**Software Architecture**

Streaming Wearables and Stress Measurement Platform

|  |  |
| --- | --- |
| **Date** | 3-Oct-22 |
| **Version** |  |
| **Status** |  |
| **Author** | Victoria Fong, Milan Koster, Michael Osuntuyi, Marinda Boshoff |

**Version**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Amendments** | **Status** |
| 0.1 | 26-09-2022 | M Boshoff | Initial setup of document | Draft |
| 0.1 | 19-09-2022 | M Boshoff | Introduction, ERD, System Context Diagram, Container Diagram | Draft |

Table of Contents

[Glossary 4](#_Toc125046676)

[Introduction 5](#_Toc125046677)

[Backend 6](#_Toc125046678)

[Architecture 6](#_Toc125046679)

[Entity Relation Diagram (ERD) 7](#_Toc125046680)

[C4 Models 8](#_Toc125046681)

[System Context Diagram 8](#_Toc125046682)

[Container Diagram 9](#_Toc125046683)

[API Endpoints 10](#_Toc125046684)

[Frontend 12](#_Toc125046685)

[UI Mockups 12](#_Toc125046686)

[Final Designs 19](#_Toc125046687)

# Glossary

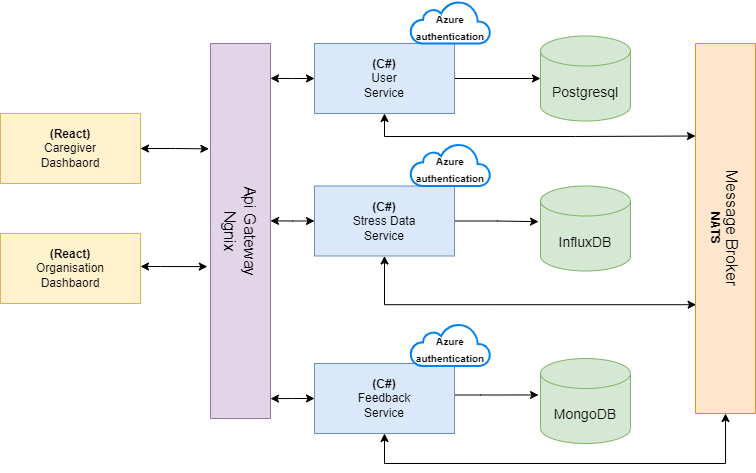
|  |  |
| --- | --- |
| **Phrase/Abbreviations** | **Description** |
| User(s) mentioned in the API endpoints User service section | Stands for both Patients and Caregivers |
| TenantID | Every mention of tenantID signifies organisationID which will be added by the swsp admin in the organisation dashboard |
| Patient Groups and Users mentioned in the user service | Every patient group and user are connected to a tenantID |
|  |  |

# Introduction

The goal of this document is to show how the system is built. The architecture of the system will be explained on different levels and design choices will be shown and discussed. In the first section the backend architecture is shown and the second section show the frontend mockups.

# Backend

## Architecture



## Entity Relation Diagram (ERD)

Graphical user interface, diagram

Description automatically generated

The entity relation database diagram illustrates how the entities in the SWSP relate to each other within the SWSP.

**Organisation – User**An organization can have 0 to many users. A user must be linked to exactly one organization.

**Organisation – Patient Group**An organization can have 0 to many patient groups. A patient group must be linked to exactly one organization.

