

Use-case relationships:
- Applying environmental adaption by modifying the room settings.
- Appling environmental adaption depending on the activity inside the room.

Pre-conditions:
- The system knows the time of the day.

Basic flow:
1. The BMS sensors measure the current temperature, humidity, weather and other environmental changes.
2. The BMS uses room cameras to determine the number of people in the room.
3. The BMS sends the readings of the sensors, current weather and the number of people in the room to the AI system.
4. The BMS connects to the AI system to determine the activity inside the room, which will affect the decision making.
5. The BMS connects to the AI system to identify the most suitable temperature inside the room according to the sensors readings and the outside weather.
6. The BMS connects to the AI system to identify the most suitable light level according to the activity in the room, time of the day and outside weather.
7. The AI system sends back the new room settings to the BMS to apply them.
8. The BMS applies the new room settings by modifying light levels and room temperature.
9. The BMS keeps monitoring for new changes in temperature, humidity and weather.

Post-conditions:
- The room temperature has been modified to suit the outside weather and the number of people inside the room.
- The room lighting has been modified to suit the current activity in the room, time of the day and outside weather.
- After modifying room settings, the BMS keeps monitoring changes in temperature, humidity and weather.

What could go wrong:
- The sensors may get damaged, and the data about the environment will not be accurate anymore.
- The AI system fails to connect and cannot decide the new room settings.

Alternative flow (1):
1. The BMS detects a failed sensor.
2. The BMS decides that the data received by the rest of the sensors are accurate enough to apply environmental adaption.
3. The rest of the sensors measure the temperature, humidity, weather and other environmental changes.

4. The BMS uses room cameras to determine the number of people in the room.
5. The BMS sends the readings of the sensors and the number of people in the room to the AI system.
6. The AI system determines the activity inside the room.
7. The AI system determines the most suitable temperature inside the room according to the sensors readings and the outside weather.
8. The AI system determines the most suitable light level according to the activity in the room, time of the day and outside weather.
9. The AI system sends the new room settings to the BMS to apply them.
10. The BMS applies the new room settings by modifying light levels and room temperature.

Alternative flow (2):

1. The BMS detects several failed sensors.
2. The BMS decides that the data received by the rest of the sensors are not accurate enough to apply environmental adaption.
3. The BMS alerts the building manager of the fault.
4. The building manager logs on to the system.
5. The building manager bypasses the BMS automated room settings.
6. The building manager determines the desired room settings.
7. The BMS modifies the room settings according to the system manager input.
8. The BMS keeps the settings as manual until the failed sensors are replaced.

Alternative flow (3):

1. The BMS sensors measure the temperature, humidity, weather and other environmental changes.
2. The BMS uses room cameras to determine the number of people in the room.
3. The BMS sends the readings of the sensors and the number of people in the room to the AI system.
4. The AI system does not respond to the BMS.
5. The BMS alerts the building manager.
6. The building manager bypasses the BMS automated room settings.
7. The building manager determines the desired room settings.
8. The BMS modifies the room settings according to the system manager input.
9. The BMS keeps the settings as manual until the connection to the AI system is restored.

Note:

- All the data collected by the BMS sensors are stored in the data server.

B2: Documentation of a use case (2)

Name: View energy maps

Description: Creating and viewing heat maps for energy production and consumption on campus.

Actors:
- Head of the energy department
- Users (Students, Lecturers, researchers, system admin…etc)

Use-case relationships:
- The display unit displays maps of real-time energy consumption.
- The display unit displays maps of real-time energy production.
- The display unit displays maps for previous energy consumption.
- The display unit displays maps for previous energy production.
- The display device can be a 2D or 3D device with a suitable data form.

Pre-conditions:
- The display device is in a standby power-saving mode.
- The user selects a 2D or 3D display device.

Basic flow:
1. The user interacts with the display device and selects the language.
2. The user selects either to view a heat map for energy production or a heat map for energy consumption on campus.
3. For the selected type, the user selects to view a real-time map or a map for the previous records.
4. If the user selected a map for previous records, the user should select the length of time for the map (day, week, month, year).
5. If the user selected a map for previous records, the system would access the data server and bring the energy production or consumption data for the required time.
6. If the user selected the real-time map, the energy monitoring unit will provide the consumption/production data directly as a live feed.
7. The data server/energy monitoring unit sends the required data in the suitable form (for 2D or 3D display according to the display unit type) to the display device in use.
8. The display unit displays a map with the required data for the entire campus.
9. The user might interact with the map to view the details of consumption/production of a certain building/power plant (according to the map type).
10. If the user interacts with the map, the map zooms on the required building/power plant, view its energy details for specific building systems/powerplant panels and its ID details.
11. If the user has ended the display process, the display unit will return to standby power-saving mode.
12. If the user doesn't end the display process, the display device will wait 10 minutes for interaction before going back to the standby power-saving mode.

Note:
The live data feed will be stored in the data server to access it later, as a previous record.

Post-conditions:
- The display unit displayed the heat map of the required system with all of its details.
- The display unit allowed the user to interact with the map to view specific details of a building/powerplant.
- All maps included the IDs of the campus, building and system.

What could go wrong:

- The display unit fails and becomes unresponsive.
- The data server fails to connect.

Alternative flow (1):

1. The display unit does not respond to the user command.
2. The user uses his smartphone to access the campus heat map application.
3. The user interface of the app asks the user to determine the type of map required.
4. The app connects to the data server and the energy monitoring unit.
5. The app requests the required data (live feed or previous records).
6. The data server/energy monitoring unit sends the required data to the app as 2D data.
7. The app displays the required map.
8. The user might interact with the app to select certain systems/power plants to zoom in.
9. The user exits the app.

Alternative flow (2):

1. The user selects either to view a heat map for energy production or a heat map for energy consumption on campus.
2. The data server fails, and a note is displayed on the display device that only real-time data is available, and the previous records access is suspended.
3. The user selects to view a real-time map for energy production or consumption.
4. The energy monitoring unit will provide the consumption/production data directly as a live feed.
5. The energy monitoring unit sends the required data in the suitable form (for 2D or 3D display according to the display unit type) to the display device in use.
6. The display unit displays a map with the required data for the entire campus.
7. The user might interact with the map to view the details of consumption/production of a certain building/power plant (according to the map type).
8. If the user interacts with the map, the map zooms on the required building/power plant, view its energy details for specific building systems/powerplant panels and its ID details.
9. If the user ends the display process, the display unit will return to standby power-saving mode.
10. If the user doesn’t end the display process, the display device will wait 10 minutes for interaction before going back to the standby power-saving mode.

B3: Scenario's description

Use case 1: Apply environmental adaption

Scenario 1 (Normal flow):

At 9:00 Am, the BMS decides to modify the settings of room 233 to suit the ongoing software engineering lecture. The BMS uses its sensors and finds that the temperature is 28˚C, humidity 40%, and the light level is 65% inside and 80% outside the room. It also finds that the current weather is clear and sunny without any unusual environmental conditions. It also uses room cameras to calculate that there are 20 people in the room. After that, the BMS sends the collected data to the AI system. The AI system identifies the current room activity as watching a slideshow. The AI system uses the data received from the BMS with the current activity in the room to decide the suitable temperature and light level for the room. It decides to set the temperature to 24˚C, close the curtains and turn on the lights with 30% light power. The AI system sends these settings to the BMS to apply them. The BMS applies the settings and keeps monitoring the room and environment for any changes.

Scenario 2 (Damaged sensors):

At 5:00 PM, in room 120, a software engineering exam begins. The BMS decides to modify the room settings to suit the exam. It checks the room sensors and detects that 3 of the sensors in the room are damaged. The BMS decided that the data collected from the rest of the sensors aren't accurate to set the new room settings. The BMS sends an alert to the building manager. The building manager logs into the system and bypasses the BMS automated environmental adaption. He sets the room temperature to 26, close the curtains and turns on the light with 80% light power. The BMS implements the system manager room settings and keeps the room settings manual until the sensors are replaced.

Use case 2: View energy maps

Scenario 1 (Normal flow):

Ahmad, a student in the university, goes to a 3D virtual reality device to view a heat map for the campus. He activates the display device to exit standby mode. He selects English as his preferred language and selects to view a real-time energy consumption map for the campus. The display unit connects with the energy monitoring unit that starts a live feed of data that includes the real-time campus energy consumption for the display unit to view (the energy monitoring unit sends the required data in 3D form). Ahmad interacts with the device to select the food-court building on the map. The map zooms in on the building and displays its ID details, including campus ID (D 002) and building ID (FC 003). The map also displays the current consumption of energy by the entire building as 100 KW/h, and it shows a list for its internal systems with their consumption details (example: Fire alarm system, system ID: D02.FC.FA.01 consumption: 10 KW/h). Ahmad exits the map, and the display unit returns to the standby power-saving mode.

Scenario 2 (Data server fails to connect):

Samira, a student in the university, goes to a 2D interactive display screen to view a heat map for the campus. She activates the display device to exit standby mode. She selects Arabic as her preferred language and selects to view a map for previous energy production levels on campus. A message is displayed saying that previous data records are currently unavailable and only real-time data is available. Samira selects to view real-time energy production instead. The display unit connects with the energy monitoring unit that starts a live feed of data that includes the real-time campus energy production for the display unit to view (the energy monitoring unit sends the required data in 2D form). Samira interacts with the screen to select renewable energy power plant number 3 on the map, which in turn zooms in and displays its ID details, including the campus ID (D 002) and the power plant ID (REPP 03). The map also displays the amount of energy that is produced by the whole plant as 5 KW/h and demonstrates a list of the power plant details, including the total number of solar panels (950), the current efficiency of the plant panels (15%). Samira exits the map, and the display unit returns to standby power-saving mode

B4: Activity diagram*

A diagram that illustrates the workflow of the activities in the system during a certain scenario. The diagram is based on the scenario explained in B2. Use case diagram 2: View energy maps, second scenario (data server fails to connect). The scenario is found above.

*A PDF version of the Activity diagram can be found in the supporting documents folder in a higher resolution.

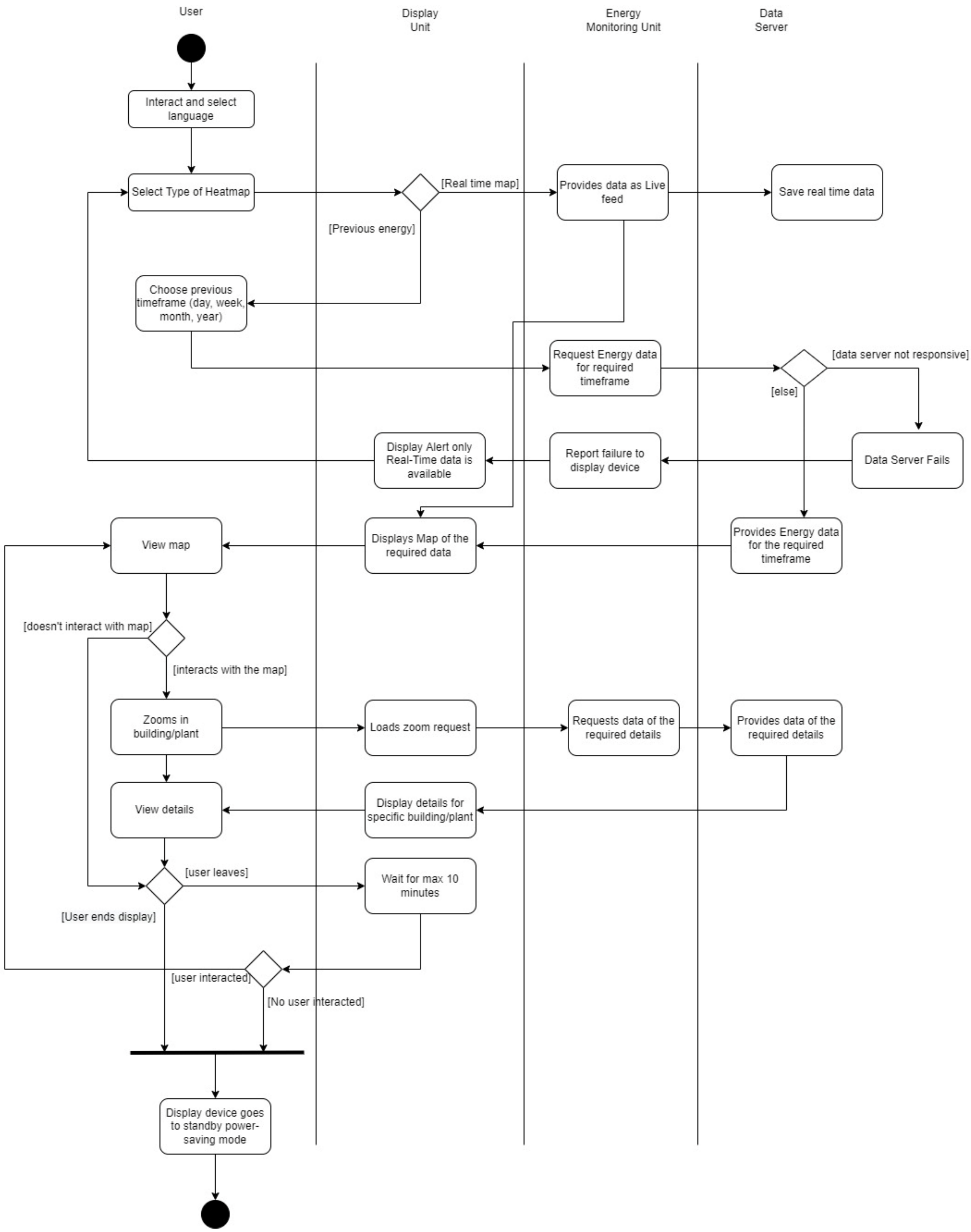

Figure 2: Activity diagram

B5: Class analysis

We started by going through the entire system description to apply the noun-verb analysis, where the **bold** text represents a noun/class, and the *italic* text represents a verb/method. The text is rewritten here with modified style for the words to further explain, the analysis is explained after.

Overview:

The smart **energy management system** will *manage **energy** production, consumption,* and **storage**. It will provide the necessary features to ensure that the **campus** is aligned with governmental guidelines to achieve net-zero emission by 2050.

