Table of Contents

Assignment 1

A. Data Gathering Techniques

B. User Profiles

C. Task Analysis

D. Software Engineering Aspects

Database

Appendix A

Assignment 2

Conceptual Models:

Storyboard Model

Experience Map

Database Side

Assignment 3

Informed Consent Form

Selecting users

Field or controlled study

Application of Software Engineering

Database

Nielsen’s Heuristics

Prototype

Usability vs Heuristic evaluation

Evaluation Approaches

Application of Software Engineering:

Database Internal Model

Assignment 1

A. Data Gathering Techniques

We drafted a questionnaire with questions to basically gather requirements from users that could be used to build the system. We also gathered requirements from our own experience with home automation and security systems and tried to figure out what we could have embedded in our system.

Questionnaire is drafted below this section(Assignment 1) in Appendix A.

B. User Profiles

Profile 1 – Primary User (homeowner)

Persona: John Doe (Parent/Owner of the house)

John Doe is a medical doctor, divorced and lives with his two children, however, John is barely in the house because of his job and only comes back in the night time. He has a big house.

Experience: John Doe is computer savvy - he has experience working with computers. The proposed system would not be hard to operate for this user.

The user has the authority to change default system settings.

Scenario 1

Even though John works till very late, he is never without his phone and internet connection. John would like an extra blanket of security in knowing that his children and possessions are safe inside the house, and that he will be notified should anything in his house go wrong.

Profile 2 – Secondary User (other inhabitants of the house)

Persona: Mary Doe, Kian Doe (John Doe's children, ages 17 and 10 respectively)

The children go to school at 8am in the morning and come back at 3:30pm.

Experience: Novice users of this kind of system but have knowledge in the use of computers/technology.

This user can only use existing settings (i.e. the user cannot change the settings of the primary account)**.**

Scenario 2

The children are almost always on their phones, and they are fairly responsible children, but they are children and sometimes like to stay in their rooms for long hours without coming out – sometimes they forget to turn the lights on in the house. And in case of an emergency, they should be able to also get a notification so they can act fast. Suppose the children are in the house alone when a fire starts, they will hear the alarm that the temperature has spiked, and will be able to either escape and call for help, or stop the fire if they have the know-how. Simultaneously, John will be alerted of this on his phone via a notification and he can call for help.

C. Task Analysis

Scenario 1

John is planning a trip and would like to configure the system to automatically switch on the lights at sunset and switch them off at sunrise.

Analysis

0. Configuring light sensors
    1. Login to the system.
       1.1 Enter Username and password
    2. Navigate to a specific room
    3. Edit sensor settings
    4. Specify date to apply settings

Scenario 2:

Suppose the children are in the house alone when a fire starts, they will hear the
alarm that the temperature has spiked and will be able to either escape and call for help, or stop the fire if they have the know-how. Simultaneously, John will be alerted of this on his phone via a notification and he can call for help.

Analysis

0. Check room temperature
    1. Login to the system
    2. Navigate to the room of interest (that was specified in the SMS notification)
    3. Observe room temperature from the dashboard temperature widget
    4. If it is too high, call for help, otherwise use a fire extinguisher

D. Software Engineering Aspects

(UML class diagram attached below document: Appendix B)

Technical reviews (Umbrella Activity)

We plan to review completed modules to find and remove errors to not propagate them to the next module.

Software Engineering Goals - Quality

Security - We plan to ensure security in every stage of the software development life cycle
User will be able to provide user authentication which can be verified
Functionality - The software needs to work according to the requirements suggested
Usability/ Understandability - user must know how to use the system, that is the interface they interact with needs to be easy to use.
Reliability - Give more time to testing, so that the software is reliable

General Principles

Communication: We plan to have frequent face-to-face meetings (i.e. stand-up meetings)
Planning: We plan to involve stakeholders in all planning activities.

Modelling: The design team will strive to produce the simplest model that will describe the software.

Practice principles

Abstraction and modularity were the two software engineering principles central to the design of our web application. We had to look for and implement patterns for site architectures and the entire web application is an integration of modules (modules included navigation bars, a set of interlinked pages, and the different types of programming languages used).

E. Project Requirements

Volere Requirements

Project Drivers

1. Project Purpose: To build a home automation and security system in order to minimize the risk of the users' house being compromised in any way.
2. Stakeholders: The developers of the system (team 8), and the potential users

Project Constraints

1. Mandated constraints: The project must be completed within a month and must have incorporated all the feedback of the users. It must be traceable to user requirements.
2. Relevant facts and assumptions: The assumption is that the user owns a house or apartment. Another assumption is that the user wants or needs a home automation and security system.

Functional Requirements

1. System must offer authentication.
2. System must visualize sensor data.
3. System must have an alert feature.

Data Requirements

1. User first name, last name.
2. User password.

Non-functional requirements

Environmental Requirements

1. A good wifi connection.
2. A house/apartment (a place where the landlord will not have a problem with sensor installation)

Usability Requirements

1. System interface should be simple.
2. Interface should be easy to navigate and remember.
3. User-friendly interface, with no or minimal lag.

Project Issues

The main issue for the project would be the time constraint. Other issues are costs, the amount of money it will cost to purchase the equipment (i.e. Sensors, raspberry pi, cables, etc.).

Database

Entities:

1. User
2. Room
3. Sensor

Business Rules

1 User owns Many Rooms, at least one.
1 Room is owned by 1 User, exactly one.
1 Sensor is associated to 1 Room, exactly one.
1 Room has Many Sensors, at least 4.
1 Event is triggered by 1 Sensor, exactly one.
1 Sensor triggers Many Events, possibly none

Appendix A

Questionnaire

Consent: I agree that the data I provide will be used for any research purposes. Other researchers will have access to this information only if they agree to preserve the confidentiality of the information gathered from this questionnaire.

Brief Explanation: Home Automation and Security involves being able to control your home remotely and having minimal human intervention by introducing control systems. As an example lights could be turned on and off without even being inside your house. This also includes a monitoring service - a homeowner can be informed when an intruder enters a home with the use of a service such as the SMS.

Please fill in the following:

Full Name:
Occupation:
Age:
1.What type of home do you live in?
2. Do you live alone? Or with a roommate/roommates, wife and/or kids?
3. How many people are in the household?
4. Do all household members get home at the same time?
5. How frequently is the house left with nobody in it?
6. Do you own any pets?
7.Do you ever leave them alone in the house?
8. Do you consider yourself tech savvy(well versed with technology)?
9. What is your experience with using computers and the internet?
10. Do you have fast internet access(wifi) where you live?
11. Do you currently have a home automation and security system?
12. (if yes to question 4) What components (if any) of that system would you possibly wish to change?
13. (if yes to question 4) Are you comfortable with its interface(the way it looks)?
14. (if no) Give a description of your ideal home automation and security system.
15. Which components should the interface have?
16. Do you feel that online security is important?
17. For authentication, would you rather use your email and a password, or your phone number and a password? Or both?18. On a scale of 1-5 how much do you feel that you are at risk from online threats?
19. How do you want to access the system? and from where?

Assignment 2

Conceptual Models:

Conceptual model 1:

This model is basically a web application that has a navigation bar on the left to help you hop from one page to another. It makes use of icons for easy use and memorability, so that the user is able to know where to go and what to do.

