User Experience Design Document

Project name: TASMU Robotics
Project team members: Muneera Al-Baker, Anastasia Cheypesh, Ahmad Mohamad

Date: 30/3/2023

Overview:

Our healthcare worker ECG monitoring application is designed to improve the health and wellbeing of healthcare workers by providing real-time monitoring of their stress levels. The application uses a robot that has camera sensors to capture ECG signs, which are then analyzed using machine learning algorithms to identify signs of stress. The data is displayed on a web application dashboard, allowing Hamad Medical Corporation (HMC) administrators to monitor the stress levels of their healthcare workers and take action to mitigate stress when necessary.

User Goals:

The primary goal of HMC administrator's is to use the application to monitor their healthcare workers stress levels and take action to manage stress proactively and to improve employee health and wellbeing.

User Persona:

1. **Healthcare workers:** Our primary user persona is a healthcare worker who is responsible for patient care and may experience high levels of stress due to long work hours, high workload, and challenging patient situations.

2. **Admin:** Our second primary user persona is a hospital administrator who is responsible for managing employee health and wellbeing.

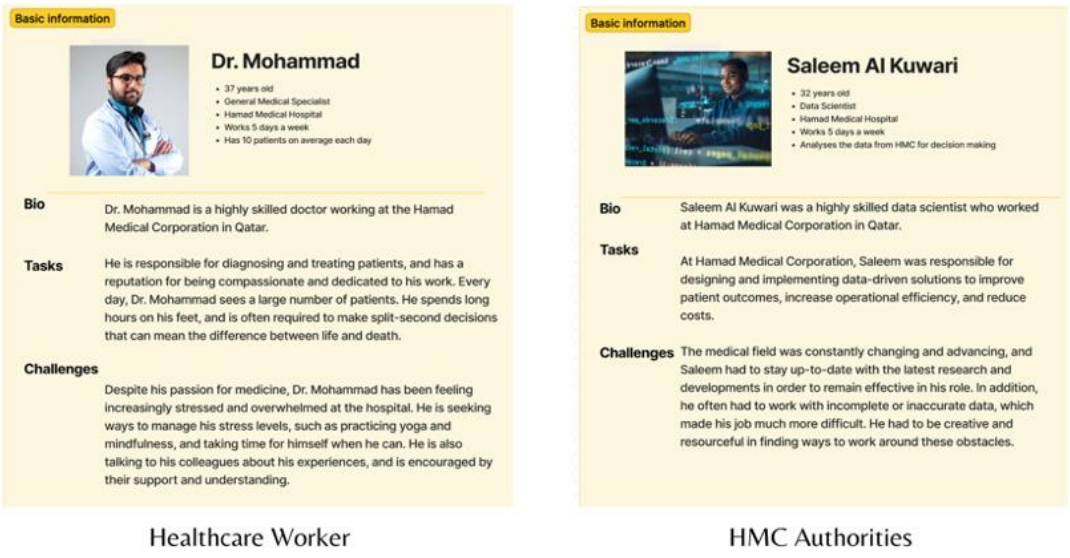


Figure 1. Personas

Design Principles:

1. User-Centered Design: The application is designed with the user in mind, taking into account their needs, preferences, and workflow.
2. Simplicity: The application is designed to be simple and easy to use and monitor stress levels, with minimal steps required to capture ECG data in a very short time.
3. Accessibility: The application is designed to be accessible to users with different levels of technical skills.
4. Visual Hierarchy: The application dashboard uses a clear visual hierarchy to prioritize important information and make it easy for hospital administrators to identify healthcare workers experiencing high levels of stress.

Design Elements:

1. Login Screen: The application has a login screen that requires healthcare workers and hospital administrators to enter their credentials to access the dashboard.
2. Home Page: The home page includes the latest news and events for the hospital.
3. Dashboard: The dashboard displays a list of all healthcare workers including (first name, last name, test date, test time, result) and a real-time graph of ECG data captured by the sensors, with a visual indicator of the user's stress situation.
4. Robot dashboard: This dashboard displays a list of all robots including (robot id, room number, sensor Id and robot status).
5. Log out: The application has a logout button to exit.