The **system** will *monitor the stored energy levels and energy storage conditions* to minimise risks and avoid power shortage. It will monitor **energy usage and consumption** and ***provide*** **heat maps** for **visualization**. The **system** will *provide a real-time comparison of power consumption* between different **buildings**. The results of these comparisons will be displayed across the **campus** to educate **people** about energy.

With the cutting edge IoT **sensors** implemented across the **campus**, it will be possible to create a **digital twin**. The **digital twin** will help with the research process*, anticipate the changes in energy requirements*, support early fault detection and *simulate possible energy handling scenarios*. It will also be used to test modifications, possible updates, and scenarios, and see how they will affect the **campus** system in real life.

As the **data** collected by the **sensors** accumulates in **data server**. The **system** will allow **researchers** from inside and outside the **university** to access them. This will help advance research with university-specific problems and provide futuristic solutions for them. As access from outside will be required, the system should maintain a certain level of **data** security to avoid any risk.

Other features like **solar panels** *cleaning*, *efficiency measurement*, and **environmental adaption** of **buildings** will be automated using an external **AI system** that *implements sensors* to *monitor the* **surrounding environment** and **other conditions**. The **platform** can be extended to include other features as the **system** expands.

Finally, the **system** will be compatible with **2D** and **3D** interactive **display devices**. The devices will stay in *standby power-saving mode*. They will include any required accessibility features and *support* **multiple languages** like **Arabic** and **English**.

Energy mapping:

Different **map types** are available using **2D** and **3D display devices**. The **energy monitoring unit** will be able to *visualize* **energy consumption** and **production**. **Heat maps** *will show* the **real-time consumption** of each **building** and allow **users** to *access* **previous consumption records**. It will also show **buildings specific systems consumption**. It will also *map energy production* in the **university renewable plants** to *include* the **real-time production** levels, **previous energy production records** and *comparison* between different times of the year and for **campus plants**.

Energy storage management

Energy storage management unit will **monitor energy storage** on campus to keep **stored energy** within the **allowed limits.** It will consume **small amounts of energy** surplus using **entertainment systems** like **light shows and music**. If **dissipation** isn't enough**, the unit** will *transfer the excess* into the **city grid**. The **unit** will also *draw energy* from the **city grid** to cover any shortage in power.

Finally, the **unit** will *compare energy requirements* with the *predicted production levels* to decide if there will be a need to dissipate, transfer or request energy from the **city grid**.

<u>Environmental adaptation:</u>

The **building management system** will use **sensors** and **cameras** to *monitor* and collect information about the surrounding environment and will use the **AI system** *to identify room activity* and optimise energy usage. It depends on the outside weather, **humidity**, **time of day** and *number of people* in the room to *decide room setting*. *It selects the suitable light source* and when to *adjust the room temperature*. This will enhance the system performance and reduce the overall power consumed by the university.

<u>Early faults detection:</u>

Fault monitoring unit will communicate with the **AI system** to *identify and locate potential faults* that might lead to **grid** failure. It will be able to *determine the risk level of each fault* and the affected systems. It collects **data** and monitors the **electric grid** to provide real-time observation of faults. When a fault is detected, the **unit** uses *AI to determine its type* and the time required for maintenance. It will *send an immediate alert* to the **maintenance department** with these details. It will also *suggest actions to prevent damage* until the fault is fixed. Finally, the **system** *will track the maintenance process* and contact the head of maintenance if the fault was not fixed in time. In time, the system can *provide data about the possibility of each fault* and *suitable actions to prevent it*. It will also *send an alert for repetitive faults* and *unusual energy consumption*.

B5.i: Noun-Verb Analysis:

A Noun-verb analysis of the initial specification was conducted below with the aim of finding classes and the appropriate methods for the described system. All initial possibilities of classes and methods were then put through a round of deliberation following which they were either accepted or rejected for integration into the system. The nouns and verbs initially selected as classes and their subsequent methods are presented in a table which details whether they have been accepted as a class or a method. A short justification for their selection or cancellation is also provided.

Word/Phrase	**Accepted**	**Reason**
Energy management system	No	Refers to the whole System and hence is too general
Energy	Energy	Refers to the energy in the building hence should be a class
Manage energy production	*monitorStoredEnergy*	Refers to an action of measuring energy consumption of an object and hence should be a method
Manage energy consumption	*monitorEnergyConsumption*	Refers to an action of measuring energy consumption of an object and hence should be a method
Storage	**EnergyStorageManagement**	Refers to all the attributes and functions regarding energy storage and hence should be a class
Campus	No	Irrelevant to the system
System	No	Means the same as energy management system and is too general
monitor the stored energy levels	*monitorStoredEnergy*	Duplicate

storage conditions	*monitorStorageCondition*	Refers to an action of EnergyMonitor and hence should be a method
Energy Usage and Consumption	**EnergyMonitor**	Refers to energy monitoring functions and hence should be a class
Heat maps	**HeatMap**	Refers to all the attributes and functions regarding a heat map and hence should be a class
provide heatmaps	*provideEnergyConsumptionMap*	Refers to the act of displaying energy consumption heatmap, hence a method
provide heatmaps	*provideEnergyProductionMap*	Refers to the act of displaying energy production heatmap, hence a method
Visualisation	**ZoomingOnBuilding**	It provides the functionality of detailing all energy attributes of a specific building and hence is a class
System	No	Duplicate
provide a real-time	*showLiveData*	Refers to act of showing real time data and hence should be a method
Comparison of power consumption	*comparePowerConsumption*	Refers to an action of EnergyMonitor and hence should be a method
Buildings	**BuildingInfo**	Building Information should be a class
Campus	No	Duplicate
People	No	Refers to actors of the system

Iot Sensors	**Sensors**	This is a particular component of the system that acquire environmental data and hence should be a class.
Campus	No	Duplicate
Digital Twin	**DigitalTwin**	Refers to all attributes and functions of a digital twin and hence should be a class
anticipate the changes in energy requirements	*calculateEnergyRequirement*	Refers to the calculation of energy requirement using the digital twin, hence a method
simulate possible energy handling scenarios	*simulateOptimalEnergy*	Refers to the calculation of optimal energy for current condition using digital twin, hence a method
Campus	No	Duplicate
data	No	Too general
sensors	**Sensors**	Duplicate
Data server	**DataServer**	Refers to an interface to an external data server.
System	No	Duplicate
researchers	No	Refers to actors
university	No	Irrelevant to the system
system	No	Duplicate
data	No	Duplicate

solar panels	**SolarPanel**	Refers to attributes and function of solar panels, hence a class
solar panels cleaning	*cleanPanels*	Refers to the function cleaning solar panels, hence a method of SolarPanel class
Efficiency measurement	*getSolarPanelEfficiency*	Refers to the action of getting the efficiency of a solar panel and hence should be a method
environmental adaption	**AISystem**	Refers to an external component that decide optimal use of energy with data provided, hence an interface
buildings	**BuildingInfo**	Duplicate
AI System	**AISystem**	Duplicate
Implements sensors	*checkStatus*	Refers to the action of checking the health condition of a sensor, hence should be a method
monitor	*monitorEnvironment*	Refers to the function of monitoring environment by the AISystem interface, hence a method
Surrounding environment	**EnvironmentalConditions**	Refers to the definition of the current environmental condition, hence a class
and other conditions	*checkEnvironmentalConditions*	Refers to the function of checking the environmental conditions and hence should be a method
platform	No	Too general
system	No	Duplicate
system	No	Duplicate

2D	**TwoDDisplay**	This is a particular action of viewing HeatMap, it should provide the information about the screen and hence becomes a class
3D	**ThreeDDisplay**	This is a particular action of viewing map, it should provide the information about the device and hence becomes a class
display devices	**DisplayDevice**	Refers to the devices used to visualise heat mapping data, hence a class
standby power-saving mode	*returnsToStandByMode*	This is a particular action of 2D and 3D devices and hence should be a method
support	*setLanguage*	This is the main action of setting display language, hence should be a method
multiple languages	*SystemLanguages*	Refers to all the languages the system is usable in, hence a class.
English	No	Refers to a particular supported language
Arabic	No	Refers to a particular supported language
Map types	**MapType**	Refers to selection of available map types, hence a class
2D	**TwoDDisplay**	Duplicate
3D	**ThreeDDisplay**	Duplicate
display devices	**DisplayDevice**	Duplicate
energy monitoring unit	**EnergyMonitor**	Duplicate

visualize	*showZoomDetails*	This is a particular action of ZoomingOnBuilding hence should be a method
energy consumption	**EnergyConsumptionRecords**	Responsible for collating and storing all data regarding campus energy consumption, hence a class
energy production	**EnergyProductionRecords**	Responsible for collating and storing all data regarding campus energy production, hence a class
Heat maps	**HeatMap**	Duplicate
will show	*viewHeatMap*	Refers to action of showing the heat map hence should be a method
real-time consumption	**RealTimeConsumptionMap**	This should be a subclass of the HeatMap class
building	**BuildingInfo**	Duplicate
Users	No	Refers to actors of the system
access	*provideEnergyConsumptionMap*	Duplicate
previous consumption records	**PreviousConsumptionMap**	This should be a subclass of the HeatMap class
buildings	**BuildingInfo**	Duplicate
specific systems consumption	**EnergyConsumptionRecords**	Many detailed information should be record here, so it should be a subclass of HeatMap
map energy production	*provideEnergyProductionMap*	Duplicate
University	No	Duplicate

renewable plants	**PowerPlantDetails**	Provides information about energy production plants, hence should be a class
include	*selectTimeframe*	Refers to an action of including data of specific time hence should be a method
real-time production	**RealTimeProductionMap**	This should be a subclass of the HeatMap class
previous energy production	**PreviousProductionMap**	This should be a subclass of the HeatMap class
records	**EnergyProductionRecords**	Duplicate
comparison	*comparePowerConsumption*	Duplicate
Campus	No	Duplicate
Plants	**PowerPlantDetails**	Duplicate
Energy storage management unit	No	This is the subsystem's name, cannot to be a class
monitor energy storage	**EnergyStorageManagement**	Duplicate
campus	No	Duplicate
Stored Energy	**EnergyStorageManagement**	Duplicate
allowed limits	**SettingLimit**	This is a particular action of keeping stored energy within the allowed limits and hence becomes a class.
small amounts of energy	No	Too general

entertainment system	No	Too general
light shows and music	No	Too general
dissipation	No	Too general
the unit	No	Too general
transfer the excess	*transferExcessEnergy*	This is a particular action of EnergyStorageManagement class and hence becomes a method.
the city grid	**CityGrid**	This should be an interface that contains a set of operations about energy
the unit	No	Duplicate
draw energy	*drawRequiredEnergy*	This is a particular action of EnergyStorageManagement class and hence becomes a method.
the city grid	**CityGrid**	Duplicate
the unit	No	Duplicate
compare energy requirements	*compareEnergyRequired*	This is a particular action of EnergyStorageManagement and hence becomes a method.
predicted production levels	*predictProductionLevel*	This is a particular action of EnergyStorageManagement and hence becomes a method.
the city grid	**CityGrid**	Duplicate
the building management system	**BmsAction**	This is a particular component of the system that acquire environment data and transfer the data to the AI system, hence a class.

Sensors	**Sensors**	Duplicate
cameras	**RoomCamera**	This is a particular component of the system that acquire environment data and hence should have a class of its own
To monitor	*monitorEnvironment*	Refers to action of monitoring the environment and hence should be a method
AI system	**AiSystem**	Duplicate
identify room acitivity	*identifyRoomActivity*	Refers to action of identify room activity to suggest settings to minimise energy and hence should be a method
Optimize energy usage	*setSettings*	Refers to action of appliying the recommended settings of the AISystem and hence should be a method
weather	**Temperature**	This is a particular component of the system that supply the original data, hence a class
humidity	**Humidity**	This is a particular component of the system that supply the original data, hence a class
time of day	**Lightlevel**	This is a particular component of the system that supply the original data, hence a class
Number of people	*identifyNumberOfPeople*	Refers to action of collecting information about number of people and hence should be a method
decide room setting.	*applySettings*	Refers to action of applying the settings recommended by the AISystem, and hence should be a method

selects the suitable light source.	**NewSettings**	This carries out environmental adaption to optimise energy and hence becomes a class
adjust the room temperature	**NewSettings**	Duplicate
Fault monitoring unit	**SystemFaults**	Contains the attributes of all possible faults and necessary methods to detect them, hence a class
AI	**AiSystem**	Duplicate
Identify and locate potential faults	*detectFault*	Refers to continuous scanning and detection of a fault, hence a method
Grid	No	Too General
Determine the risk level of each fault	*detectFaultLevel*	Refers to grading a detected fault, hence a method
Data	No	Duplicate
Electric grid	No	Irrelevant
Unit	No	Duplicate
Uses AI to determine its type	*identifyFaultType*	Refers to identifying type of fault, hence a method
Send an immediate alert	*generateAlert*	Refers to creating an alert when a fault is detected, hence a method
Maintenance department	No	Is an actor
Suggest actions to prevent damage	*suggestActions*	Refers to the activity of generating damage mitigation methods for detected fault, hence a method.

System	No	Too General
Will track the maintenance process	*trackRepair*	Refers to the activity of monitoring status and generating alerts after a set time, hence a method
contact the head of maintenance	*generateAlert*	Duplicate
Provide data about the possibility of each fault	*predictFault*	Refers to the activity of predicting faults, hence a method
suitable actions to prevent it	*suggestActions*	Duplicate
send an alert for repetitive faults	*trackRepetitiveFaults*	Refers to the activity of tracking repetition of system faults.
send an alert for unusual energy consumption	*generateAlert*	Duplicate

B5.ii: Responsibility Driven Analysis (CRC cards):

The previous process of noun-verb analysis was conducted to discern the base classes and methods that would play a role during the development of the system. To further the understanding of the specific classes and their connections, the noun-verb analysis was paired with the below detailed responsibility-driven analysis of the system.

The results of these two methods with be imperative in the creation of a first cut class diagram. Which is used in the creation of a full class diagram, detailing the methods and attributes of each class present in the system.

<u>Note:</u>
The ones starting with capital letter represents a class, and the ones starting with a small letter represents a method.