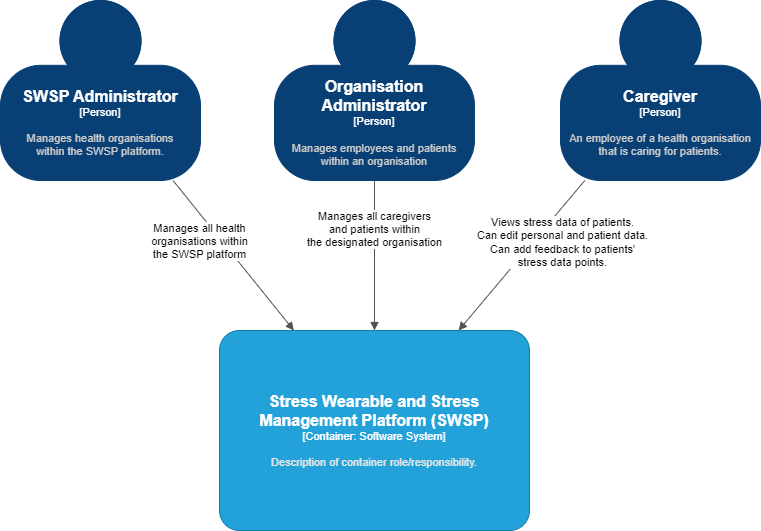
**Patient Group – User**A Patient group can have 0 to many users. A user can be linked to 0 to many patient groups.

**Feedback**A Feedback point is written by one user where a user can have multiple feedbacks. Further, a feedback point belongs to one stress point and vice versa.