Design Process:

1. Research: We have conducted user research to understand the needs, preferences, and workflows of healthcare workers and hospital administrators.
2. Entity Relational Diagram (ERD): We have created an ERD to understand the relationships between the different entities of our project.
3. Architecture Design Pattern: We have chosen the suitable architecture patterns for our project based on our research.
4. Wireframing: We have developed two types of wireframes (low and high) to visualize the application's structure and layout.
5. Site Map: We have created a site map for our application to visualize the different pages of the application.
6. Prototyping: We have created prototypes of the application to test usability and identify areas for improvement.
7. Usability Testing and User Acceptance test: We have conducted usability testing and user acceptance tests with different end users to gather feedback and improve the application's design.
8. Iteration: We have used feedback from usability testing to refine the application's design and improve user experience.

Architecture Design Patterns:

Based on the complexity of our proposed solution, and after the discussion we had with our teaching team, we have studied all design architectural patterns and decided to use more than one architectural pattern, each for a different reason.

1) Layered Pattern:

Our main advantage to using the layered pattern is that it can improve the modularity, maintainability, and scalability of our system by dividing it into smaller and more manageable components.

This pattern applies to our project since our solution consists of different tasks that can be divided into different layers such as: the sensor's readings of heart activity, the data algorithm and stress levels assessment, and the robot behavior and interaction.

Therefore, we decided to have a layered pattern that has the following layers:

- Presentation layer: contains the presentation of the resultant stress levels of healthcare workers for HMC authorities. As well as the presentation of the robot and its interaction with the user.
- The business/logic layer: contains all the logic that is required by the software application to meet its functional requirements and to process the collected data through a set of algorithms.
- Persistence layer: it is where retrievable information about healthcare workers will be stored.

2) **Client-Server Pattern**

The client component makes requests to the server component, which processes the request and returns a response. The main advantage of using the client-server pattern is that it can provide a clear separation of concerns between the client and the server, making it easier to maintain and update each component independently.

Since client-server architecture pattern is used when a client requests and receives service from a centralized server, in our case, HMC authorities are the clients who are requesting information about their healthcare workers and should be receiving the resultant stress levels.

3) **Blackboard Pattern**

The main advantage of using the Blackboard pattern is that it can provide a flexible and scalable approach to solving complex problems by dividing the problem into smaller subproblems, which can be solved independently by different components.

However, our project deals with artificial intelligence and machine learning, and the blackboard pattern is mostly used for this type of projects.

Entity Relational Diagram (ERD)

We have created an Entity Relational diagram that shows how each entity communicates with another and what the relationships are between them.

The entities are as follow:

1- Robots: Our main device that will be interacting with the healthcare worker

2- Healthcare workers: Our main users who will be interacting with the robots

3- Departments: Each healthcare worker belongs to a certain department

4- sensors: These are the sensors that will be attached to the robot. Different sensors are attached to different robots.