Energy	
Responsibilities	**Collaborators**
The responsibility of this class is to represent the energy consumption/production in the building.	Sensors

EnergyMonitor	
Responsibilities	**Collaborators**
The responsibility of this class is to monitor energy storage.	HeatMap DataServer PowerPlantDetails

ThreeDDisplay	
Responsibilities	**Collaborators**
The responsibilities of this class are to provide information about the VR device.	MapType

SolarPanel	
Responsibilities	**Collaborators**
The responsibilities of this class are to define a solar panel and carry out maintenance operations.	PowerPlantDetails

DigitalTwin	
Responsibilities	**Collaborators**
The responsibility of this class is to simulate an optimal campus scenario using the external AiSystem.	AiSystem

MapType	
Responsibilities	**Collaborators**
The responsibilities of this class are to validate users to choose a map type.	ThreeDDisplay TwoDDisplay DisplayDevice

SystemLanguages	
Responsibilities	**Collaborators**
The responsibilities of this class are to validate users to choose suitable language for the mapping system.	DisplayDevice

RealTimeConsumptionMap	
Responsibilities	**Collaborators**
The responsibilities of this class are to process the real time data transmitted from EnergyConsumptionRecords for display.	EnergyConsumptionRecords HeatMap

PreviousConsumptionMap	
Responsibilities	**Collaborators**
The responsibilities of this class are to process the old data transmitted from EnergyConsumptionRecords for display.	HeatMap EnergyConsumptionRecords

RealTimeProductionMap	
Responsibilities	**Collaborators**
The responsibilities of this class are to process the real time data transmitted from EnergyProductionRecords for display.	HeatMap EnergyProductionRecords

PreviousProductionMap	
Responsibilities	**Collaborators**
The responsibilities of this class are to process the old data transmitted from EnergyProductionRecords for display.	HeatMap EnergyProductionRecords

HeatMap	
Responsibilities	**Collaborators**
The responsibilities of this class are to provide consumption and production as a heatmap.	PreviousProductionMap RealTimeConsumptionMap RealTimeProductionMap PreviousConsumptionMap EnergyMonitor DisplayDevice

BuildingInfo	
Responsibilities	**Collaborators**
The responsibilities of this class are to record each building's information.	ZoomingOnBuilding

ZoomingOnBuilding	
Responsibilities	**Collaborators**
The responsibilities of this class are to zoom in on chosen building and display extended details.	BuildingInfo DisplayDevice

EnergyConsumptionRecords	
Responsibilities	**Collaborators**
The responsibilities of this class are to provide records of energy consumption records.	RealTimeConsumptionMap DataServer PreviousConsumptionMap

TwoDDisplay	
Responsibilities	**Collaborators**
The responsibilities of this class are to provide information about the 2D device.	MapType

EnergyProductionRecords	
Responsibilities	**Collaborators**
The responsibilities of this class are to provide records of energy production.	DataServer PreviousProductionMap RealTimeProductionMap

PowerPlantDetails	
Responsibilities	**Collaborators**
The responsibilities of this class are to provide the total number of solar panels and the average efficiency of panels.	SolarPanel EnergyMonitor

DataServer	
Responsibilities	**Collaborators**
The responsibilities of this interface are to connect to an external server to store and retrieve all data related to the energy system.	EnergyMonitor EnergyProductionRecords EnergyConsumptionRecords EnergyStorageManagement Sensors

DisplayDevice	
Responsibilities	**Collaborators**
The responsibilities of this class are to provide the user with an interactive system for viewing data.	HeatMap MapType SystemLanguages ZoomingOnBuilding

EnergyStorageManagement	
Responsibilities	**Collaborators**
The responsibilities of this class are to ensure energy levels stay within safe limits and take the necessary actions as advised by the AI System, when they cross the set limits.	SettingLimit AiSystem CityGrid DataServer

SettingLimit	
Responsibilities	**Collaborators**
The responsibilities of this class are to set the limits which stored energy should meet.	EnergyStorageManagement

CityGrid	
Responsibilities	**Collaborators**
The responsibility of this interface is to provide methods required for interaction with the city grid.	EnergyStorageManagement

AiSystem	
Responsibilities	**Collaborators**
The responsibility of this interface is to provide a set of operations about energy comparison and prediction.	EnergyStorageManagement NewSettings SystemFaults DigitalTwin BmsAction

Sensors	
Responsibilities	**collaborators**
The responsibility of this class is to collect the data of the temperature, humidity, light level in the room.	Temperature Humidity LightLevel DataServer EnvironmentalConditions

BmsAction	
Responsibilities	**collaborators**
The responsibilities of this class are to collect the data from the EnviornmentConditions and room cameras and send them to the AI system and control the indoor environment.	RoomCamera AiSystem EnvironmentalConditions

NewSettings	
Responsibilities	**collaborators**
The responsibility of this class is to set room settings as advised by AiSystem.	AiSystem

RoomCamera	
Responsibilities	**collaborators**
The responsibility of this class is to collect the data on the number of people in the room.	BmsAction

Temperature	
Responsibilities	**collaborators**
The responsibility of this class is to measure temperature data and provide it to Sensors.	Sensors

Humidity	
Responsibilities	**collaborators**
The responsibility of this class is to measure humidity data and provide it to Sensors.	Sensors

Lightlevel	
Responsibilities	**collaborators**
The responsibility of this class is to measure light level data and provide it to Sensors.	Sensors

EnvironmentalConditions	
Responsibilities	**collaborators**
The responsibility of this class is to use sensor's data to define the current environment condition.	Sensors BmsAction

SystemFaults	
Responsibilities	**collaborators**
The responsibility of this class is to monitor the system for any faults and provide the functionality to assess and mitigate damage.	AiSystem

B5.iii & B5.iv: First-cut class diagram* and detailed class diagram*:

The first-cut class diagram is an initial diagram that represents the relations between the classes without referring to the system methods and details. While the detailed class diagram illustrates the structure of the system by showing the classes, their attributes, and the relationships among each class.

Both the first-cut diagram and the detailed class diagram can be found in the pages below.

*A PDF version of the first-cut class diagram and the detailed class diagram can be found in the supporting documents folder in a higher resolution.

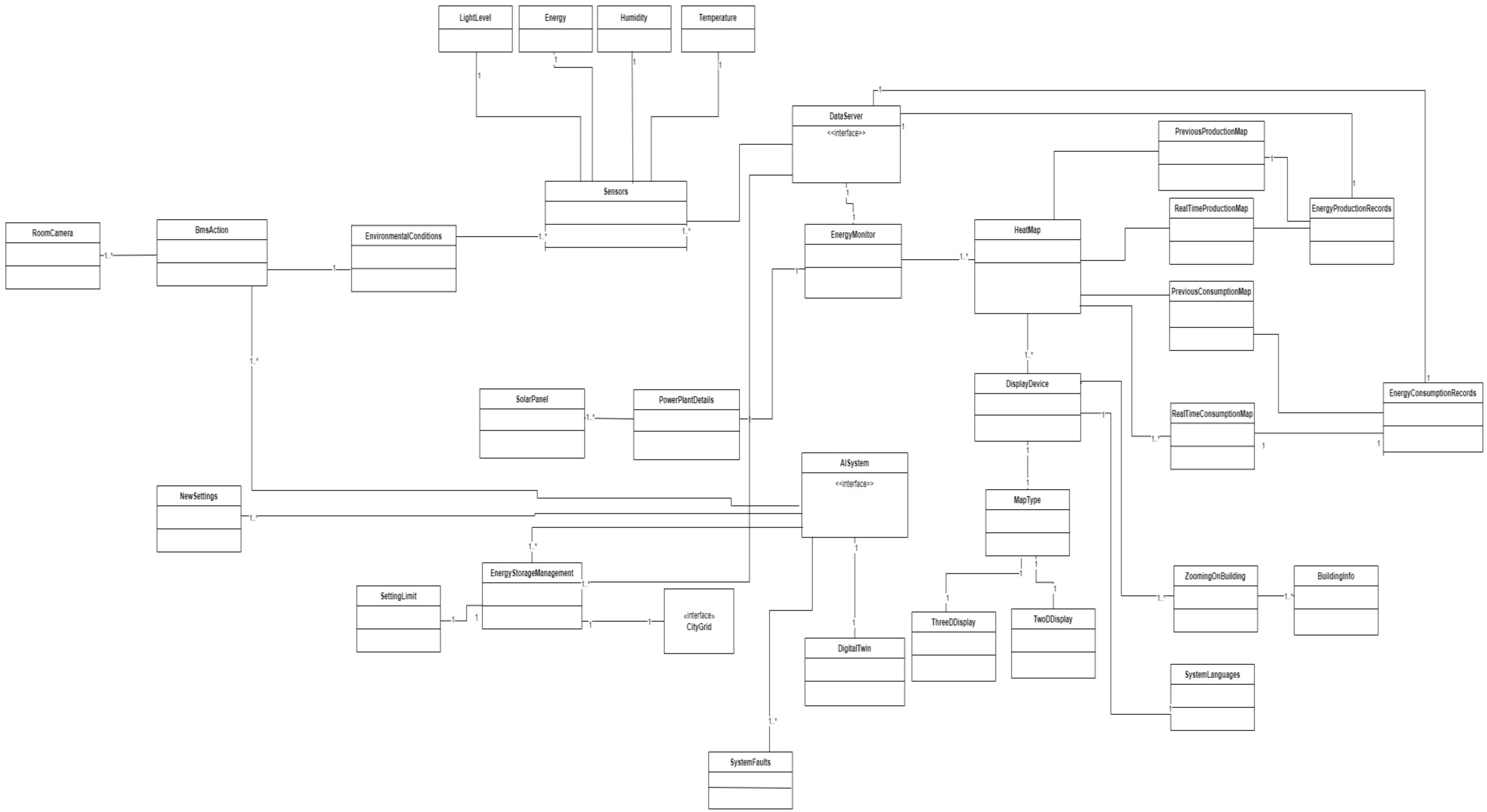

Figure 3: First-cut class diagram

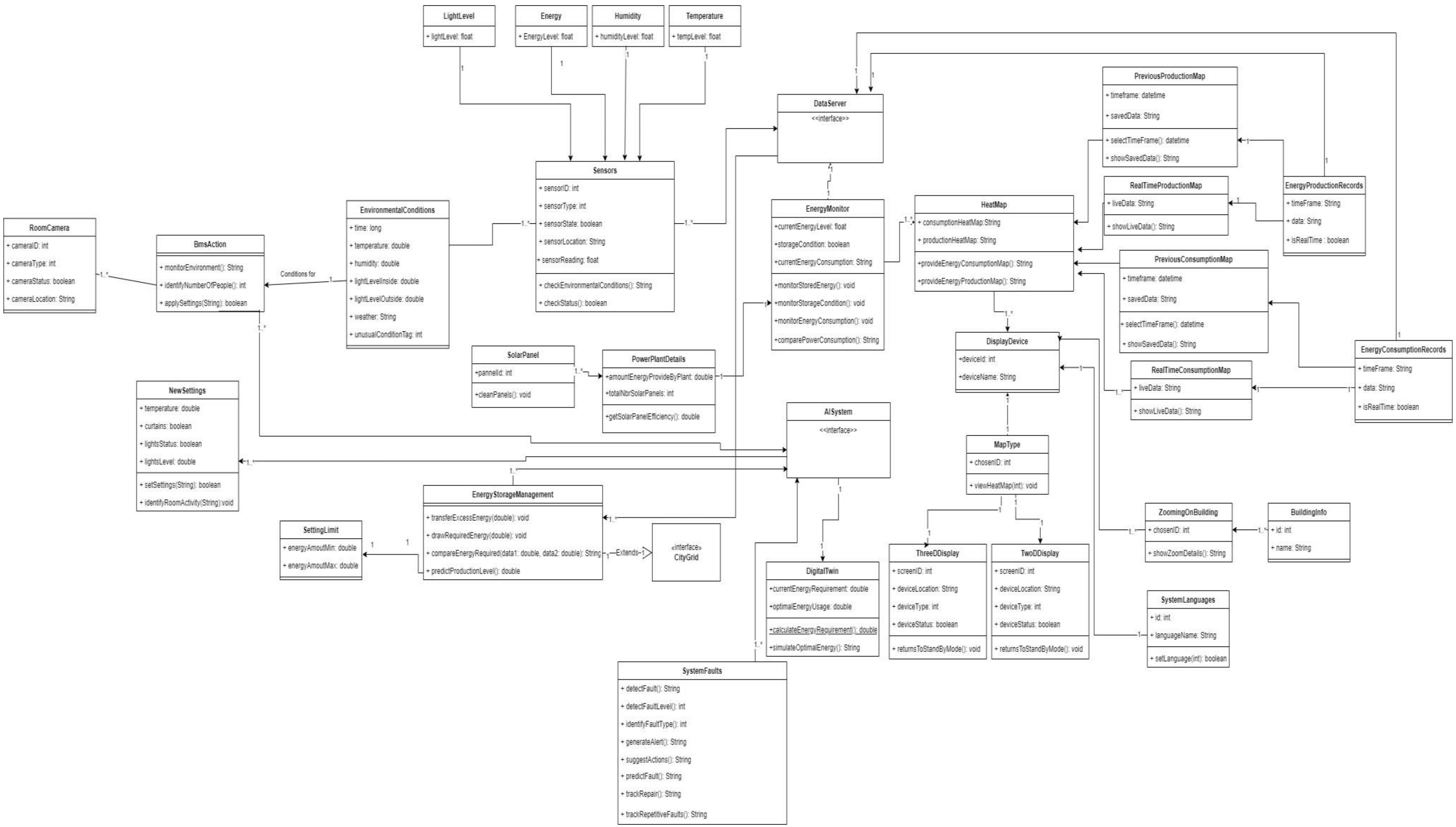

Figure 4: Detailed class diagram

B6: Object diagram*

An instance of the class diagram that illustrates the status of the objects and classes of the system during a certain scenario at a point of time.

The object diagram shown on page 45 Figure 5, is based on the scenario explained in B2. Use case diagram 2: View energy maps, second scenario (data server fails to connect). The scenario is found in page 19.

*A PDF version of the object diagram can be found in the supporting documents folder in a higher resolution.

B7: Sequence diagrams*

The sequence diagram illustrates the interactions and behaviours through a single scenario execution in the system.