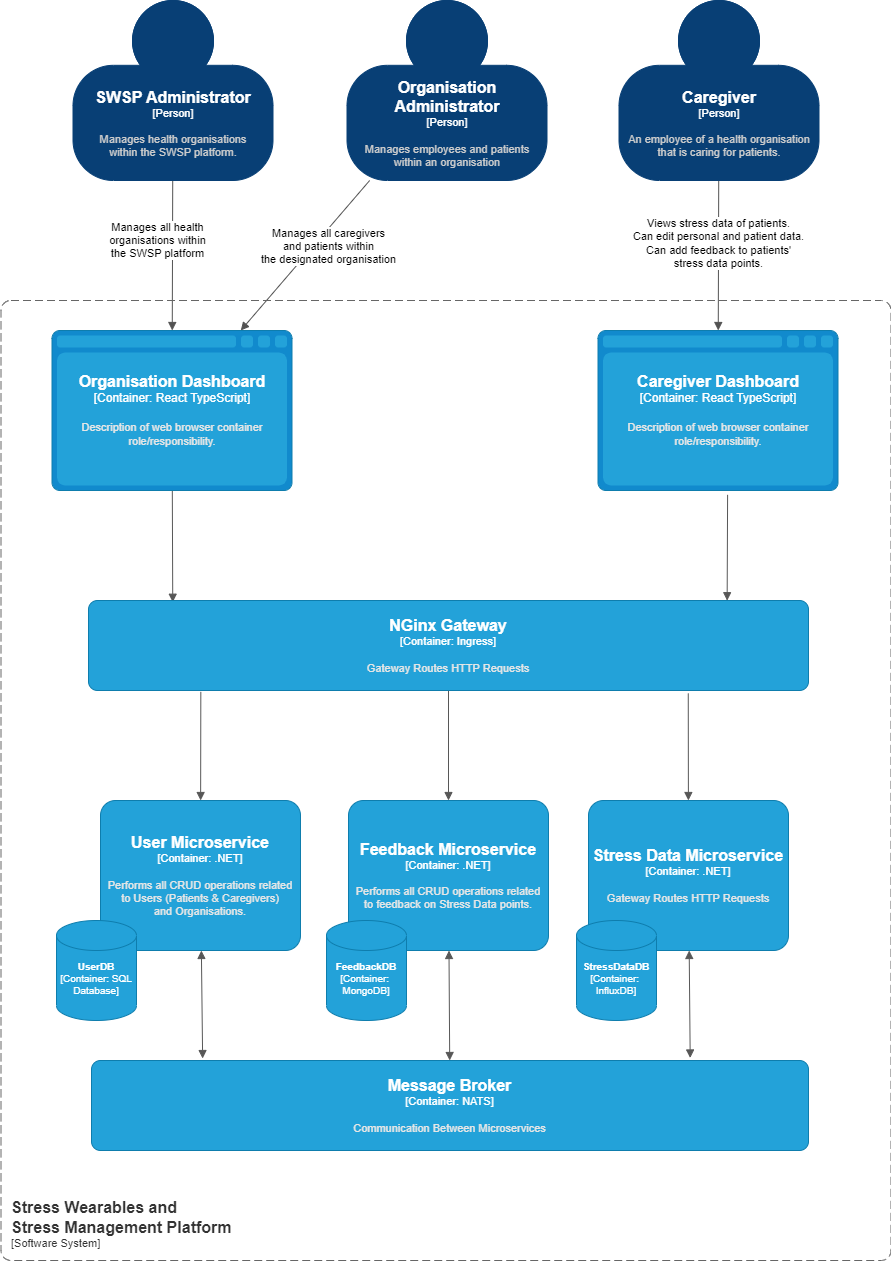
## C4 Models

The C4 model is an "abstraction-first" approach to diagramming software architecture, based upon abstractions that reflect how software architects and developers think about and build software.

### System Context Diagram



### Container Diagram



## API Endpoints

### User Service (Organisation Controller)

|  |  |  |
| --- | --- | --- |
| Resource | HTTP | Description |
| /organizations | Get | Get all organisations that have been added |
| /organizations/:id | Get | Get a specific organisation based on the organisation ID |
| /organizations/patientgroups/:id | Get | Get patient groups connected to the specified organisation ID |
| /organizations | Post | Add an organisation |
| /organizations/:id | Put | Update an organisation information |
| /organizations/:id | Delete | Delete a specific organisation from the database |

### User Service (User Controller)

|  |  |  |
| --- | --- | --- |
| Resource | HTTP | Description |
| /users | Get | Get all users (Patients and Caregivers) from an organisation. |
| /users/:id | Get | Get a specific user based on the User ID. |
| /users/patients | Get | Get all patients from an organisation. |
| /users | Post | Add a user. |
| /users/:id | Put | Update a specific users' information. |

### User Service (Patient Group Controller)

|  |  |  |
| --- | --- | --- |
| Resource | HTTP | Description |
| /patient-groups | Get | Get all patient groups connected to a tenantID (organisationID) passed on after user log in |
| /patient-groups/:patientGroupID | Get | Get patient a specific patient group by ID (this patient group is also connected to a specified tenantID passed on after user logs in) |
| /patient-groups/:patientGroupID/patients | Get | Returns all patients in the specified patient group |
| /patient-groups/:patientGroupID/caregivers | Get | Returns all caregivers in the specified patient group |
| /patient-groups/patients/:userId | Get | Get all patient groups belonging to a specific patient (based on patient UserID) |
| /patient-groups/caregivers/:userId | Get | Get all patient groups belonging to a specific caregiver (based on caregiver UserID) |
| /patient-groups | Post | Add a patient group |
| /patient-groups/:patientGroupID/user | Post | Assign a user (Patient/Caregiver) to a specific patient group |
| /patient-groups/:id | Put | Update a specified patient group information |
| /patient-groups/:id | Delete | Delete a specified patient group |
| /patient-groups/:groupId/user | Delete | Remove user (Patient/Caregiver) from patient group |

### Feedback Service

|  |  |  |
| --- | --- | --- |
| Resource | HTTP | Description |
| /feedback | Get | Get all feedback. |
| /feedback/{id} | Get | Get feedback by its id. |
| /feedback/patient/{patientId} | Get | Get all feedback from a patient based on their patient id. |
| /feedback/timespan/{patientId} | Get | Retrieves all feedback from a patient based on their patient id and indicated timespan. |
| /feedback/{id} | Post | Creates new feedback. |
| /feedback/{id} | Put | Updates feedback by id. |
| /feedback/{id} | Delete | Remove feedback by id. |