- **metaphors and analogies:**
  - Using visual aids such as a logout icon for the logout button, a camera icon for the camera sensor, and the light bulb picture for the light sensor. There is a bell icon for notifications.
- **Interaction types:**
  - A drop down menu is located on the navigation bar to help move between rooms.
- **concepts**: user (with attributes: userId, username, password and type), room (with attributes: roomId, name, description, userId), sensor (with attributes: sensorId, name, description, type, roomId) and event (with attributes : eventId, type, data, sensorId).
- **relationships**:
  - A user has rooms.
  - Rooms have sensors.
  - Sensors trigger events.
- **mappings**:
  - Each sensor in the system corresponds to the physical sensor in the room.
  - Each room in the system corresponds to the physical room.
  - Each switch mechanism in the system corresponds to the physical switch, that can override that specific sensor.

Conceptual model 2:

A collection of linked pages, where the history of pages you have gone through to arrive

here is shown. Uses a side navigation bar.

- **metaphors and analogies:**
  - Using visual icons such as a house for home button, thermometer for temperature sensor and Light bulb for light sensor.
  - On and off switch mechanism on the interface is reminiscent of a physical switch that you can toggle on and off.
- **Interaction types:**

  - Drop down menu arrows, instructing command.
  - Feedback from buttons, such as colour change.

- **concepts**: user (with attributes: user-id, username, password and type), room (with attributes: room name, room-id), sensor (sensor id, sensor type, sensor name and description) and event (sensor id, is-enabled, sensor data)
- **relationships**:
  - A user has rooms.
  - Rooms have sensors.
  - Sensors trigger events.
- **mappings**:
  - Each sensor in the system corresponds to the physical sensor in the room.
  - Each room in the system corresponds to the physical room.
  - Each switch mechanism in the system corresponds to the physical switch, that can override the LDR sensor (by turning the light on or off manually)

Conceptual Model 3:

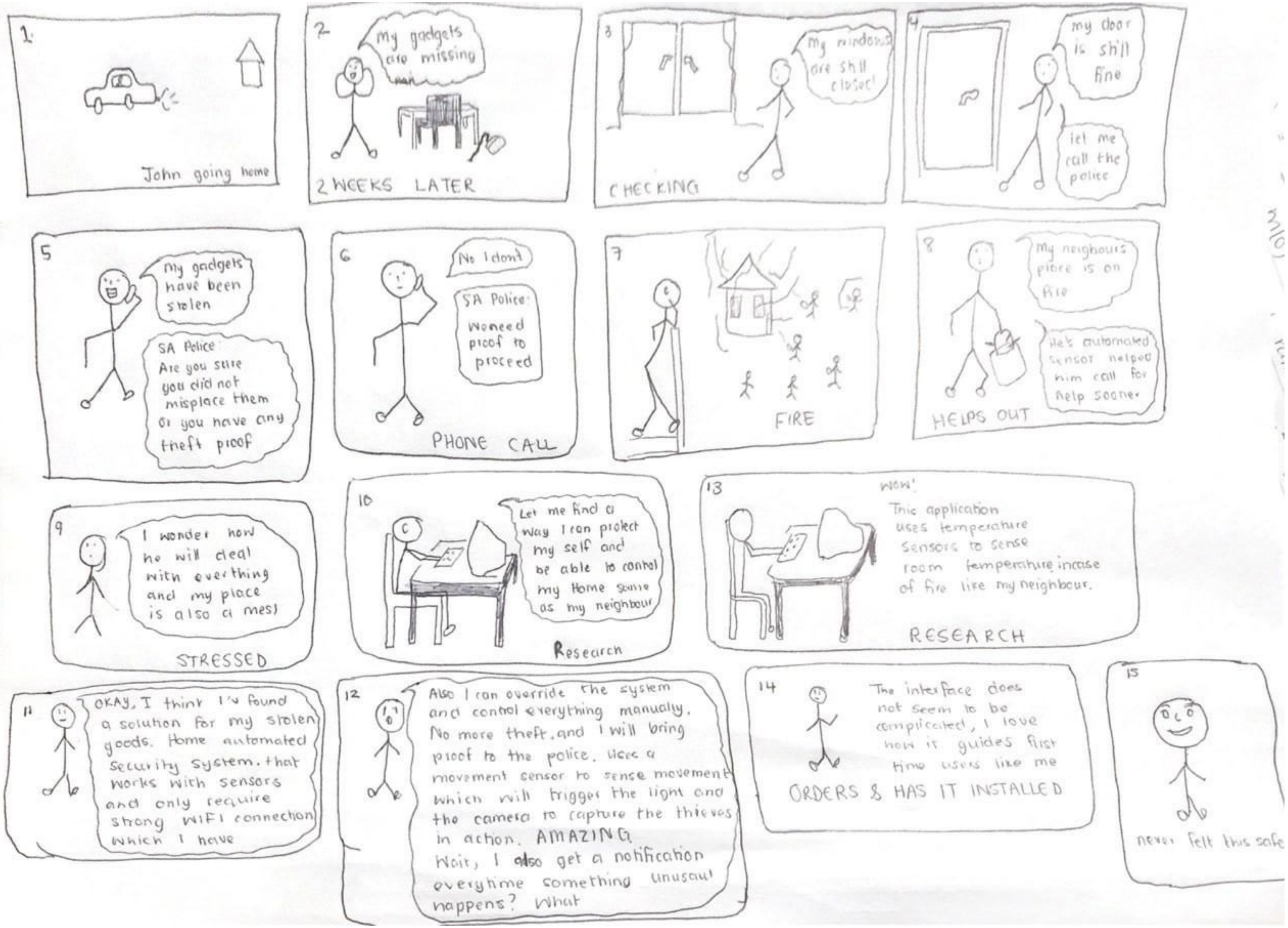
Also collection of linked pages, where the history of pages you have gone through to arrive here is shown. Also has a top navigation bar.

- **metaphors and analogies:**
  - On and off switch mechanism on the interface is reminiscent of a physical switch that you can toggle on and off.
- **Interaction types:**
  - Drop down menu arrows on the specific room for the admin side, instructing to add or delete the room.
  - Feedback from buttons, such as colour change.
- **concepts**: user (with attributes: user-id, username, password and type), room (with attributes: room name, room-id), sensor (sensor id, sensor type, sensor name and description) and event (sensor id, is-enabled, sensor data)
- **relationships**:

    - A user has rooms.
    - Rooms have sensors.
    - Sensors trigger events.
- **mappings**:
    - Each sensor in the system corresponds to the physical sensor in the room.
    - Each room in the system corresponds to the physical room.
    - Each switch mechanism in the system corresponds to the physical switch, that can override the LDR sensor (by turning the light on or off manually)

We are choosing the first the conceptual model based on simplicity, understandability and easy navigation. It is more user friendly.

Storyboard Model


1.
John going home
2
My gadgets are missing
2 WEEKS LATER
3
My windows are still closed
CHECKING
4
my door is still fine
let me call the police
5
My gadgets have been stolen
SA Police: Are you sure you did not misplace them or you have any theft proof
6
No I dont
SA Police: We need proof to proceed
PHONE CALL
7
FIRE
8
My neighours place is on fire
He's automated sensor helped him call for help sooner
HELPS OUT
9
I wonder how he will deal with everthing and my place is also a mess
STRESSED
10
Let me find a way I can protect my self and be able to control my Home same as my neighbour
Research
13
WOW!
This application uses temperature sensors to sense room temperature incase of fire like my neighbour.
RESEARCH
11
OKAY, I think I've found a solution for my stolen goods. Home automated security system. that works with sensors and only require strong WIFI connection which I have
12
Also I can override the system and control everything manually. No more theft, and I will bring proof to the police. Uses a movement sensor to sense movement which will trigger the light and the camera to capture the thieves in action. AMAZING
Wait, I also get a notification everytime something unusual happens? What
14
The interface does not seem to be complicated, I love how it guides first time users like me
ORDERS & HAS IT INSTALLED
15
never felt this safe

User Feedback:

“Does the system automatically call the helpline or should I?”