5- ECG signs: These sensors are the collected signs of user's heart activity

6- Stress levels: These are the resultant stress levels

7- Rooms: These are the rooms that the robots are assigned to

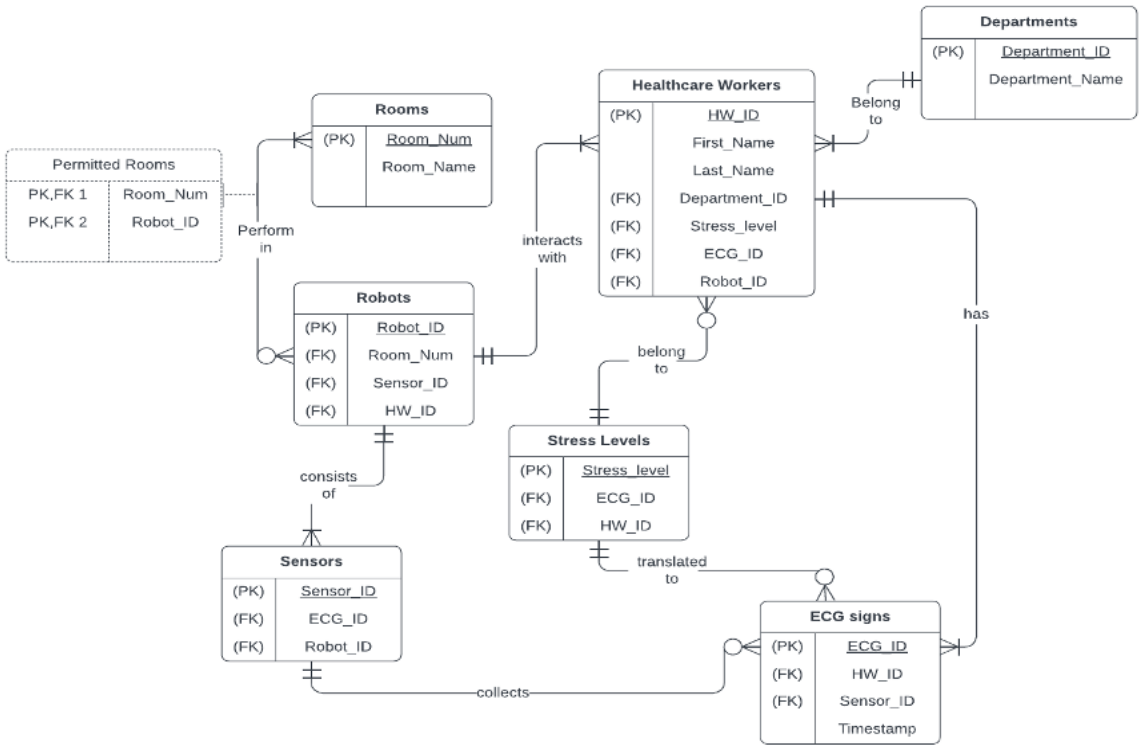


Figure 2. Entity Relationship Diagram

The relations between entities go as follows:

- Each **Healthcare worker** belongs to one department, each **Department** can have one or many healthcare workers.
- Each **Healthcare worker** interacts with one **Robot** (to reduce redundant data about the user), each robot can interact with one or many Healthcare workers.
- Each **Robot** can perform in one or many rooms, each **Room** can have zero or many robots.
- Each **Robot** consists of one or many sensors, each **Sensor** can be attached to one and only one robot.
- Each **Sensor** can collect zero or many ECG signs, each **ECG sign** can be collected from one sensor.
- Each **ECG sign** can be translated to one stress level, each **Stress level** can reflect zero or many ECG signs.
- Each **ECG sign** can belong to one and only one Healthcare worker, each **Healthcare worker** can have one or multiple ECG signs.
- Each **Healthcare worker** can belong to one stress level, each **Stress level** can have zero or many healthcare workers.

Wireframing:

1. UI wireframes and workflows (low fidelity)

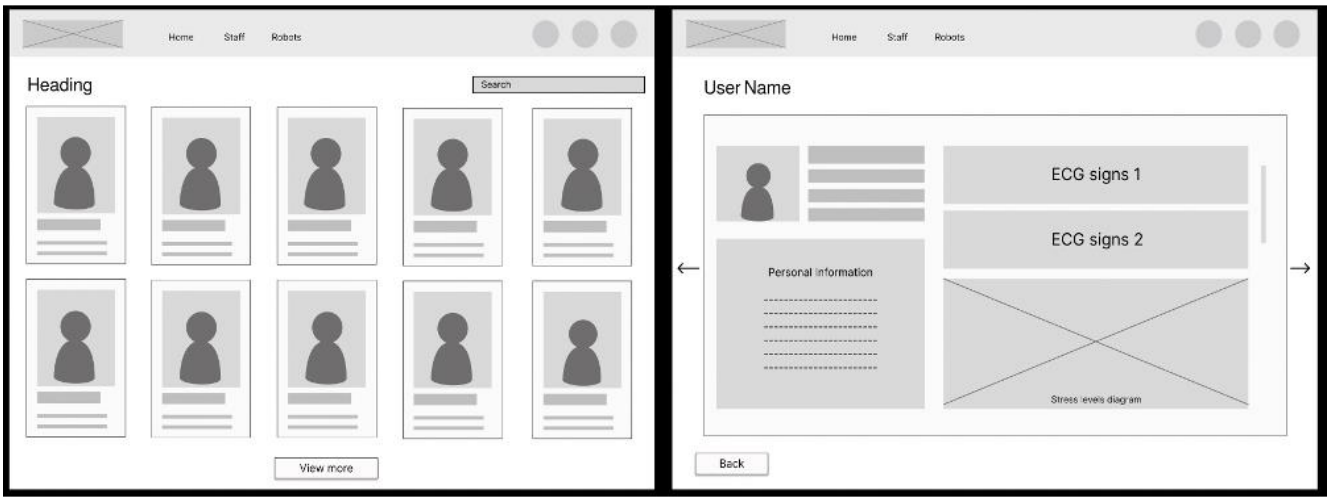


Figure 3. Low fidelity workflows

2. UI wireframes and wire flows (high-fidelity)

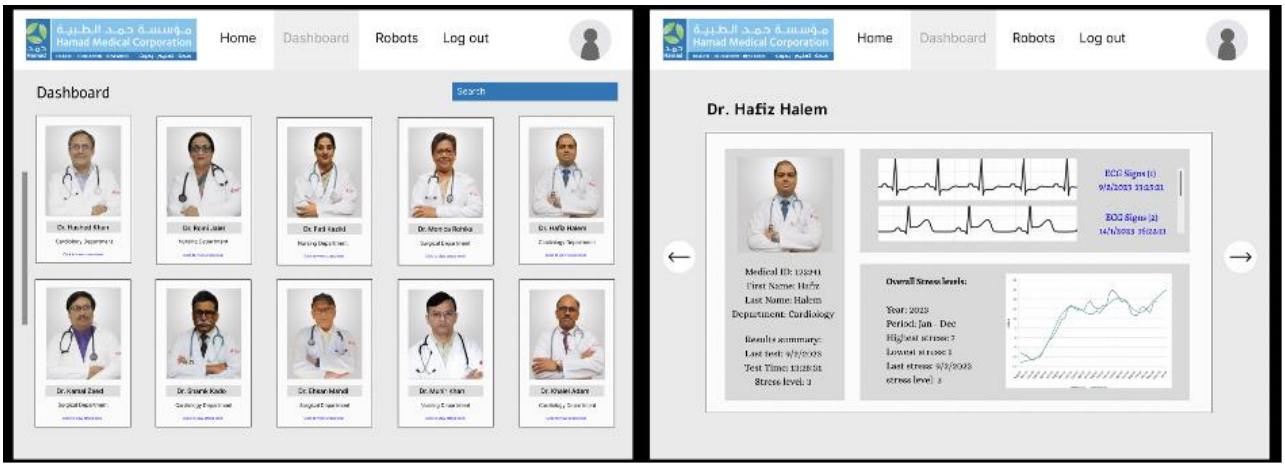


Figure 4. High fidelity workflows

Site Map:

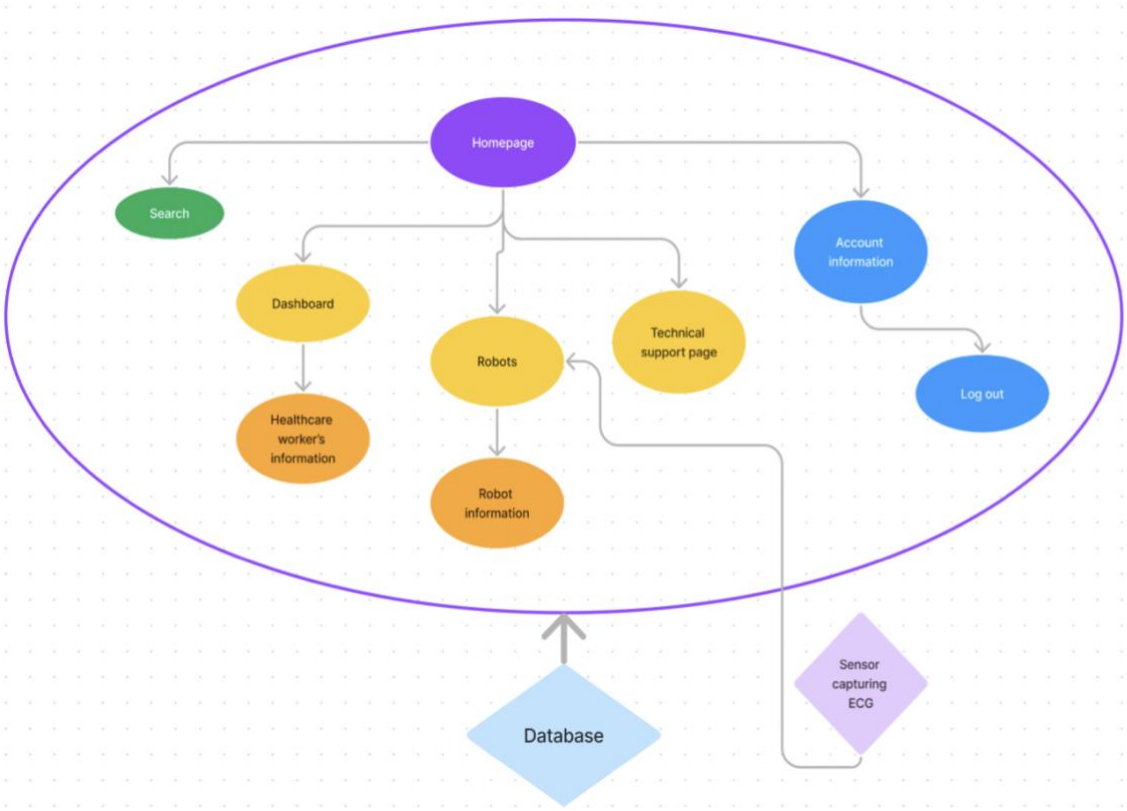
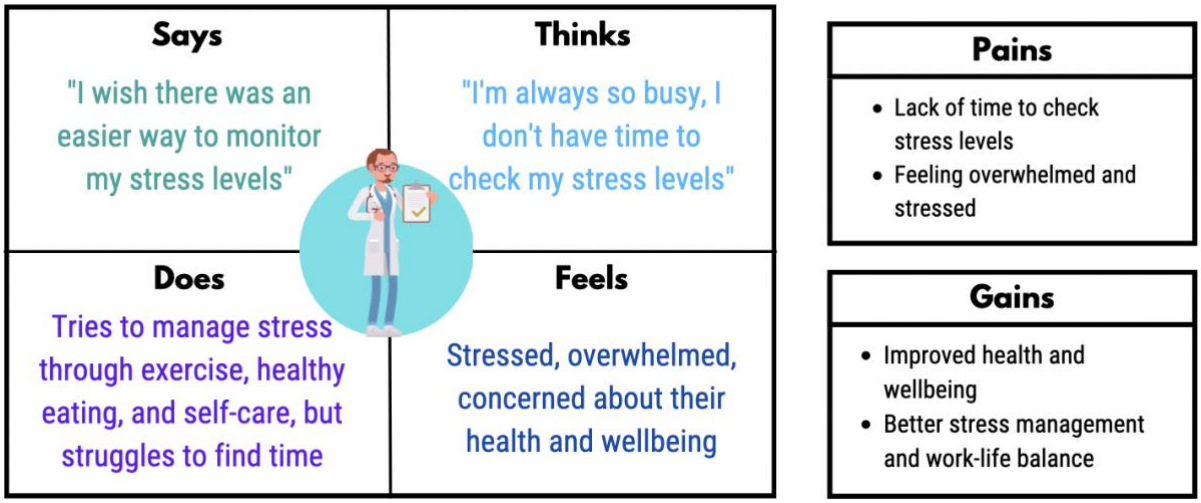


Figure 5. Site Map

Empathy Maps:



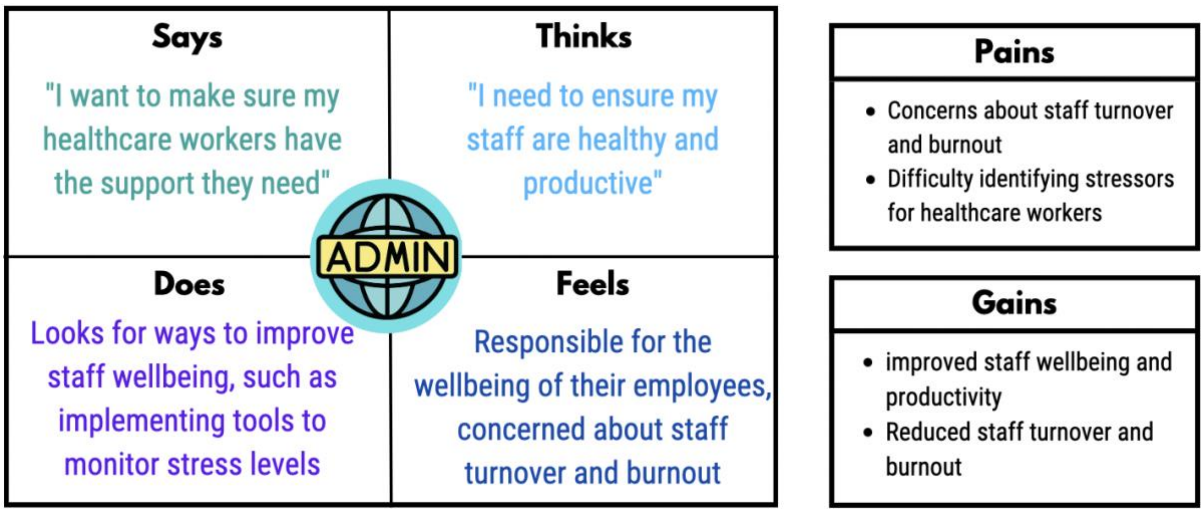


Figure 6. Empathy Maps

Experience Map:

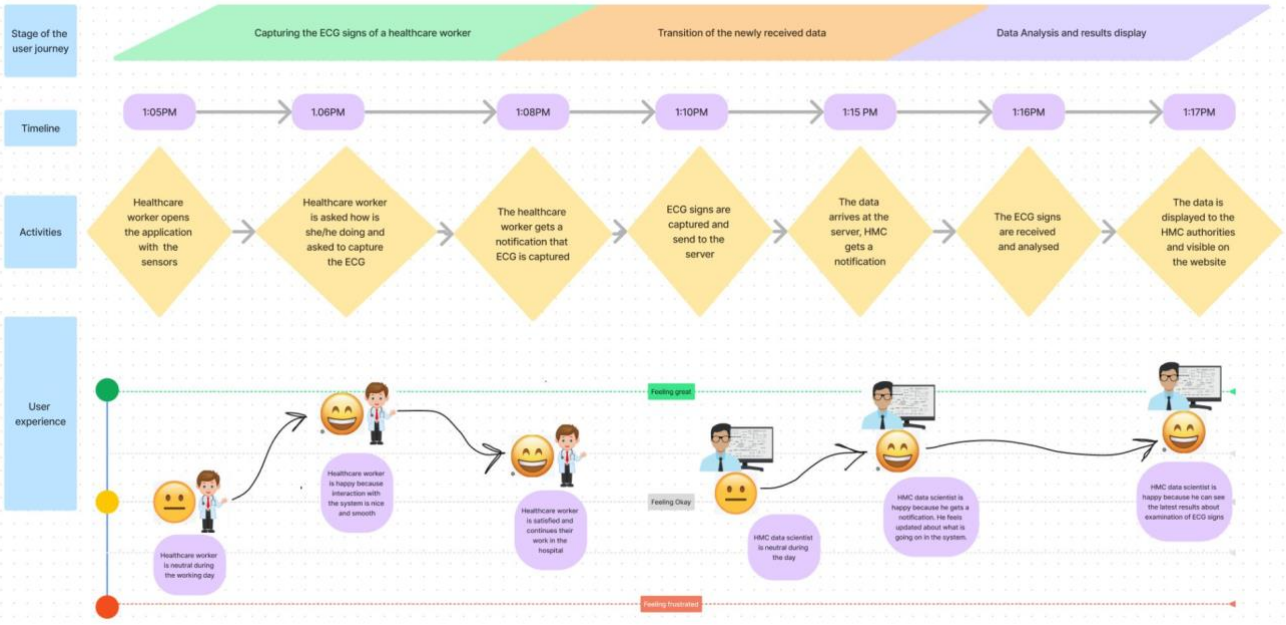


Figure 7. Experience Maps

Conclusion:

Our healthcare worker ECG monitoring application is designed to provide real-time monitoring of stress levels for healthcare workers, improving employee health and wellbeing and ultimately leading to better patient care outcomes. By focusing on user-centered design principles and using a design process that emphasizes research, prototyping, and usability testing, we aim to create a solution that is simple, accessible, and effective for healthcare workers and hospital administrators.