Sequence diagram (1):

The diagram shown on page 46 is based on the environmental adaption use case. It shows the scenario where the AI system becomes unresponsive. As this case may happen through the basic workflow, both the normal workflow and the alternative workflow are included. Both flows are explained on pages 14 and 15.

Note:
As the AI system, sensors, cameras, lights, AC's and the data base are running (Alive) most of the time in real life, the diagram only reflects the parts where they are performing a task in the sequence to make it easy to track, which is why they are not shown as live all the time.

Sequence diagram (2):

The diagram shown on page 47 is based on the view energy maps use case. It shows the normal workflow scenario of the use case, which is explained on pages 16 and 17.

Note:
As the display scenario is based on the continuous interaction between the user and the display device, they are both displayed as alive all the time in the diagram. Unlike the energy monitoring unit and the data server that has a certain time where they are performing a task through the workflow to make it easy to track, which is why they are not shown as live all the time.

*A PDF version of both sequence diagrams can be found in the supporting documents folder in a higher resolution.

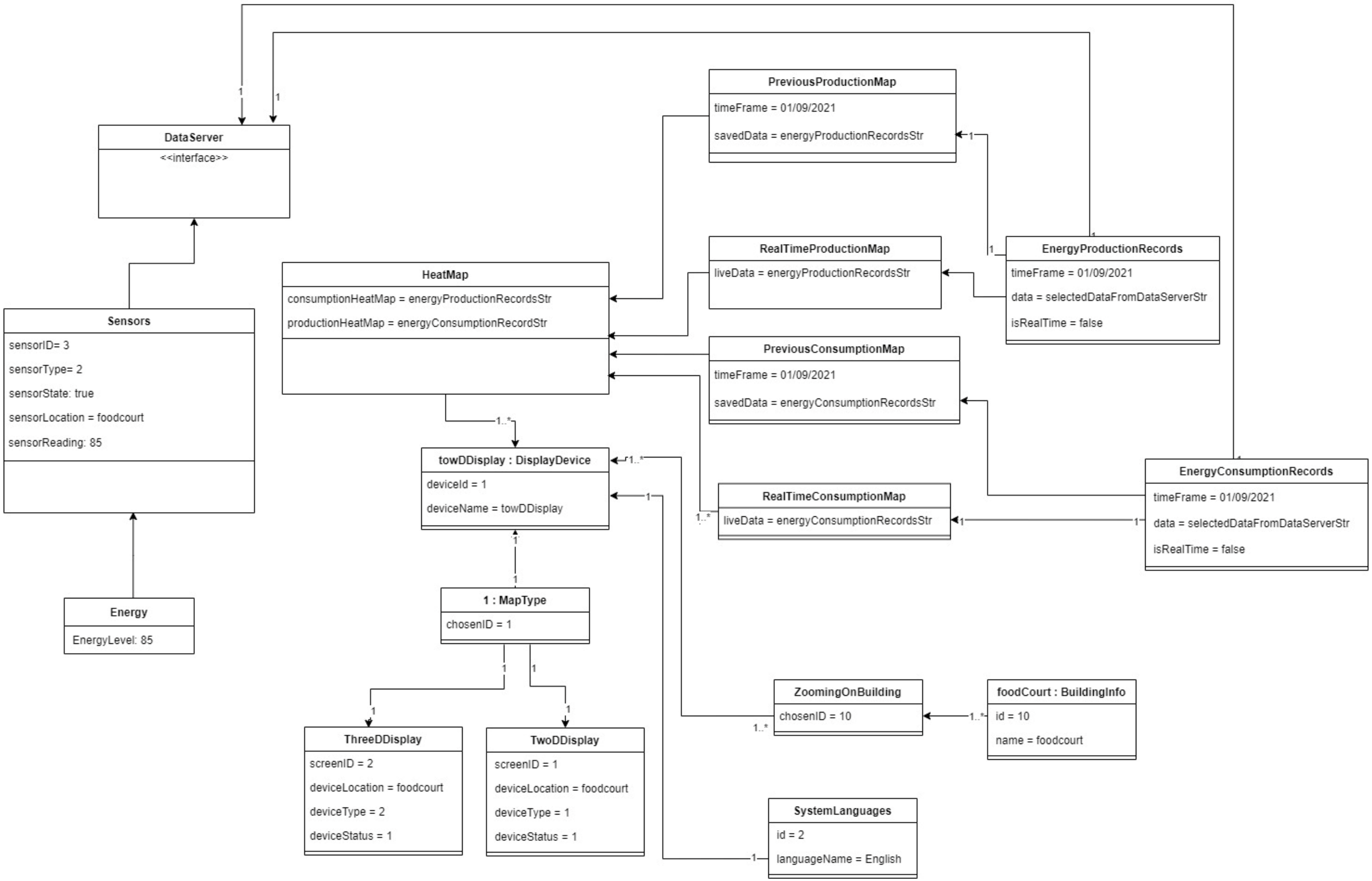

Figure 5: Object diagram

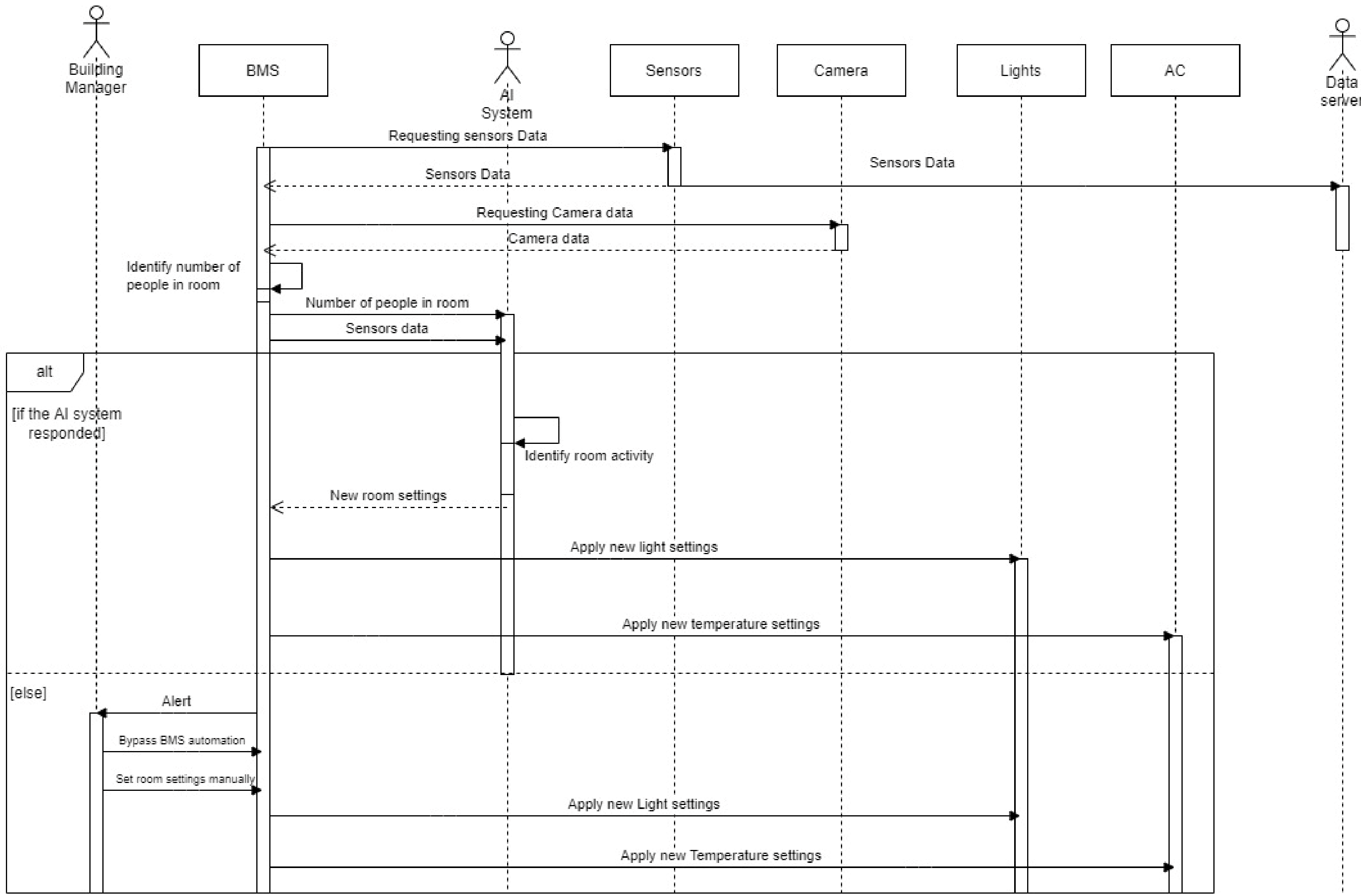

Figure 6: Sequence diagram (1)

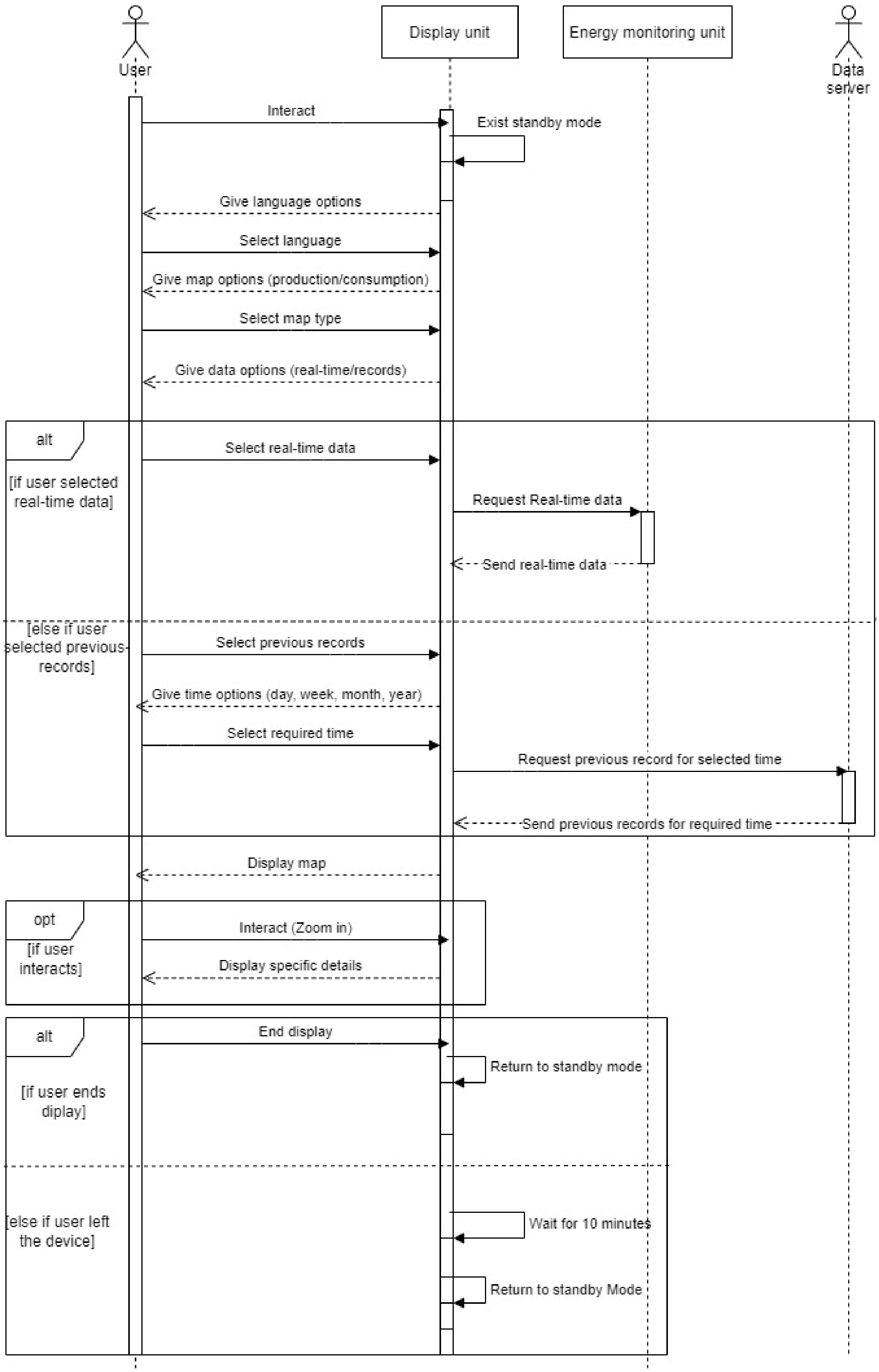

Figure 7: Sequence diagram (2)

B8: State machine diagrams*

The state machine diagram illustrates the states of an object, or a subsystem can have on runtime.

<u>State machine diagram (1): Building Management System</u>

The state machine diagram shown on page 49 is for the BMS component. It reflects the states inside the BMS when running. The parallel states inside the BMS are reflected in the diagram.

<u>Note:</u>
The AI system shown in the figure is an interface that is only included to help illustrate the states of the diagram.

<u>State machine diagram (2): Display devices</u>

The state machine diagram shown on page 50 is for the display unit, where all the possible states are included. As the interaction with the display devices depends on the choices of the user, several guard conditions are included as well for the parallel states.

<u>Note:</u>
The data server interface and the energy monitoring unit are included to illustrate the states of the diagram.

*A PDF version of both machine state diagrams can be found in the supporting documents folder in a higher resolution.