“Can I get the alert via sms? So that I don't have to log into the app to see the notifications”

Card Based Prototype

The card based prototype is attached again below the document for better viewing.

User Feedback:

“I like the yellow to indicate when light is on”.

“Calling the camera sensor part CCTV doesn't make sense to me. I would rather you call it CAMERA”.

“The system looks easy to use”.

“I would like some colour. It looks rather bland at the moment”

“I would like to be able to rename the room name instead of it being just 'room 1'”

“The system needs to provide some sort of notification”.

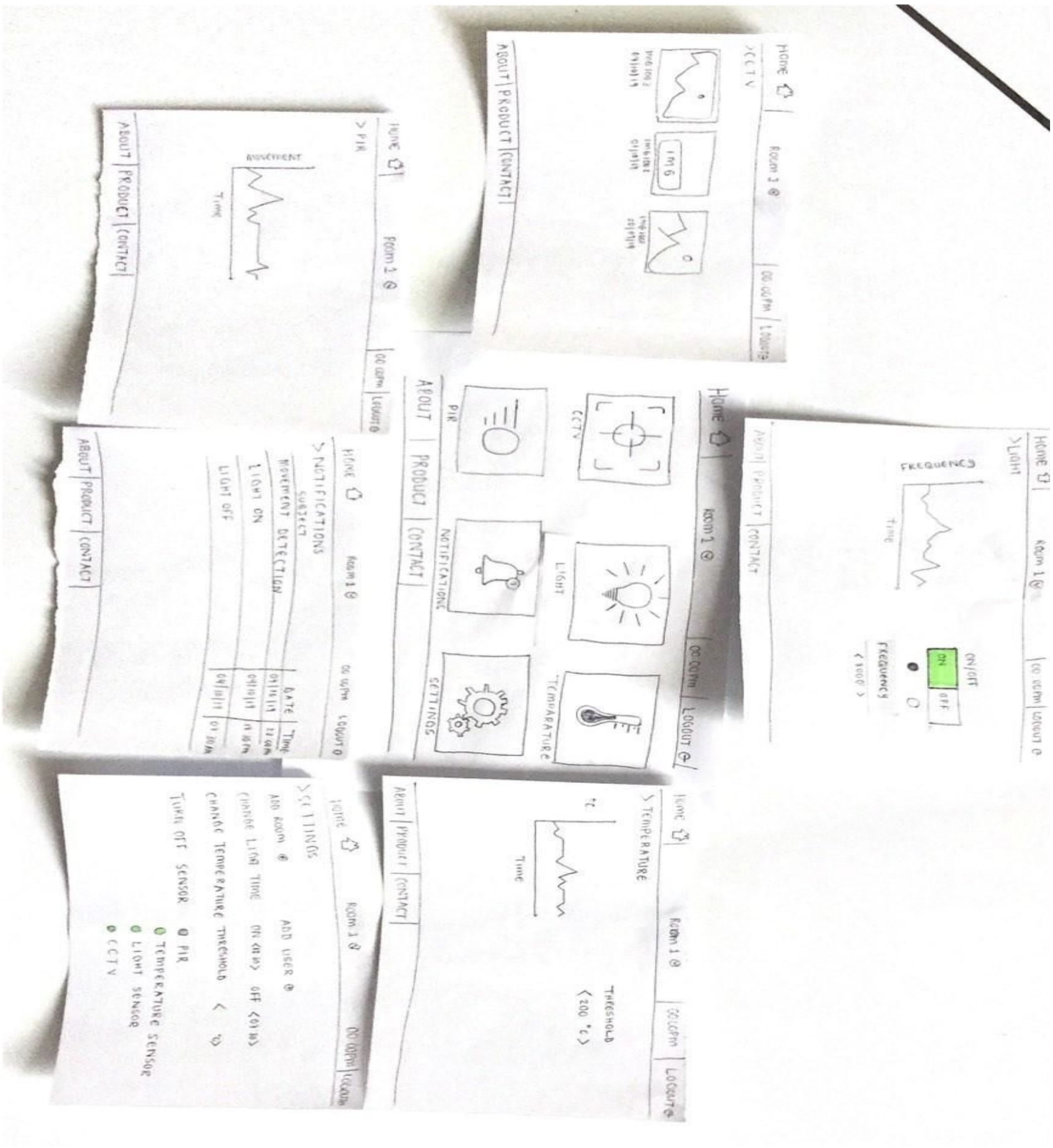
“I want to see real time data without any intervention”.

“I want to be able to download images taken by the camera”.

“I want to be able to see the history of events”.

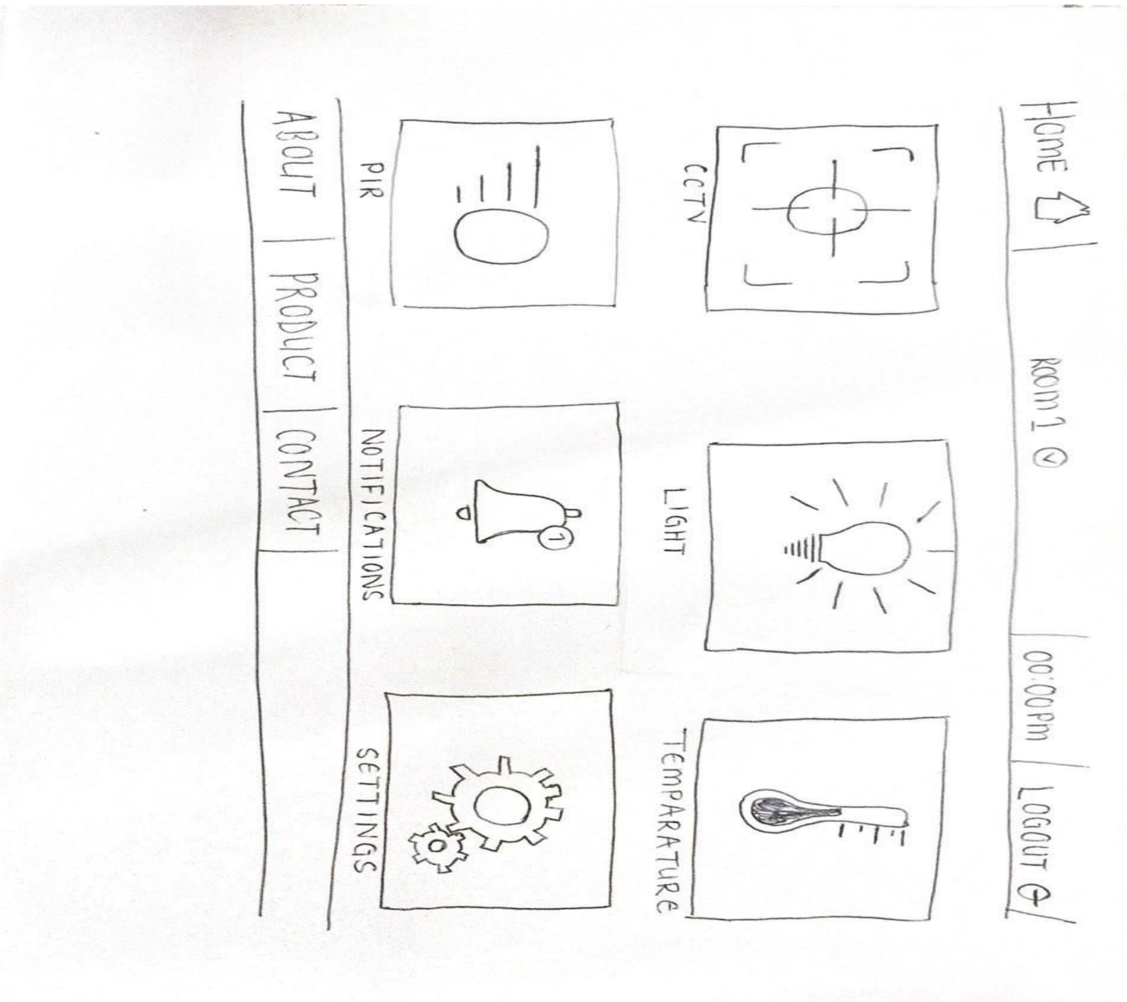
“Image should have date labels so that I can see when the image was taken”.