### Stress Data Service (HrvMeasurements Controller)

*HRV stands for Heart Rate Variability*

|  |  |  |
| --- | --- | --- |
| Resource | HTTP | Description |
| /hrvMeasurements | Get | Get all HRV measurements. |
| /hrvMeasurements/{id} | Get | Get a HRV measurement by its Id. |
| /hrvMeasurements/patient/{patientId} | Get | Get all HRV measurements belonging to a patient. |
| /hrvMeasurements/date/{patientId}  ?date={date} | Get | Gets all HRV measurements belonging to a patient on a date. |
| /hrvMeasurements/date/{patientId}/timespan?startTime={date}&endTime={date} | Get | Gets all HRV measurements belonging to a patient between a start date and end date. |
| /hrvMeasurements/wearable/{wearableId}  /timespan?startTime={date}&endTime={date} | Get | Gets all HRV measurements belonging to a wearable between a startDate and endDate. |
| /hrvMeasurements | Post | Creates new HRV measurement. |
| /hrvMeasurements/{id} | Put | Updates HRV measurement by id. |
| /hrvMeasurements/{id} | Delete | **Not Implemented.** Deletes HRV measurement by id. |
| /hrvMeasurements/nats/simulate | Get | Simulates a Nats publish stress:created for developer purposes. |