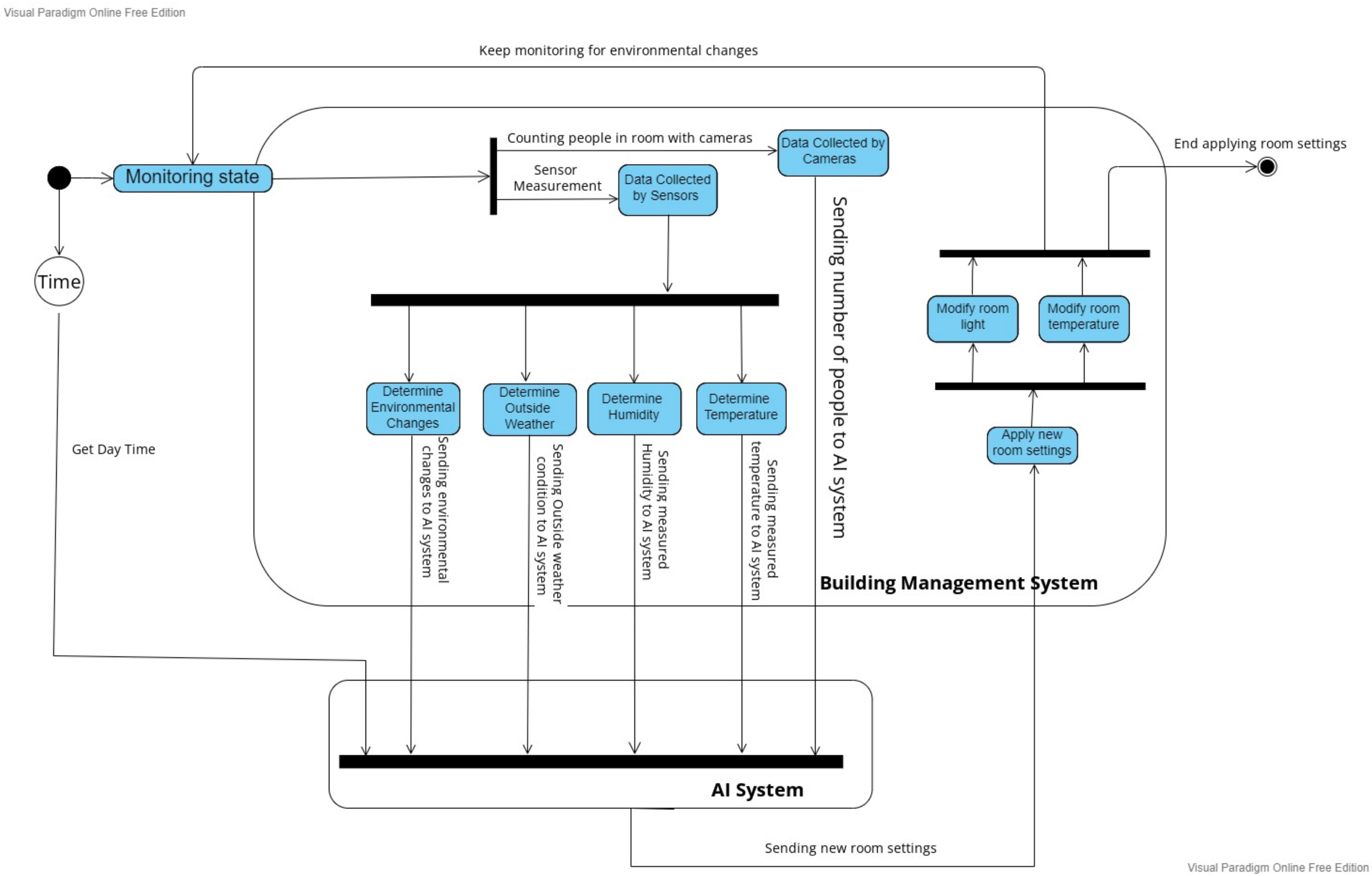

Figure 8: State machine diagram (1)

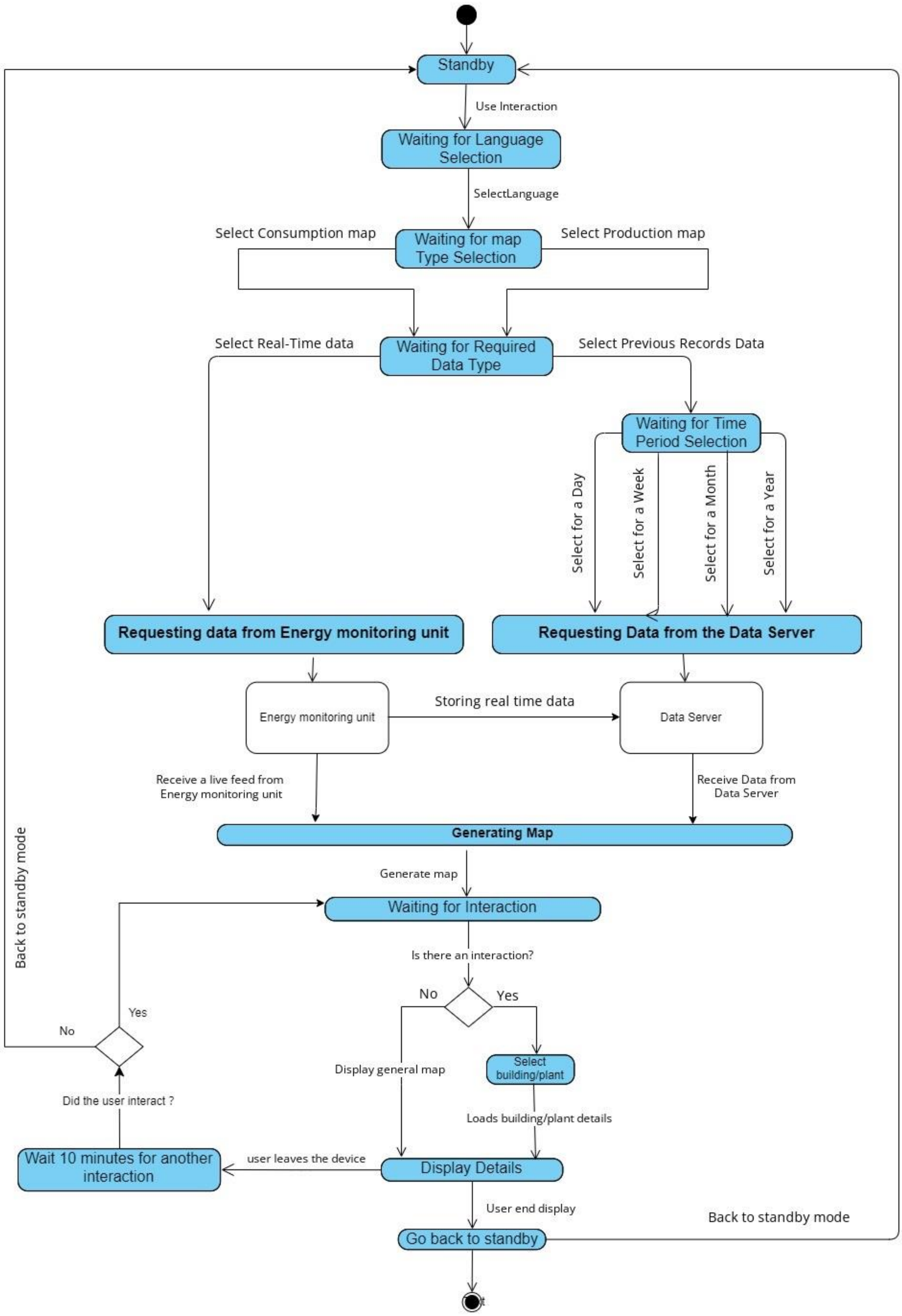

Figure 9: State machine diagram (2)

Part C: Software Architecture Style, Modelling and Evaluation

C1: Candidate architecture styles component diagrams

C2: Candidate architecture styles deployment diagrams

C3: Comparison between the architectures

C1: Component diagrams

Based on the complex nature of our system, we studied several candidate architectures to implement. Eventually, we found that the best architectures that fit the system needs were the layered architecture and a hybrid between the blackboard, client-server and peer-to-peer architectures.

Layered architecture component diagram:

The component diagram of the layered architecture is shown on page 53. The system is composed of four layers that include all the main components and interfaces. The colours in the back are used to illustrate the borders of the layers.

- Layer 1: BMS, room camera and solar panel
- Layer 2: AI system interface, Data server interface and Energy monitoring unit
- Layer 3: Fault monitor unit and Heat mapping component
- Layer 4: Display devices

Blackboard, client-server and peer-to-peer architectures component diagram:

The component diagram of this hybrid architecture is shown on page 54. The distribution of the components in this hybrid architecture is as follows:

The blackboard architecture includes:
- AI system interface
- Fault monitor unit
- Building control units (the peer-to-peer is a sub architecture in the blackboard)

The client-server architecture include:
- Data server interface
- Display units that include:
    - Heat mapping component
    - Display devices

The peer-to-peer sub-architecture includes several building control units, where each unit includes:
- BMS
- Solar panel
- EMU
- Room camera
- Database (local)

The details of the building control unit and the building management system are shown on page 58 Figures 12 and 13.

Assumptions:
- Each building includes a local database, in consistency with the non-functional feature of the separate database previously explained in part A.
- The AI system pools the data that it receives before applying any functionality aligning with the blackboard architecture.