“Can different notifications have different colour”.



Sketch of Application Main Screen

The sketch is below



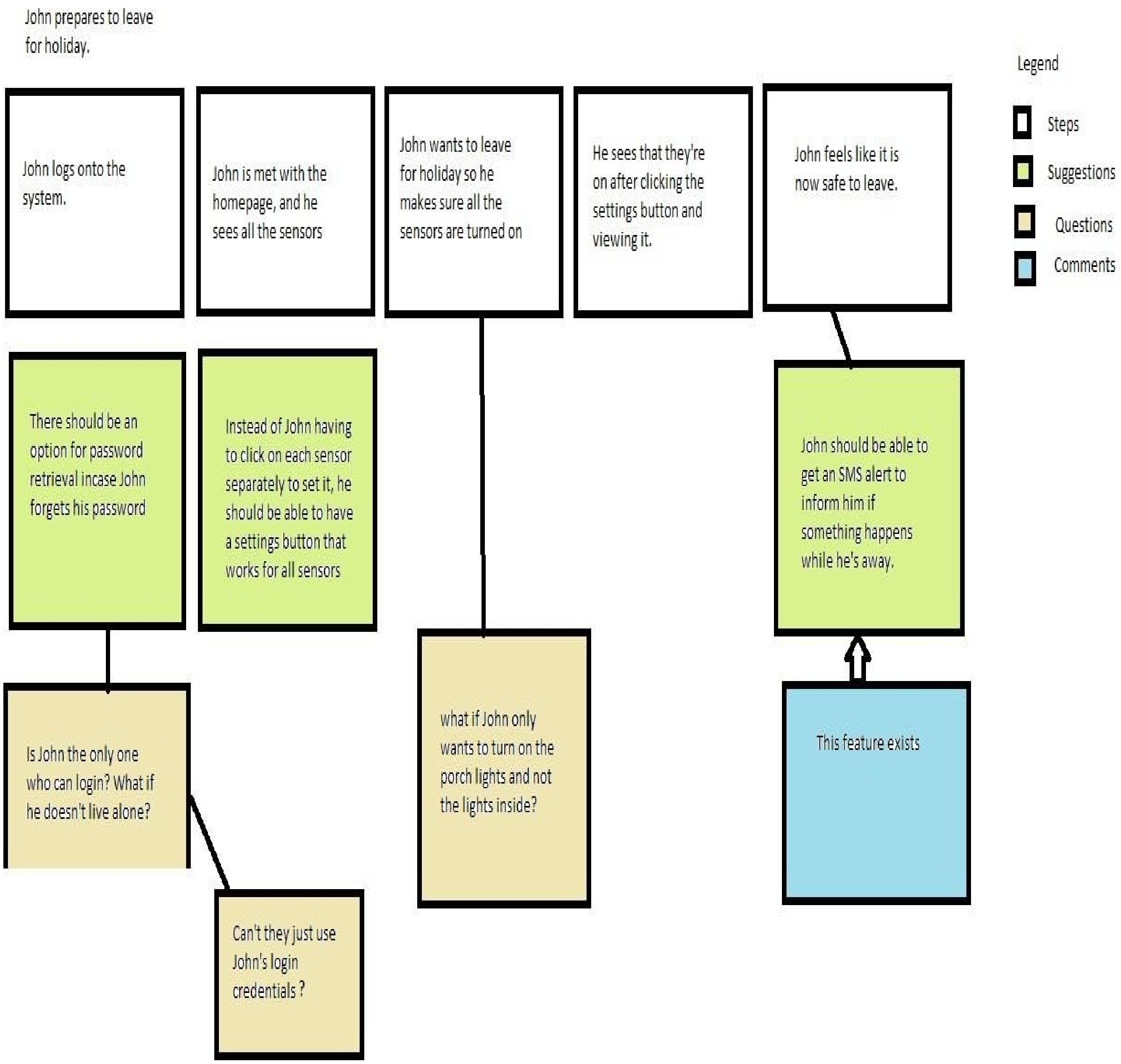
Where am I? : We decided it would work best if the user is met with the sensor icons at first glance. We particularly chose icons instead of just words because it is so much more aesthetically pleasing to see pictures instead of words.

What's here?: There are also words beneath the sensor icons to explicitly show which sensor is which and so the user knows exactly what is there, and when they click on that icon they know information about that particular sensor will show up.

Where do I go?: The icons also helps the user remember where to go for what very easily, when they see the thermometer icon they can immediately think “Oh, there's where I go for the temperature”

All of this makes it easier for the user to use the application and understand how it works.

Experience Map



John prepares to leave for holiday.

John logs onto the system.

John is met with the homepage, and he sees all the sensors

John wants to leave for holiday so he makes sure all the sensors are turned on

He sees that they're on after clicking the settings button and viewing it.

John feels like it is now safe to leave.

There should be an option for password retrieval incase John forgets his password

Instead of John having to click on each sensor separately to set it, he should be able to have a settings button that works for all sensors

John should be able to get an SMS alert to inform him if something happens while he's away.

Is John the only one who can login? What if he doesn't live alone?

Can't they just use John's login credentials ?

what if John only wants to turn on the porch lights and not the lights inside?