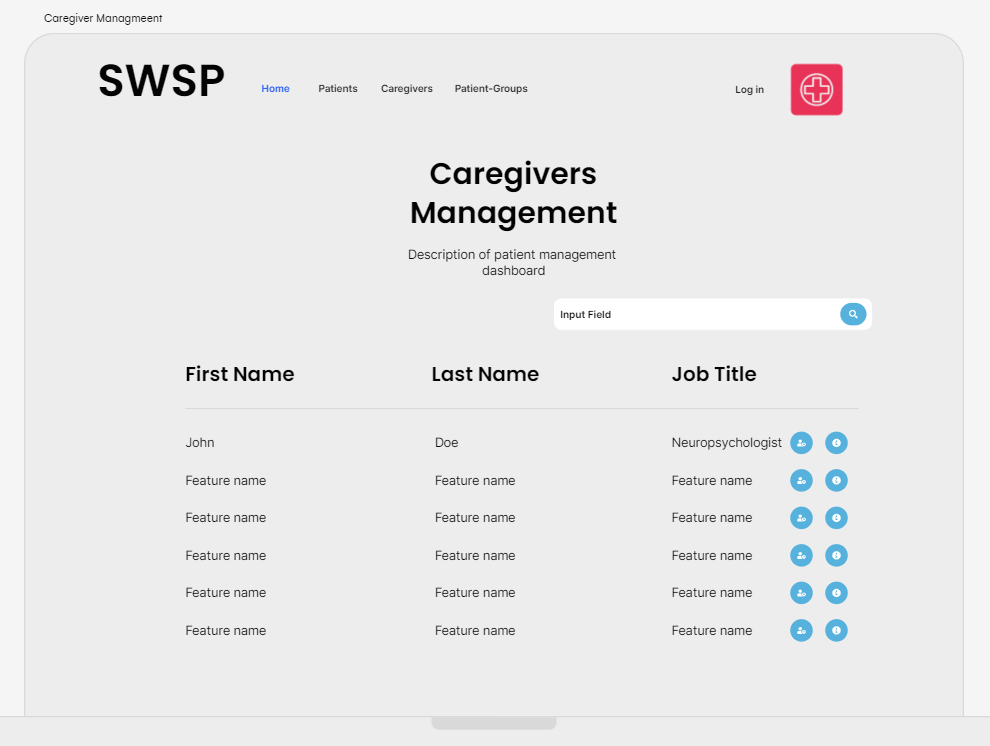
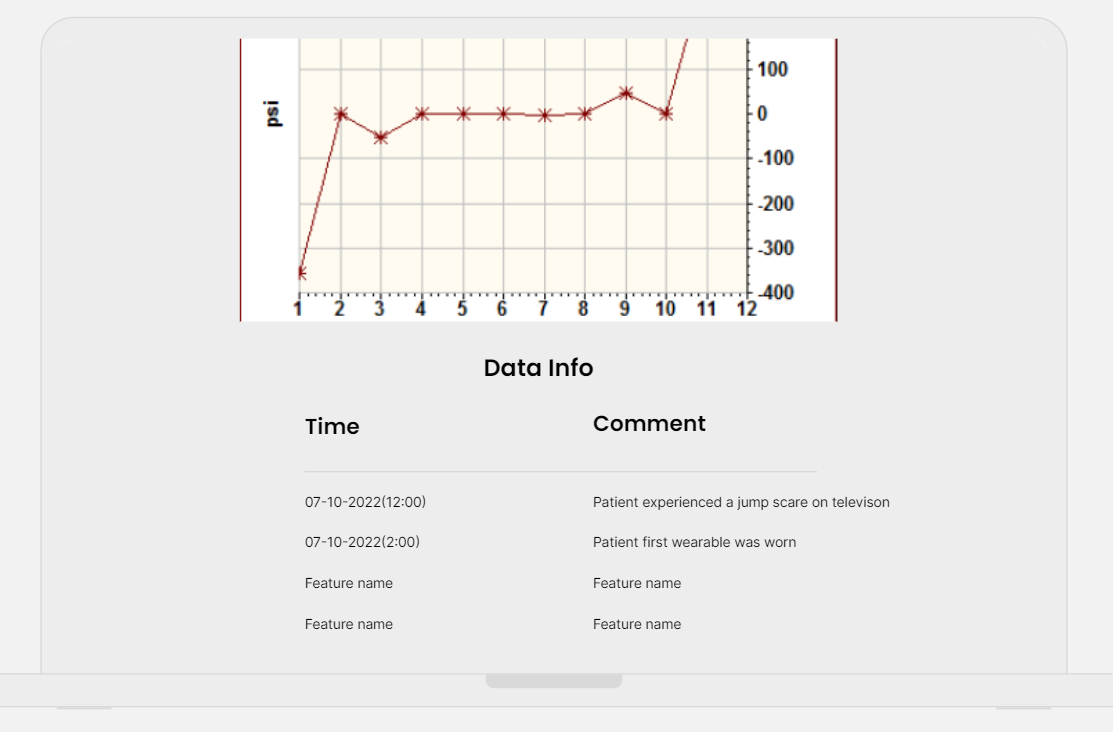
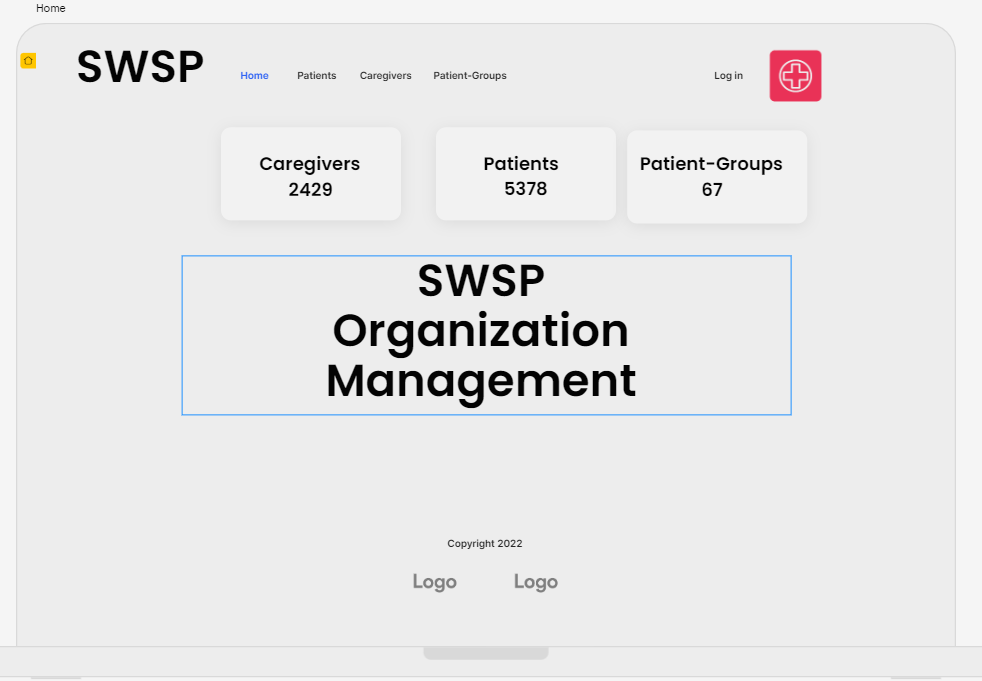