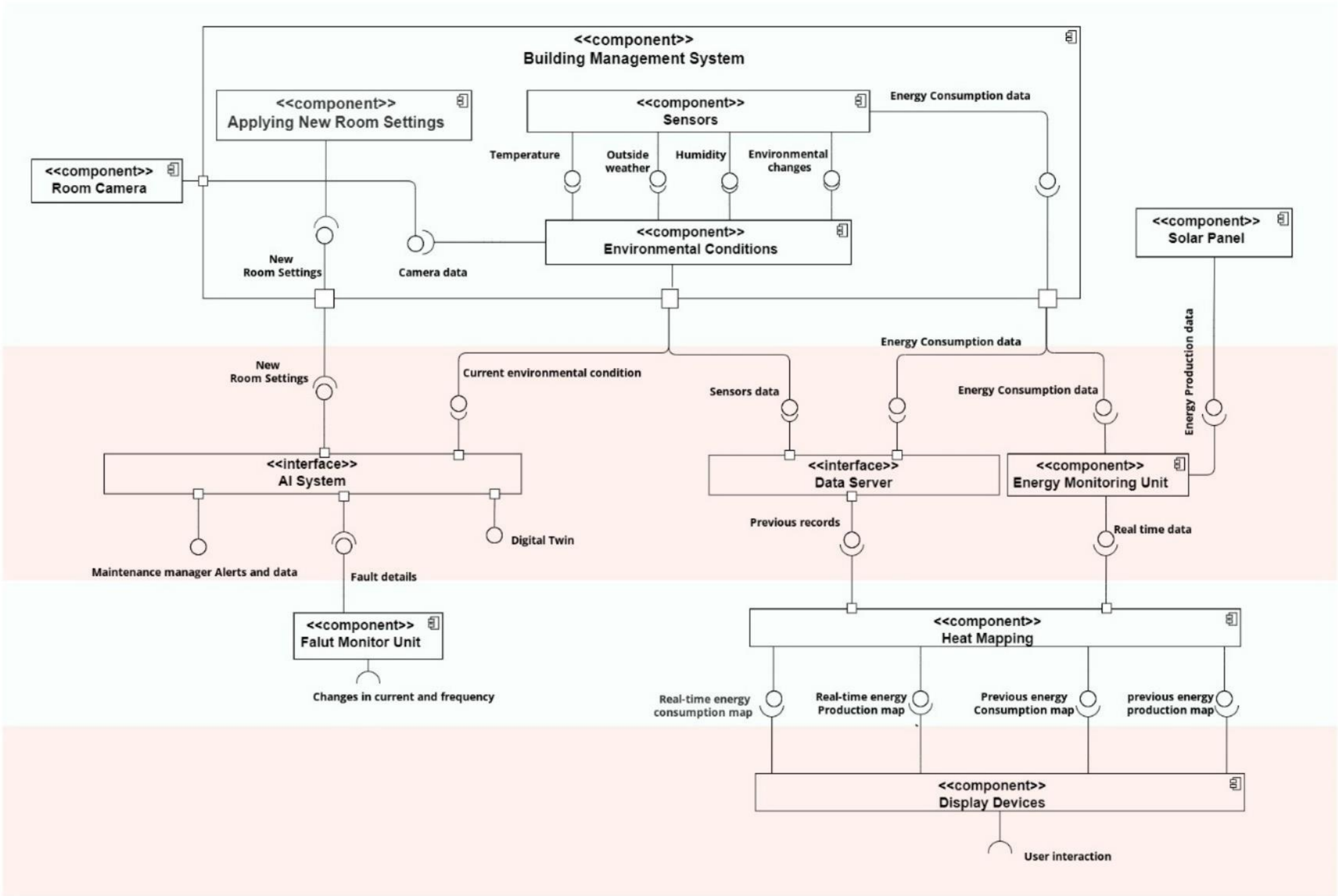

Figure 10: Layered architecture component diagram

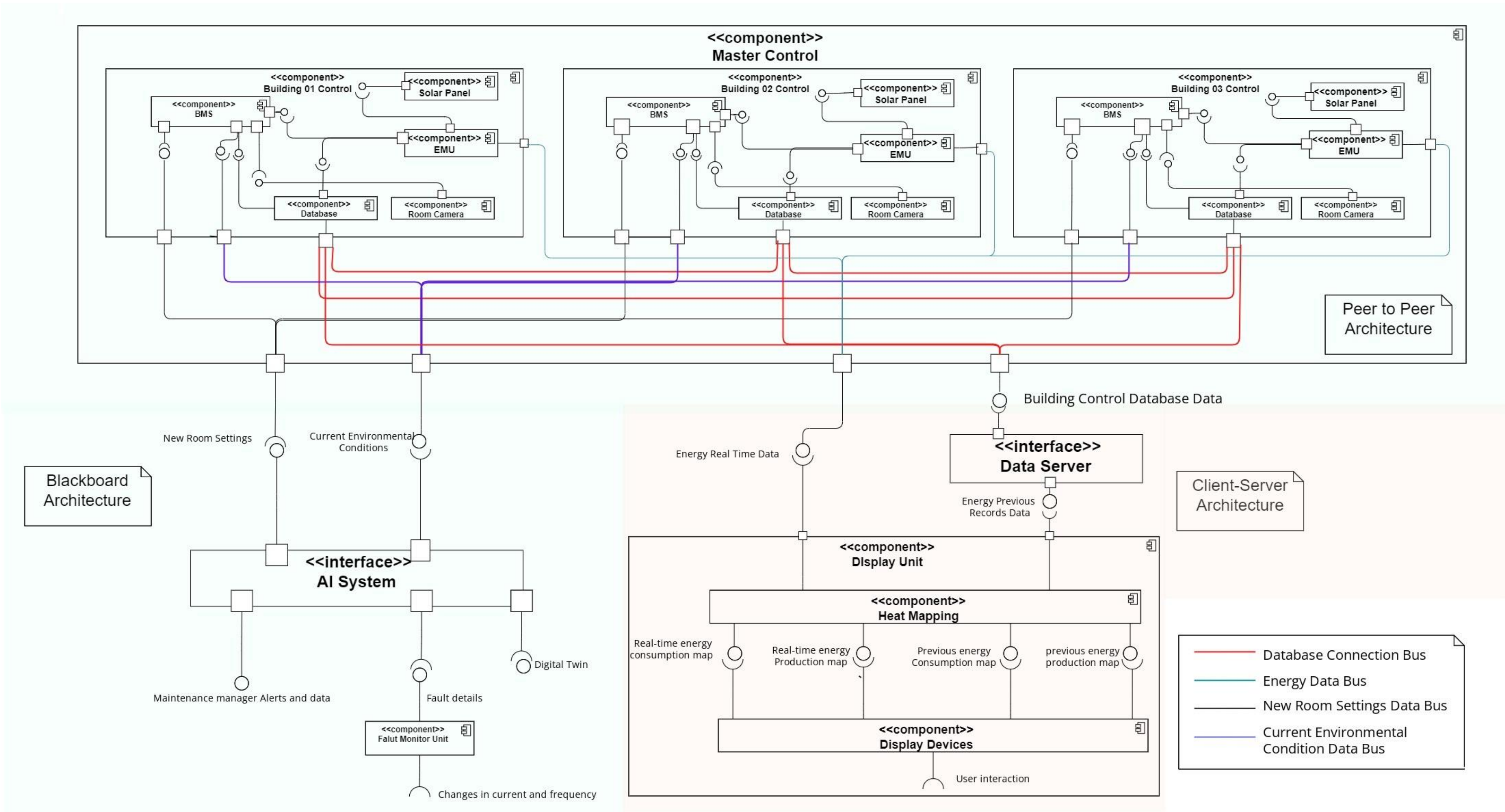

Figure 11: Blackboard, client-server and peer-to-peer architecture component diagram

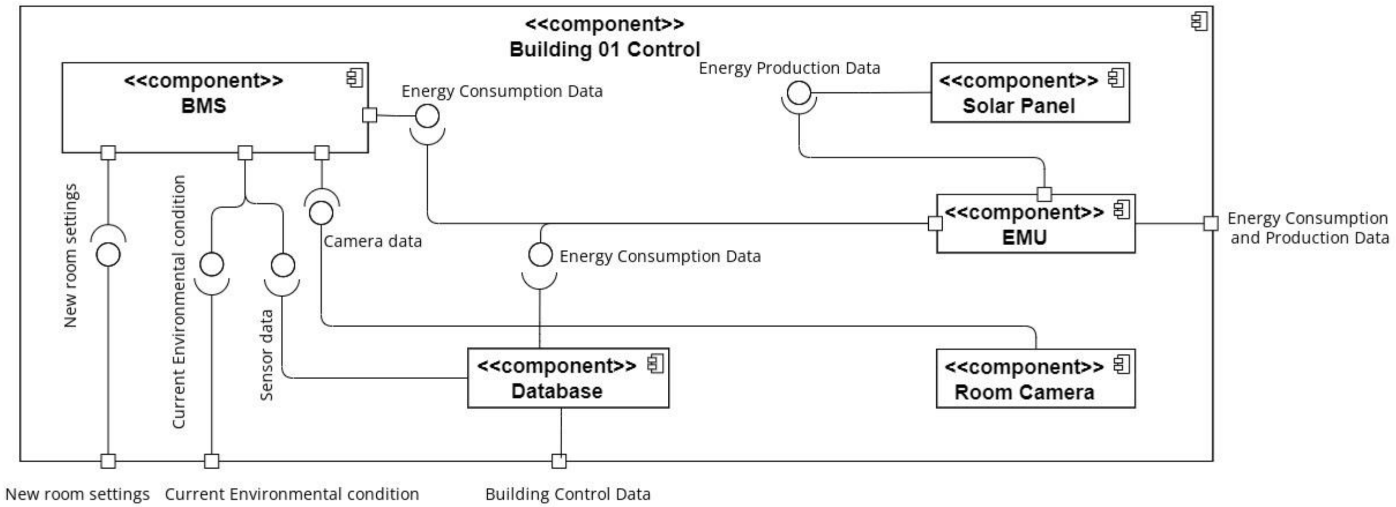

Figure 12: Building control details

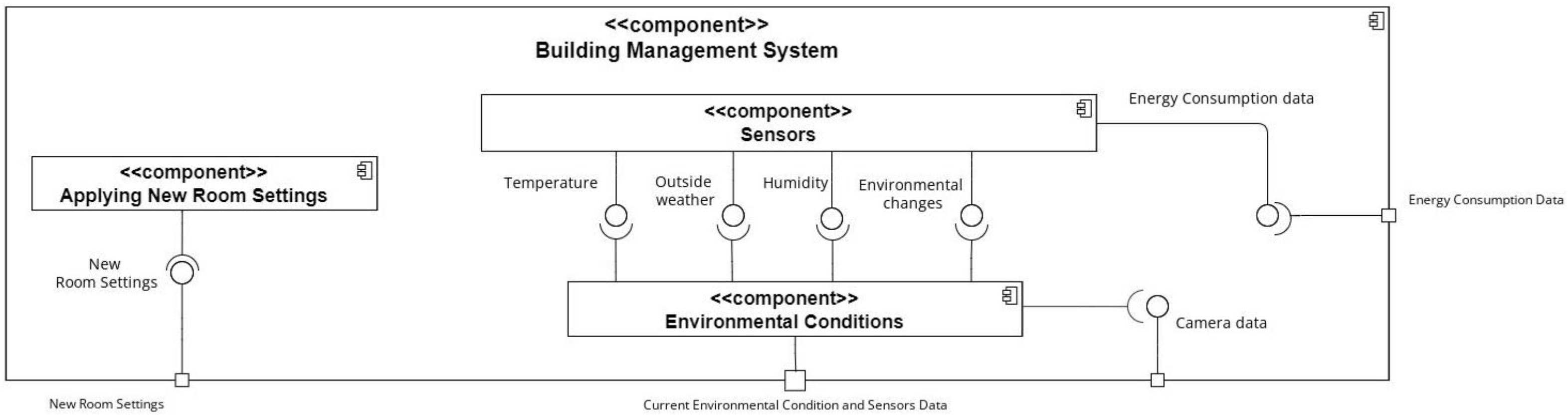

Figure 13: BMS details

C2: Deployment diagrams

A diagram that models the runtime configuration as a static view and visualizes the distribution of components in an application.

The deployment diagram of the layered architecture is shown on page 57, figure 14.

The deployment diagram of the blackboard, client-server and peer-to-peer architecture is shown on page 58, figure 15.

The same assumptions made for the component diagram are considered for the deployment diagram.

*A PDF version of both component diagrams and deployment diagrams can be found in the supporting documents folder in a higher resolution.

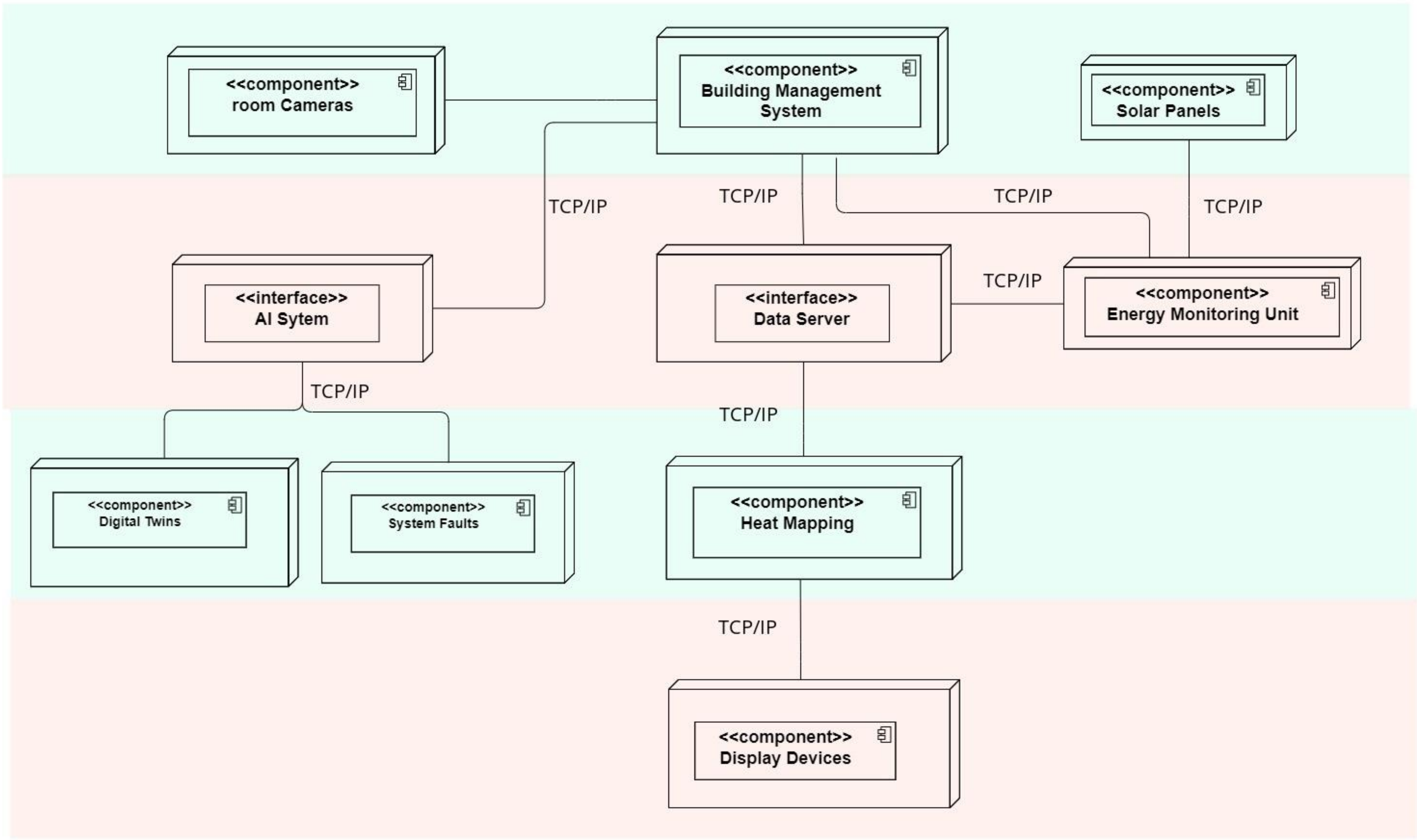

Figure 14: Layered architecture deployment diagram

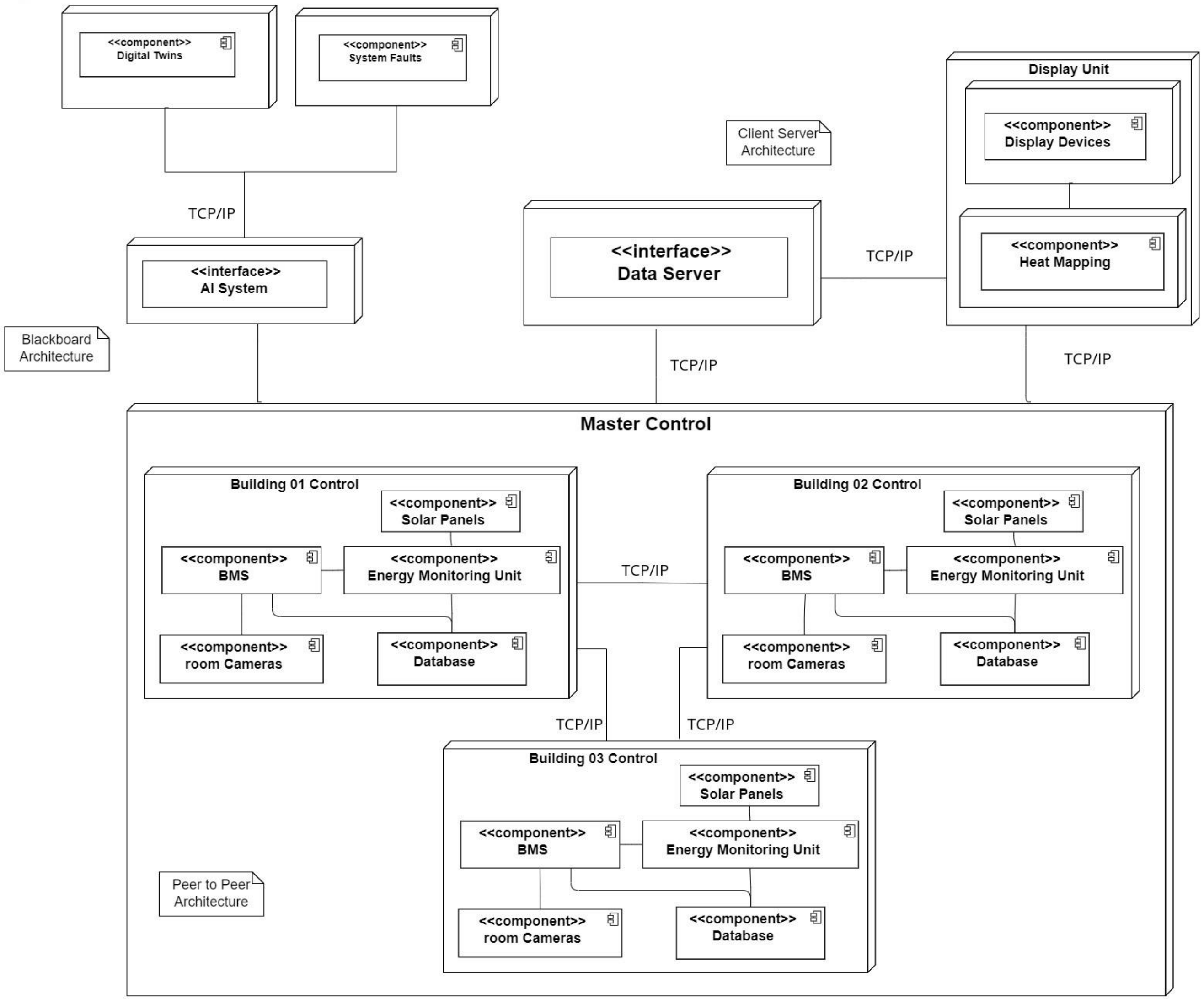

Figure 15: Blackboard, client-server and peer-to-peer architecture deployment diagram

C3: Architecture comparison

Two software architectures are proposed for the system based on the features and requirements of the system as the guidelines for the design process. An analysis of the benefits and trade-offs of the proposed architectures is conducted below to select the ideal system design.