This feature exists

Legend

□ Steps

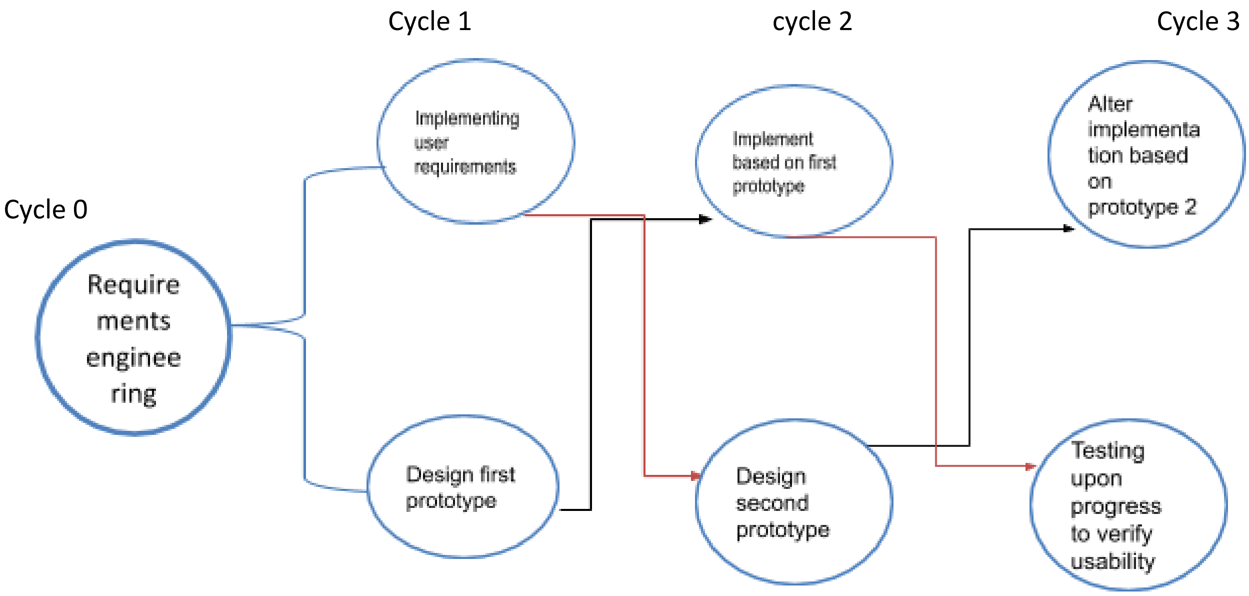
🟩 Suggestions

▨ Questions

🟦 Comments

Our System is similar to marker movement, since it is a web application that allows automation. The system automatically detect darkness and light and either switch on or off lights respectively. It can also detects high temperature when reaches 40°C and alert the user immediately via sms.

User Research: Usability Testing - We created Card-based prototype that helped to avoid walls of text, which can appear intimidating or time-consuming and allows users to dive deep into their interests quicker. Cards divide content into meaningful sections, similar to the way text paragraphs group sentences into distinct sections. They can gather various pieces of information to form one coherent piece of content in a controlled environment.
Cycles:


Cycle 1
cycle 2
Cycle 3
Cycle 0
Requirements engineering
Implementing user requirements
Design first prototype
Implement based on first prototype
Design second prototype
Alter implementation based on prototype 2
Testing upon progress to verify usability

Link to mockup of homepage: https://drive.google.com/open?id=1bnGEM9AZwMQ8PZeWQ0Ef5hkKvvUKqqm8

Link to modified prototype:

https://drive.google.com/drive/folders/1BFP8iv6r6h-eOV8GyB7gF-7vl68GvB5O?usp=sharing

Dashboard interface provides critical, pertinent data at a glance.

a) Assistant dialogue - A user is using the interface for the first time. He/She does not know what is where on the interface. A see-through layer will be positioned over the screen to visually explain how to use the interface. This layer will be present only the first time the user uses the system.

Accelerator - If too much information is presented at once, users might suffer from analysis paralysis. Some content will be hidden by default. Users who are interested in viewing this content will have to click on accelerators (button or other widgets) to see it. This keeps the interface as simple as possible.

One-hand design- Users do not want to use their whole body to interact with the system. They would rather multi-task.

A one-hand design will be used for the mobile version of the interface. This design will enable users to interact with the system without repositioning the holding hand or using a second one.

- We select dark-pattern for our project since it can market our product swiftly.

Database Side

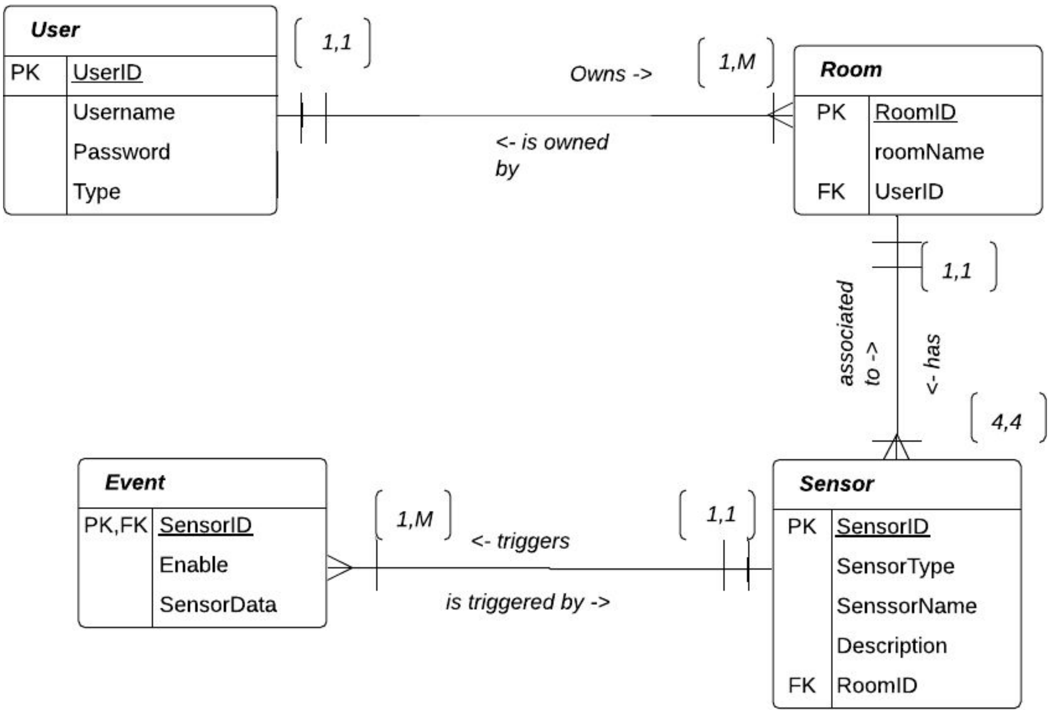
Entities

1. User
2. Room
3. Sensor
4. Event

Business Rules

- 1 User owns Many Rooms, at least one
- 1 Room is owned by 1 User, exactly one
- 1 Sensor is associated to 1 Room, exactly one
- 1 Room has Many Sensors, exactly four
- 1 Event is triggered by 1 Sensor, exactly one
- 1 Sensor triggers Many Events, at least one

Entity Relationship Diagram (ERD)



Assignment 3

Standard Tasks:

- Viewing room current temperature status - a user can view room temperature on the dashboard.
- Viewing room current light status - a user can view room light status(On/Off) on the dashboard.
- View room camera images - a user can view images taken by the camera on a particular room.
- View notifications - a user can click on the navigation bar on the left of the web application and go to notifications to view the alerts they have gotten.
- Download camera images - a user can download images taken by the camera.

Informed Consent Form

Brief Explanation: Home Automation and Security involves being able to control your home remotely and having minimal human intervention by introducing control systems. As an example lights could be turned on and off without even being inside your house. If an intruder were to get into your house, you would get a notification of that instance as well as a picture of said intruder, and you would be able to get help.