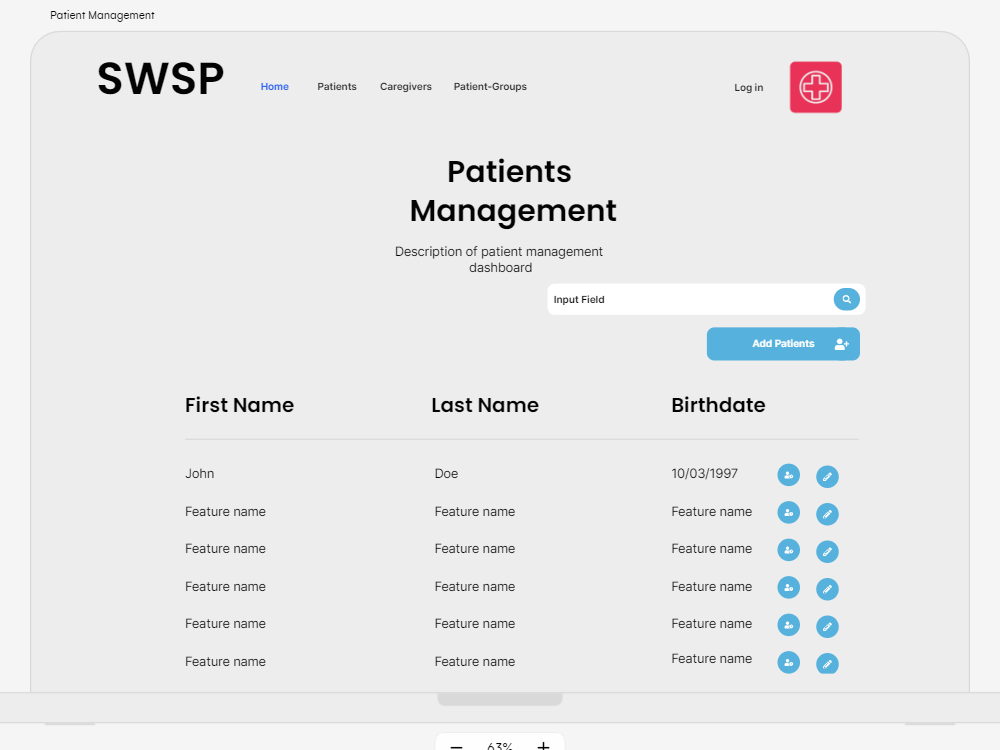
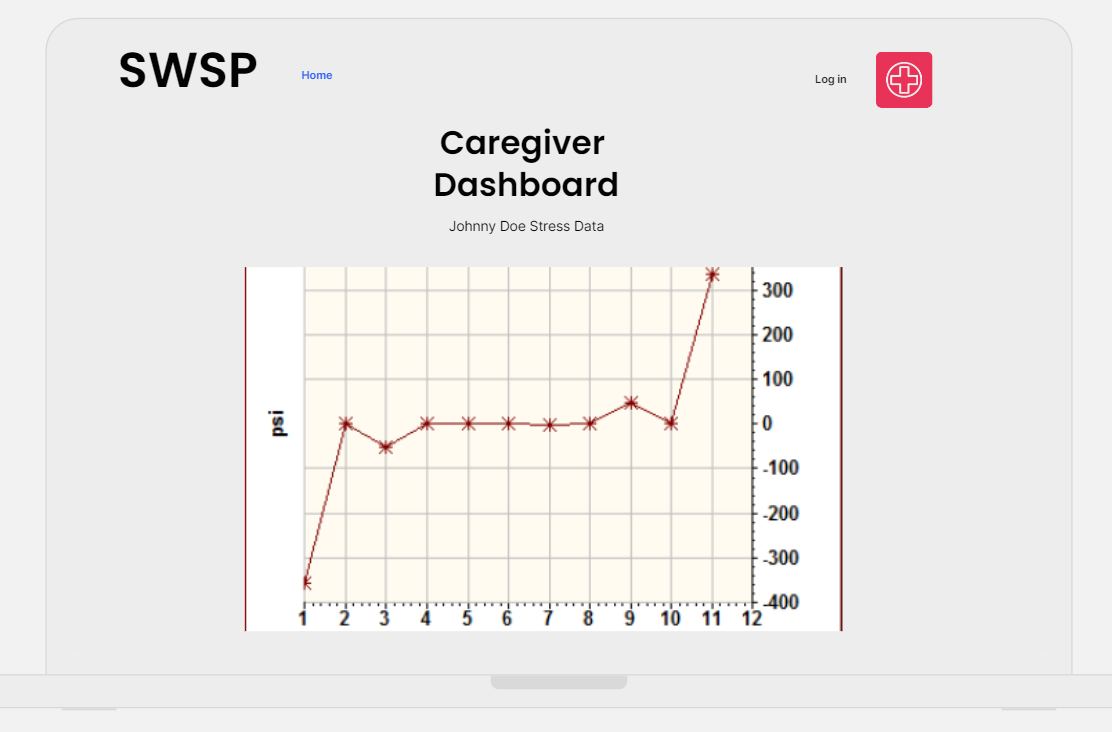
### Stress Data Service (Patients Controller)