Architecture-A is proposed as a layered design. It defines the separation of components into levels to allow the system to adapt and change components as required. Therefore, addressing maintainability and evolvability. The system will be portable as only the outermost layers will need to be changed to fit new environments, allowing implementation in multiple campuses. This approach will allow for a higher level of security with critical systems at lower levels. Trade-offs are performance, scalability, and robustness. Due to the closed layers, messages will need to travel through all layers between their source and destination, resulting in performance overheads. Additionally, system robustness will be a concern as critical faults in lower layers can affect the entire system.

Architecture-B is proposed as a combination of three architectures tailored to fit the responsibilities of the components of the system. It implements the AI system as a blackboard along with the building control units and the fault monitoring unit as the blackboard clients. This allows multiple components to send information while accessing functionality provided by the AI system. Furthermore, large amounts of data can be shared efficiently, addressing the performance requirement. System robustness can be ensured by having backup options from the service provider to use in case of failure. Components related to building functionality are grouped in a building control unit for each building. Each unit is deployed in a peer-to-peer architecture with others. This allows the system to remain functional in case of an individual fault. The devices and heat mapping component are included in the display unit which is connected to the data server using a client-server architecture. This allows scalability for these components, allowing the university to address the increase in users in the future while keeping the system robust. Implementation and portability are the main trade-offs of this architecture.

While architecture-A presents a portable and maintainable system with low-cost implementation and modifiability in the future, the performance and robustness that architecture-B provides are crucial for a system that manages critical functions. Hence architecture-B is chosen as the primary design for the system.

Part D: Software testing

- Testing plan
- Testing objective
- Testing strategy
- Testing exit criteria

D: Software testing

Introduction

The purpose of the testing plan is to assess the aspects of the software and ensure they are reliable and working accurately.
We will identify the test items that include the components of the functional and non-functional requirements which will be tested. After that, we will explain the testing strategy. The testing strategy consists of the testing methods and what to do when a test fails. The quality factors and how to measure them will also be discussed.
The pass/fail criteria will include all the test cases, inputs and expected results.
Finally, the exit criteria will determine whether the past/fail criteria resulted in the desired outcome or there was a need to refer to the test-fix diagram.

Test items

The test items will include the components that are related to the requirements to be tested. They include:

- BMS
- Users
- sensors (temperature)
- system faults
- solar panels
- Display device
- 3D display
- 2D display
- Room Camera

Quality factors and how to measure them:

- **Reliability**:
  The FMU ability to switch to the alternative system within one hour from the failure.

- **Security**:
  The ability of the system to prevent unauthorised users from access.

- **Response time:**
  The amount of time needed for the display device to get the data after the user requests it.

- **Performance**:
  The ability of the room cameras to count the number of people in the room accurately.

- **Usability**:
  The ability of the users to interact with the display devices and select types of maps to display.

- **Interoperability**:
  The ability of the BMS to connect to the sensors and other systems to exchange data between them.

Features to be tested

Functional requirements:

1. The BMS must be able to identify the number of people in the room at any time.
2. The BMS must be able to monitor the temperature inside and outside rooms.
3. The BMS should prevent people in the room from opening windows manually when the AC is running.
4. The FMU could alert the maintenance team if a certain fault is repeated, how many times and in which parts.
5. The FMU should categorise the risk of each fault into low, medium, or high risk.
6. The 3D display units should display data in 3D using virtual reality and holograms.
7. The panels monitoring unit could automatically clean the solar panels when their efficiency drops because of dirt.

Non-functional requirements:

1. The system should be compatible with different devices like PC's and mobile phones.
2. The system must prevent any unauthorised users from accessing the data.
3. An alternative fault monitoring system could be online within 1 hour, in case of a system breakdown.

Features not to be tested:

All remaining requirements of the system (functional and non-functional) will not be tested due to the time constrain and the requirement of 7 functional and 3 non-functional requirements only.

Testing strategy

The features will undergo a series of tests which includes black box and white box testing along with their different methods according to each feature and its components. This will help determine whether the features pass or fail the test cases.

If a feature fails a test, it will go through the test-fix method, as shown in the figure below. The feature will be assigned to a developer to fix its problem, when the problem is fixed, the feature will undergo the test again. The process will repeat until the feature passes the test. The importance of a feature passing the test depends on the fault risk level and the importance of the feature in the system.

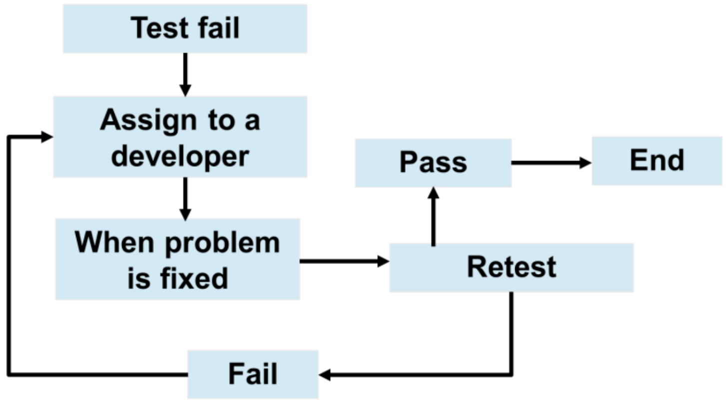

Figure 16: Testing strategy

Pass/Fail criteria

We need to run all the test cases described in the template. All the functionalities should work as described without any critical defects.

Test Case ID	Test Description	Test Steps	Test Data	Expected result	Actual result	Pass / Fail	Test comments
TD-015	The BMS must be able to identify the number of people in the room at any time.	Several people, within the range of room capacity (e.g., 15 people) enter a room. The BMS will use the room camera to count the number of people in the room.	-Number of people in the room.	The BMS should be able to provide a precise count of the number of people in the room.			
TD-032	The BMS must be able to monitor the temperature inside and outside rooms.	The building manager can record the temperature values inside and outside a room. The results are to be compared with the temperatures recorded by the BMS sensors.	-Recorded temperatures by the manager. -Temperatures measured by the BMS sensors.	The temperature values given by the BMS should be the similar to the actual values observed with a very low margin of error (less than 0.5 C).			
TD-020	The BMS should prevent people in the room from opening windows manually when the AC is on.	A student tries to open a window when the AC is on.	-Window status (opened/closed) -AC status (on/off).	The window should remain closed.			
TD-021	The BMS should prevent people in the room from opening windows manually when the AC is on.	A student tries to open a window when the AC is off.	-Window status (opened/closed) -AC status (on/off).	The window should successfully open.			

TD-037	The FMU could alert the maintenance team if a certain fault is repeated, how many times and in which parts.	The head of maintenance can cause a specific fault three different times in the same area of the system.	-The fault -Fault area -Number of times fault repeated.	The FMU should send an alert to the maintenance team, containing the details of the fault, its area, and how many time it was repeated.			
TD-048	The FMU should categorise the risk of each fault into low, medium, or high risk.	The maintenance team can create three different faults from different categories.	-Faults -Fault level	The FMU should correctly categorise each fault according to level of risk.			
TD-064	The 3D display units should display data in 3D using virtual reality and holograms.	The user can view the data using a 3D device from the campus.	-3D Display unit -Required data	The display unit should display the data in 3D.			
TD-050	The panels monitoring unit could automatically clean the solar panels when their efficiency drops because of dirt.	The system manager can cover a solar panel with dirt to reduce its efficiency.	-Solar panel efficiency.	The system should automatically clean the solar panel.			
TD-078	The system should be compatible with different devices like PC's and mobile phones.	The user can login with three different devices (e.g., a laptop, mobile phone, and computer).	-Compatibility to the device	The user should be able to login with all three devices, with a compatible view for each one.			
TD-056	The system must prevent any unauthorised users from accessing the data.	The user can login with an invalid username and password.	-Username -Password	The system should prevent the user from accessing the data.			

TD-057	The system must prevent any unauthorised users from accessing the data.	The user can login with a valid username and password.	-Username -password	The system should grant the user access to the data.			
TD-096	An alternative fault monitoring system could be online within 1 hour, in case of a system breakdown.	The head of maintenance can suspend the fault monitoring unit.	-Time needed for the alternative fault monitoring system to come online	The alternative fault monitoring system should successfully be online within 1 hour.			

<u>Exit criteria</u>

- 100% of tests planned should be run.
- 95% of tests executed should be passed.
- All the system-critical features should pass.

Part E: Usability and prototyping

E1: Provide five main screens of the prototype

E2: Video recording

E1: An interactive prototype

Below is a description of the five main screens of the prototype, along with a detailed description of the process and what is included in each step. The entire prototype screens can be found in the supporting documents, and a demo for the prototype functionality can be found in E2 video.

Map type selection:

After the user interacts with the display device and selects the preferred language, the device will require him to select the type of map as shown in the Figure below.

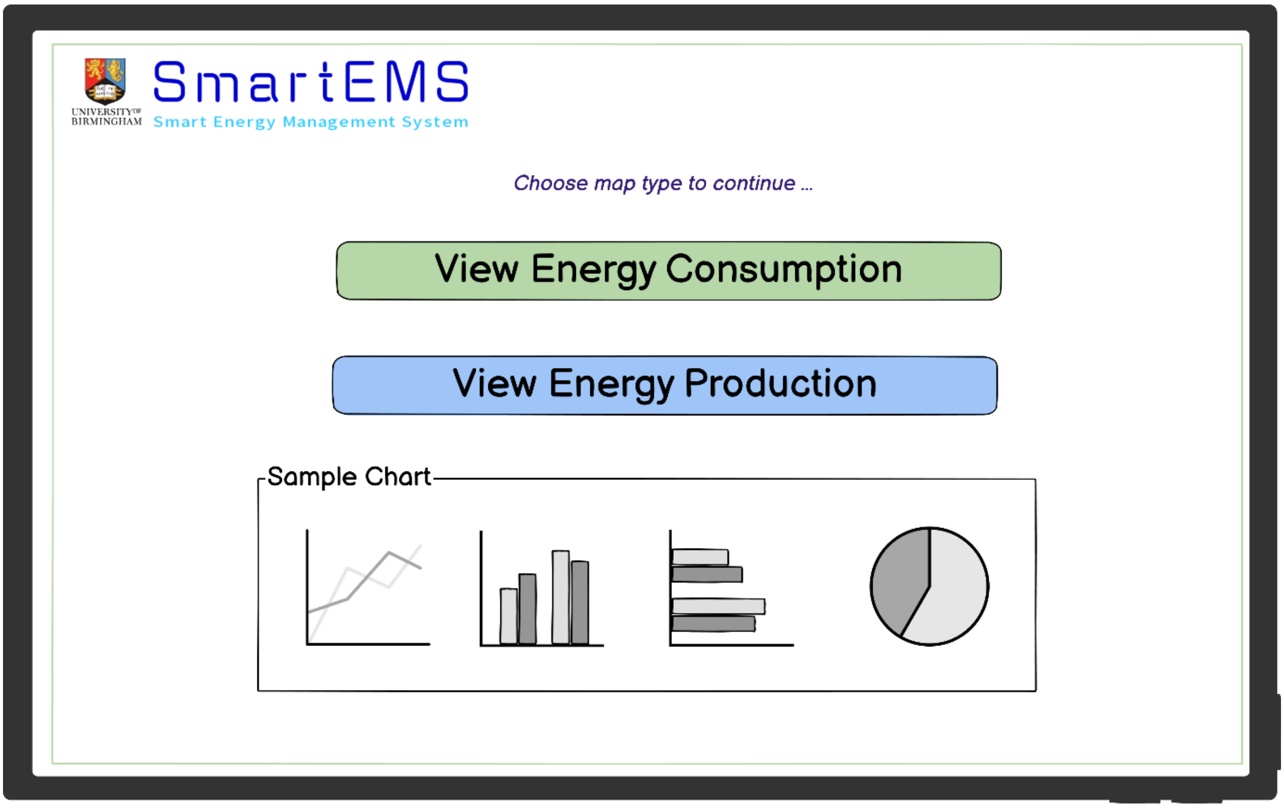

Figure 17: Map type selection

Data type selection:

As shown in Figure 18, the user can select from the tabs to determine to view the real-time or previous energy production data. He can also interact with the map and zoom in to view the exact project details, as shown in Figure 19, where the user has zoomed in power plant 3, and the exact real-time details of the power plant are shown, including its ID, number of solar panels, efficiency and energy production levels.