We would like you to test out the provided prototype of the system to see if you can perform certain tasks and give feedback.
I ,the undersigned, hereby declare that:

	Yes	**No**
I voluntarily agree to participate in the research.		
I have understood the information about the project.		
I understand that I will not get paid to participate in the research.		
I understand that participating in this study poses no threat to my well being.		
I consent to having my videos and or recordings taken during the study.		
I understand I have the right to refuse to participate in the study should I choose to do so in the process of it with no penalty whatsoever.		
I understand that other researchers will have access to this information only if they agree to preserve the confidentiality of the information gathered.		
I would like to use my name and understand that what I have said will be used in reports, and possibly other research outputs.		

Provide your name here if you answered yes to the last question:
Name:

Researchers are bound by a code of ethics that includes the following protections for subjects:

1. Protected from physical or psychological harm (including loss of dignity, loss of autonomy, and loss of self-esteem).
2. Protection of privacy and confidentiality.
3. Protection against unjustifiable deception.
4. The subject must give voluntary informed consent to participate in research.
5. Guardians must give consent for minors to participate.
6. Minors over age 13 (the age may vary) must also give their consent to participate.

Signature:

Selecting users

We gathered three users, all from the University of the Western Cape at the Computation and Mathematical Science Building in the postgraduate lounge and asked them if they could perform 4 standard tasks using the prototype of our system. The users were timed on how long it took them to finish each task. The time it took the users to finish each individual standard tasks was recorded and later analyse further. We had them read and sign the consent form before participating. The study took place in the university premises.

Users

Standard task	Likes	Dislikes	Timing	Number of errors
Viewing room current temperature status - a user can view room temperature on the dashboard.	User 1: I can view digital readings. User 2: No problem viewing User 3: I like the way the reading is displayed.	User 1: I need to see different colors to show different temperature levels.	User 1: 2 seconds. User 2: 3 seconds User 3: 4 seconds	User 3: 1 error. The user was not familiar with the way the temperature readings was display.
Viewing room current light status - a user can view room light status(On/Off) on the dashboard.	User 1: Different images are used to display different light state. One image for an ON state ,and another image for an OFF state. User 2: I like the light flickering from on to off. User3: The changing states of the light shown is a great feature.	User 2: A light is not a critical component. It should take less space on the interface.	User1:5 seconds User2: 5 seconds USer3: 4.5 seconds	0 error. Users were satisfied with the components we used to display the light functionality.
View room camera images - a user can view images taken by the camera on a particular room.	User 3: Images have different captions that show when the image was taken.	User2: Images are very big.	7 seconds. It takes longer since many images have to be loaded from the database.	1 error. One user had problem displaying the images.
Download camera images - a user can download images taken by the camera.	The location of the download icon was perfect.	User 1: The download icon did not provide feedback.	There was no actual feedback from clicking download button.	1 error. A user had difficulties in using the download button.

Field or controlled study

We are planning to conduct a field study. The reason we chose this and not the controlled study is because our application will mostly be used under natural settings, that means we have to ensure it is suitable for the purpose it was intended for. We are going to use a group of four members for the field study. Our system has four sensors, that is the reason why we chose four users. We want to observe how these users interact with each sensor and we take note on the problems they encountered while using the system. Later in the study, the participants will be asked to comment on the experience of using the system. Getting more users to participate is impractical considering the time we had to finish the project.

Planning: Approach different users instead of collecting them into one room, to avoid making them extremely uncomfortable. Ask for user consent, read to them or give them the consent form to read, understand and sign. Ask for their permission before we take videos or recordings so that they can either agree or disagree to it. This is to avoid violating the rights of the participants. Decide what recording technique we will use, either to go traditional(handwritten/typed out notes) or use videos/ audio recordings. We would most likely go the traditional route because it's easier and no hassle to set up.

Things to take into account: Whether or not the user is comfortable, whether or not they understand the study they are participating in. We would also need to make sure that neither we or our equipment is getting in the way of the user.

Data collected: The users interacting with the system(videos, if consent is provided), and feedback about their experience of using the system. This information would be summarised and presented in form of a report.

Benefits and Disadvantages of type of study

Type of study	Benefits	Disadvantages
Controlled study	1. We can test specific system components. We can make sure that key system components are working and they meet requirements.	1. We can not determine how the system will perform after it is released to the users.
Field study	1. We can see how users will use the system in the real world. 2. This study will provide useful insight. We can uncover features that are impractical.	1. This study is difficult to conduct if we are using too many participants. 2. There are too many uncontrolled variables, and as a result the study might not generate accurate data.

Application of Software Engineering

(UML class diagram attached below document: Appendix B)

Principles of SE

- **Abstraction** – reduces complexity such as the user having to interact with the web application user interface and not see the complexity behind with sensors.
- **Consistency** – familiarity with the functions, type of icons used and the position of functions such as the logout page.
- **Hiding** – a secondary user has functions they cannot access for security reasons such as adding another user, only an admin can do so.

Umbrella Activities:

- **Project Tracking** - we using an online communication tool and we have regular meetings to track the progress of the project and keep up to date.
- **Technical Reviews** - technical issues that / may arise are discussed in the meetings we conduct, and solutions to those problems/issues are discussed.
- **Work Product Preparation and Production** – we evaluated the feedback that we received in the previous week and made changes to our product accordingly. For this part of the project we had to rebuild the prototype taking into consideration a heuristic evaluation that we did.

Goals:

- **Understandability** – the interface is easy to interact with, it makes use of icons with easily readable labels to make it easier for the user to interact with the system. The user doesn't have to try and remember where everything is located or what performs what function because it is clear and easy to see.
- **Portability** – the system is portable because it can be accessed anywhere since it's web-based.
- **Security** – The user uses the login details, password and email to access the system.
- **Process** – Agile Process, the team is self-organized, cross-functional and we have a collaborative effort with end-users in the sense that we ask for their input. We make use of the agile process because in each week, we iterate on the previous week's work instead of delivering software all at once. The work is done in short 1 week cycles, where each week we have the work split up and each person does a certain task and works with someone else in the team based on how their tasks relate to each other. The development is user-centric in the sense that we involve users in each step of the development process to check whether or not the product meets the user requirements. We make an effort as a team to meet each week (and take meeting minutes) and discuss the development process and check in where we are in terms of developing the product.

Database

Entities

1. User
2. Room
3. Sensor
4. Event

Business Rules

1 User owns Many Rooms, at least one
1 Room is owned by 1 User, exactly one
1 Sensor is associated to 1 Room, exactly one
1 Room has Many Sensors, at least 4
1 Event is triggered by 1 Sensor, exactly one
1 Sensor triggers Many Events, at least one

Assignment 4

Nielsen's Heuristics

We chose Jakob Nielsen's heuristics because they are the most-used usability heuristics for user interface design, broad rules of thumb and not specific usability guidelines.

Appropriate set of heuristics:

1. Visibility of system status – user being informed about what is going on through appropriate feedback, such as real time data on display.

2. Match between system and the real world - system should speak the user's common language, with words, phrases and concepts familiar to the user.

3. User control and freedom - "emergency exit" to leave the unwanted state without having to go through an extended dialogue, such as back button.

4. Consistency and standards - users should not have to wonder whether different words, situations, or actions mean the same thing or where to find certain things such the menu bar or logout button.

5. Error prevention - eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

6. Recognition rather than recall - minimize the user's memory load by making objects, actions, and options visible such as using a light icon instead of LDR.

7. Aesthetic and minimalist design – web pages should not contain information, which is irrelevant or rarely needed, must focus on the main concept.

8. Help and documentation – Any documentation such as information should be easy to search, focused on the user's task, list concrete steps to be carried out and what the system is about.

9. Help users recognize, diagnose, and recover from errors - Error messages should be expressed in plain language(no codes), precisely indicate the problem, precisely indicate the problem, and constructively suggest a solution.

10. Flexibility and efficiency of use – The system needs to cater to both experienced and novice users

Evaluation based on our prototype:

We conducted a heuristic evaluation with the experts in team 10, because of time constraints we ended up conducting the evaluation with just those experts.

Evaluation Process:

Briefing: First we briefed our experts on the system so that they know the basic functionality of the web application, and know what to cover during the evaluation.

First evaluation phase: The evaluators used the product freely to get a feel of it, and then identified the specific evaluation elements they want to evaluate. In this case, the evaluators decided to evaluate every aspect of the web application.

Second evaluation phase: The evaluators carried out another evaluation but this time they applied the heuristics on the aspects they decided to pick in the first evaluation, and evaluated how well the heuristics fit the overall design.

Recording: The experts recorded the problems they encountered when they interacted with the system.

Debriefing session: We then sat down with the evaluators and discussed the problems they encountered and what we could possibly do to fix them.

1. Visibility of system status – experts liked how they are informed about what is going on in the system, for example the temperature updates regarding the temperature sensor are displayed.
   Experts expressed that they would rather like to see the temperature readings at first glance instead of having to click the temperature sensor icon and then seeing the readings.

2. Match between system and the real world - system speaks the users' language, with words, phrases and concepts familiar to the user such as using light sensor instead of LDR.

3. User control and freedom - system has an emergency exit option which the back button/option in a web page linked to the homepage.

4. Consistency and standards - system is not consistent in terms of positioning functions in familiar positioning and naming of common items such as alerts instead of notifications. Suggestion would be to change the name 'alerts' to 'notifications'.

5. Error prevention - no error prevention messages. A suggestion would be to add an option asking the user if they want to actually logout when they click the logout button.
   Another suggestion would be to ask the user if they actually want to add a room, or even change settings when they attempt to change them.

6. Recognition rather than recall - there is recognition in terms of icons used, easy to remember the information on the web page.

7. Aesthetic and minimalist design – the system is very minimal, use of icons and less texts.

8. Help and documentation – One expert suggested that the system does not need documentation because everything is easy to read and see.
   Other experts suggested the system lacks documentation to guide a novice user, and that even a brief explanation of what is what and what the system does would be helpful.

9. Help users recognize, diagnose, and recover from errors – the system currently has no error messages.

10. Flexibility and efficiency of use – the system is easy to understand but lacks the proper documentation to guide a first time user as to what it is actually about.
    A suggestion would be to create an about page to briefly introduce the components of the system accordingly.

Prototype

The new prototype does satisfy user requirements such as having documentation that explains how the system works, user getting a feedback on tasks performed and the Nielsen's heuristics that are appropriate for the system.

https://xd.adobe.com/view/1160e1db-ad3b-4b9e-5c04-de880cb15439-6751

First prototype:

https://drive.google.com/drive/folders/1BFP8iv6r6h-eOV8GyB7gF-7vl68GvB5O?usp=sharing

Changes

Prototype 1	Prototype 2
Had no documentation to better assist beginner users to know what to do	Added documentation to assist beginner users to know what to do/ what the web-application is about
No error prevention messages	Added error prevention messages
Interface had a lot to look at at first glance	Interface is simpler, less cluttered

Usability vs Heuristic evaluation

- In the previous evaluation the user encountered a problem of not knowing how to toggle the lights on and off, not knowing the task to perform. The heuristic evaluation agreed that the user couldn't perform that particular task because there was no documentation to guide the user.

- Users encountered a problem of not having more options with opening the images or the home button leading nowhere.

  The heuristic evaluation yielded the same findings that there are some buttons that lead nowhere when clicked, but it is not explicitly shown that you cannot click these buttons. The experts that did the evaluation decided that instead of the user being frustrated clicking a button that leads nowhere, they should be made aware that a certain option isn't click-able when it isn't.

- The user encountered the problem of not being warned about an irreversible action for example if they accidentally click on logout before they intend to actually logout. The experts who did the heuristics evaluation agreed that this was true and gave the suggestion to include error prevention messages, and this idea was implemented in the final product.

While the heuristics evaluation involves getting input from experts and that seems a bit more comprehensive and would seem like the better option to go for because the input is from experts, we prefer the usability evaluation because the system is user-centric and since the users are going to actually use the system, they are better suited to say what they do and don't like about the system.

Evaluation Approaches

Evaluation approaches			
	Usability Testing	***Field Studies***	***Analytical Evaluation***
Findings	Actionable findings to improve our system for example one user suggested the use of a big font size.	Learn the unexpected by observing users in their natural environment e.g. how much user is experienced in using a computer.	The findings on interface compliance with recognized usability principles (Heuristics).
Benefits	Full control of users.Time efficiency.Provided unbiased examination of our product.	It yields very detailed collection of data. Realistic expectations Allow system to be tested under realistic conditions.	It provides expert guide that will assist in improving the final product.
Costs	R0, 00 (Since we use prototype to conduct the analysis and no lab is hired).	It cost lots of time (More than a day).	Costs will include the cost of putting together the hardware, that would be roughly R900
Limitations	Only small sample of potential users can undertake. Users have different preferences and opinion about the product. E.g. some users would like see everything at once while others would like to see few items on the dashboard and browse menu for more items.	It is up to the participants to decide how to use the product and when. As a result feedback is not received immediately	10 Heuristics for User Interface Design.

Application of Software Engineering:

(UML class diagram attached below document: Appendix B)

Principles:

- Abstraction – reduces complexity such as the user having to interact with the web-application user interface and not see the complexity behind with sensors.

- Consistency – familiarity with the functions, type of icons used and the position of functions such as the logout page.

- Hiding – a secondary user has functions they cannot access for security reasons such as adding another user, only an admin can do so.

Umbrella Activities:

- Project Tracking - we using an online communication tool and we have regular meetings to track the progress of the project and keep up to date.

- Technical Reviews - technical issues that / may arise are discussed in the meetings we conduct, and solutions to those problems/issues are discussed.

- Work Product Preparation and Production – we evaluated the feedback that we received in the previous week and made changes to our product accordingly. For this part of the project we had to rebuild the prototype taking into consideration a heuristic evaluation that we did.

Goals:

- Understandability – the interface is easy to interact with, it makes use of icons with easily readable labels to make it easier for the user to interact with the system. The user doesn't have to try and remember where everything is located or what performs what function because it is clear and easy to see.

- Portability – the system is portable because it can be accessed anywhere since it's web-based.

- Security – The user uses the login details, password and email to access the system.

Process – Agile Process, the team is self-organized, cross-functional and we have a collaborative effort with end-users in the sense that we ask for their input.