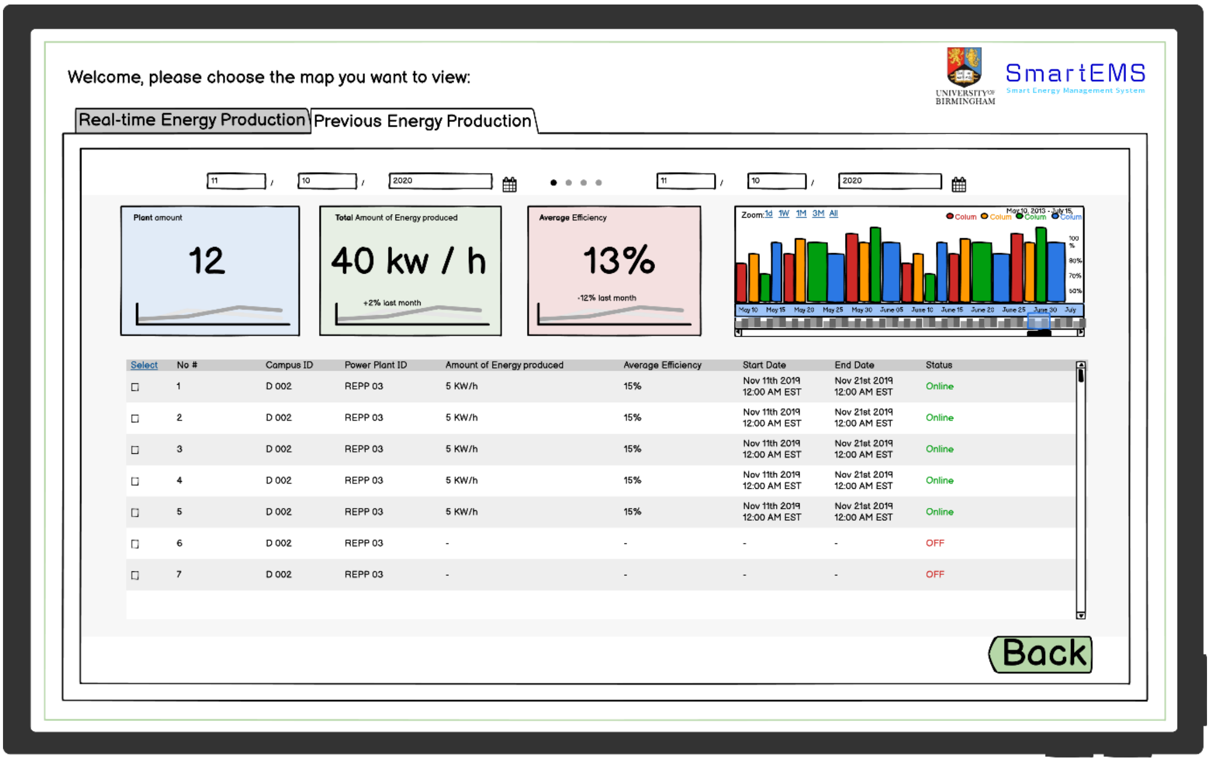

Figure 18: Previous energy production records

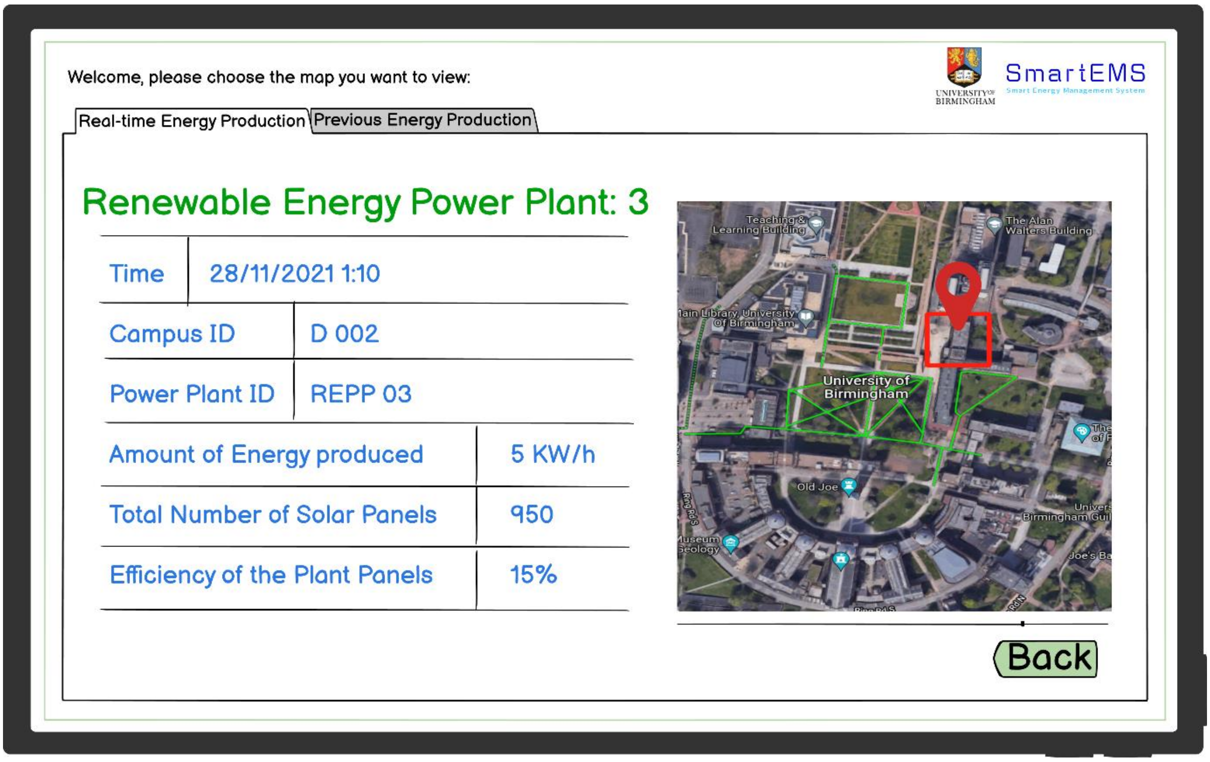

Figure 19: Real-time energy production of a specific plant

Admins interface:

The system has a separate interface for the admins, like the building manager and the maintenance manager. This interface requires a password and an email to log in.

After logging in, each admin will have a user interface with functionalities related to his tasks, as seen in Figures 20 and 21.

Figure 20 shows the interface of the maintenance manager, where he received notes regarding the faults and the unresolved matters. It also illustrates the alerts sent by the AI system that tracks the maintenance of the faults detected by the fault monitoring unit.

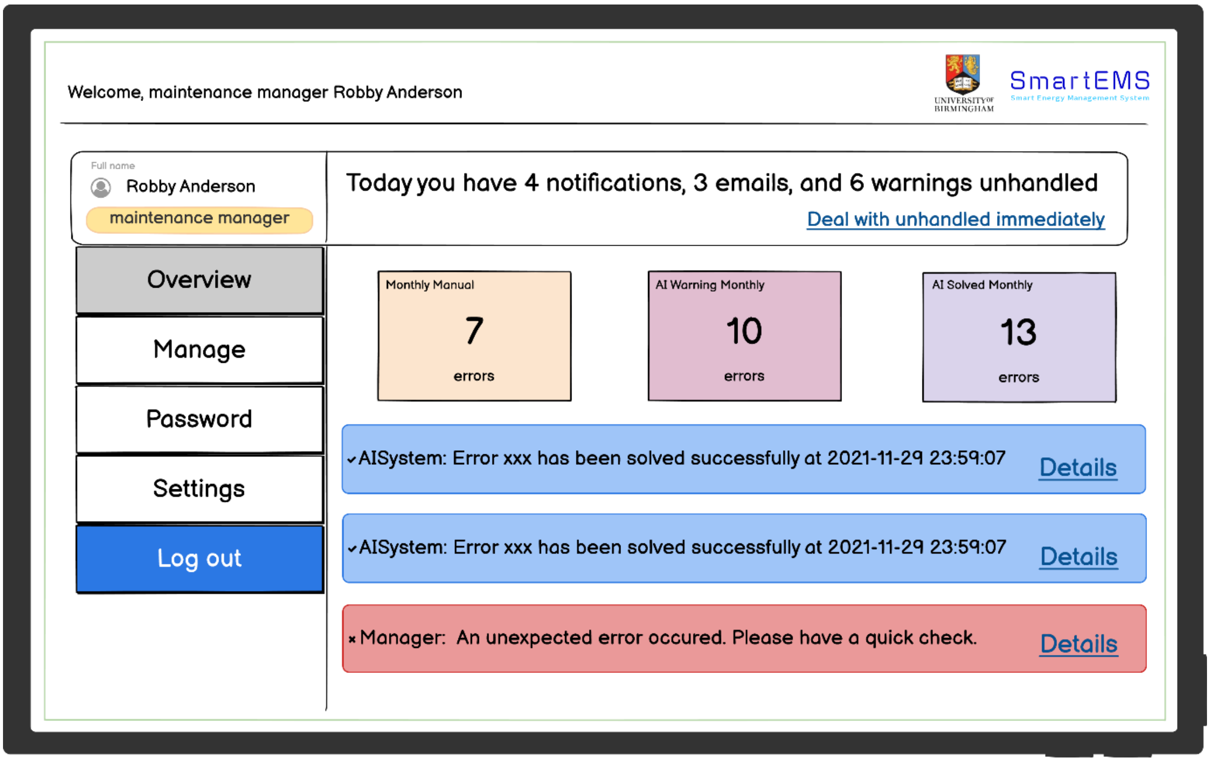

Figure 20: Maintenance manager interface

Figure 21 shows the building manager interface, where it gives him the ability to bypass the system and set the room temperature, humidity and light level and manually without interference from the AI system.

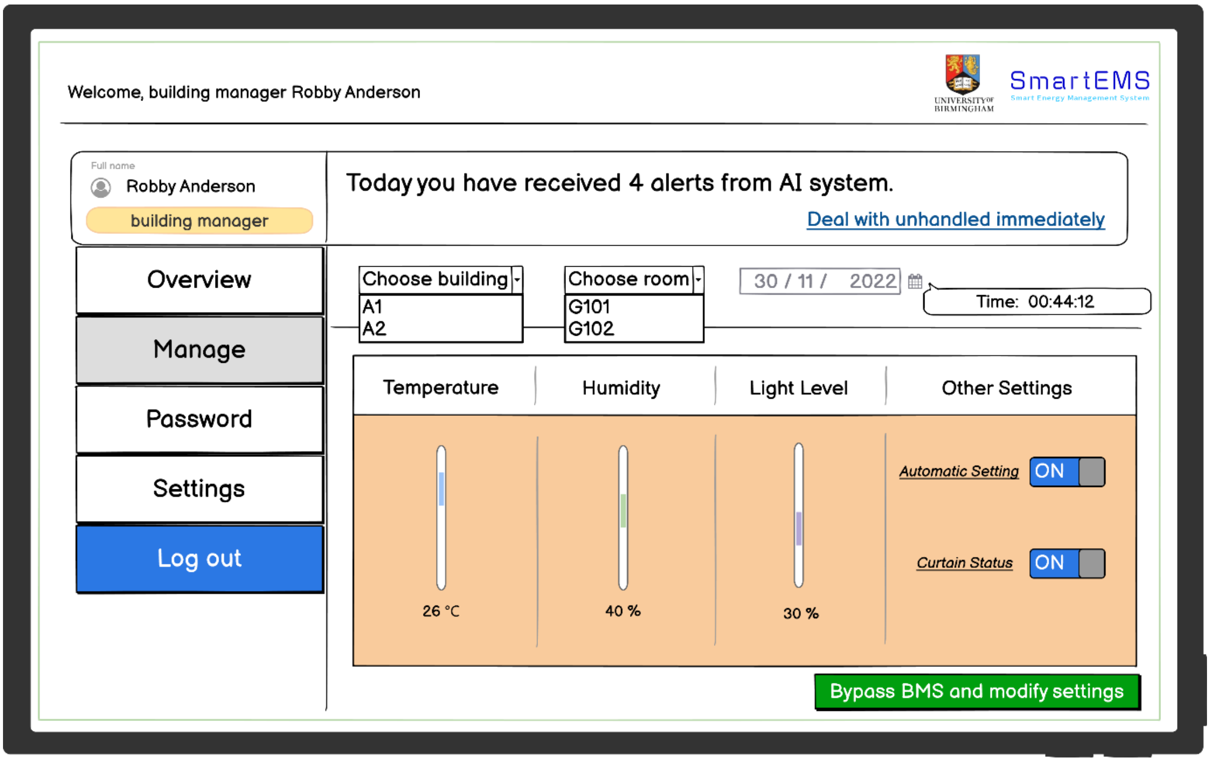

Figure 21: Building manager interface

E2: Video recording

Please refer to the attached video that describes the system, and shows a demo of its functionality.

Part F: Ethics and professional practice

An Appraisal

F: Ethics and professional practice

This appraisal shows how the Code (ACM code of ethics and professional conduct) was demonstrated in this project.

Clause of The Code*	Demonstration of Application
1	**General ethical principles**
1.1	The aim of the system is to minimise the generation of harmful emissions due to campus activities. It also provides educational and research facilitation features.
1.2	The system design includes several safety features. Furthermore, during the project, the health situation of team members was respected, where task allocation was adjusted to avoid overburden.
1.3	Users would be educated about the system's data handling process.
1.4	The tasks were fairly allocated to the team members.
1.5	Ideas given by team members were discussed and considered as possible.
1.6 + 1.7	The system prevents sharing of the private data of users taken by cameras. It also employs security features to assure that the users' data remain confidential.
2	**Professional responsibilities**
2.1 + 2.2	The team was striving to achieve high-quality results, which reflected in the meetings, QA process and referring to relevant references to confirm that each step was performed correctly.
2.3	Work rules were explained and agreed on upon the first meeting of the group, including how to handle each case and what's expected of everyone.
2.4	Peer review was applied to ensure that the work was done correctly. This process included giving and receiving feedback from team members. Feedback was also frequently requested from the lecturer.
2.5	When designing the system, possible risks were considered, and alternative solutions were provided, like the ability to bypass the automated system when needed and having alternative systems in case of a breakdown.
2.6	One of the factors considered during task allocation was the expertise of team members.
2.7	One of the system's features - "Heat mapping" - is allocated towards raising public awareness about energy, it demonstrates the important role of computing technologies in its management.
2.8	Only authorised personnel can access certain parts of the system.
2.9	The system architecture was selected to provide improved robustness. System administrators require a password protected login to make modifications.

3	**Professional leadership principles**
3.1	The public good was a major focal point in the design and aims of the system. The work management revolved around the interests of the team members.
3.3	The team manager took careful consideration of personal issues, such as health and assignments.
3.4	The project manager credited the work of team members and supported them when they had difficulty in carrying out work. The team manager included all team members in major decisions.
3.5	When members worked in parts that required extra knowledge, the team provided resources to help them achieve their potential and complete their tasks correctly.
3.6	Careful considerations were taken upon the effects of modifying any part of the system design.
4	**Compliance with the Code**
4.1	The team members generally followed work ethics in line with the Code.

*Refer to the ACM code of ethics and professional conduct.