We make use of the agile process because in each week, we iterate on the previous week's work instead of delivering software all at once. The work is done in short 1 week cycles, where each week we have the work split up and each person does a certain task and works with someone else in the team based on how their tasks relate to each other. The development is user-centric in the sense that we involve users in each step of the development process to check whether or not the product meets the user requirements. We make an effort as a team to meet each week (and take meeting minutes) and discuss the development process and check in where we are in terms of developing the product

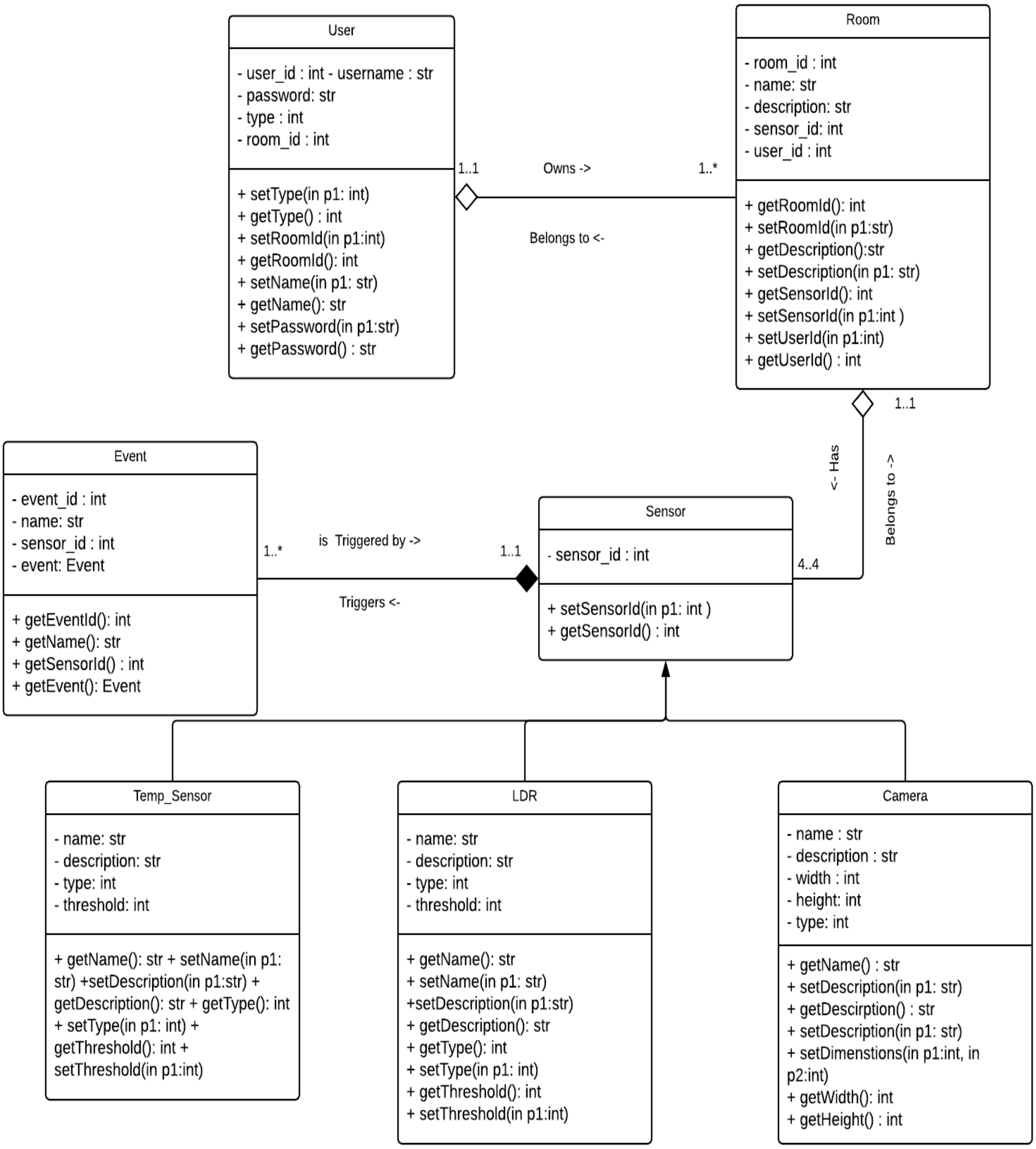
Database Internal Model

```
-- ************************************** `User`
CREATE TABLE `User`
(
`userID` int NOT NULL ,
`username` varchar(40) NOT NULL ,
`userPassword` varchar(20) NOT NULL ,
`userType` varchar(45) NULL ,
PRIMARY KEY (`userID`)
);
-- ************************************** `Room`
CREATE TABLE `Room`
(
`roomID` int NOT NULL,
`roomName` varchar(45) NOT NULL ,
`userID` int NOT NULL ,
PRIMARY KEY (`roomID`),
FOREIGN KEY (`UserID`) REFERENCES `User` (`userID`)
);
-- ************************************** `Sensor`
CREATE TABLE `Sensor`
(
`sensorID` int NOT NULL ,
`sensorType` varchar(45) NOT NULL ,
`sensorName` varchar(45) NOT NULL ,
`sensorDescrip` varchar(45) NOT NULL ,
`roomID` int NOT NULL ,
PRIMARY KEY (`sensorID`),
FOREIGN KEY (`roomID`) REFERENCES `Room` (`roomID`)

);

-- ************************************** `Event`

CREATE TABLE `Event`

(

`eventID` int NOT NULL,

`eventType` varchar(45) NOT NULL DEFAULT 0 ,

`sensorID` int NOT NULL ,

PRIMARY KEY (`eventID`),

FOREIGN KEY (`sensorID`) REFERENCES `Sensor` (`sensorID`)

);
```

Links

Link to demo video: https://drive.google.com/file/d/1RNEJBa3i57td5icab23-ZpAEdCqz4wsR/view?usp=sharing

Appendix B


User
- user_id : int - username : str
- password: str
- type : int
- room_id : int
+ setType(in p1: int)
+ getType() : int
+ setRoomId(in p1:int)
+ getRoomId(): int
+ setName(in p1: str)
+ getName(): str
+ setPassword(in p1:str)
+ getPassword() : str
1..1
Owns ->
1..*
Belongs to <-
Room
- room_id : int
- name: str
- description: str
- sensor_id: int
- user_id : int
+ getRoomId(): int
+ setRoomId(in p1:str)
+ getDescription():str
+ setDescription(in p1: str)
+ getSensorId(): int
+ setSensorId(in p1:int)
+ setUserId(in p1:int)
+ getUserId() : int
1..1
<- Has
Belongs to ->
4..4
Event
- event_id : int
- name: str
- sensor_id : int
- event: Event
+ getEventId(): int
+ getName(): str
+ getSensorId() : int
+ getEvent(): Event
1..*
is Triggered by ->
1..1
Triggers <-
Sensor
- sensor_id : int
+ setSensorId(in p1: int)
+ getSensorId() : int
Temp_Sensor
- name: str
- description: str
- type: int
- threshold: int
+ getName(): str + setName(in p1: str) +setDescription(in p1:str) + getDescription(): str + getType(): int + setType(in p1: int) + getThreshold(): int + setThreshold(in p1:int)
LDR
- name: str
- description: str
- type: int
- threshold: int
+ getName(): str
+ setName(in p1: str)
+setDescription(in p1:str)
+ getDescription(): str
+ getType(): int
+ setType(in p1: int)
+ getThreshold(): int
+ setThreshold(in p1:int)
Camera
- name : str
- description : str
- width : int
- height: int
- type: int
+ getName() : str
+ setDescription(in p1: str)
+ getDescirption() : str
+ setDescription(in p1: str)
+ setDimenstions(in p1:int, in p2:int)
+ getWidth(): int
+ getHeight() : int