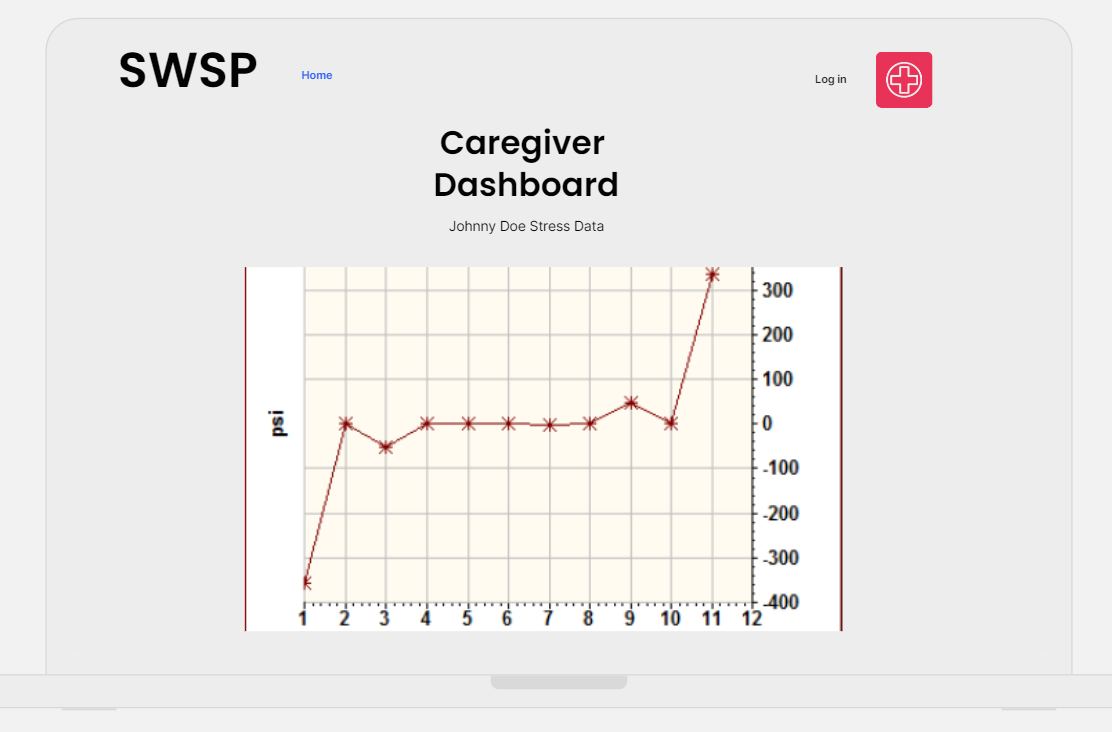
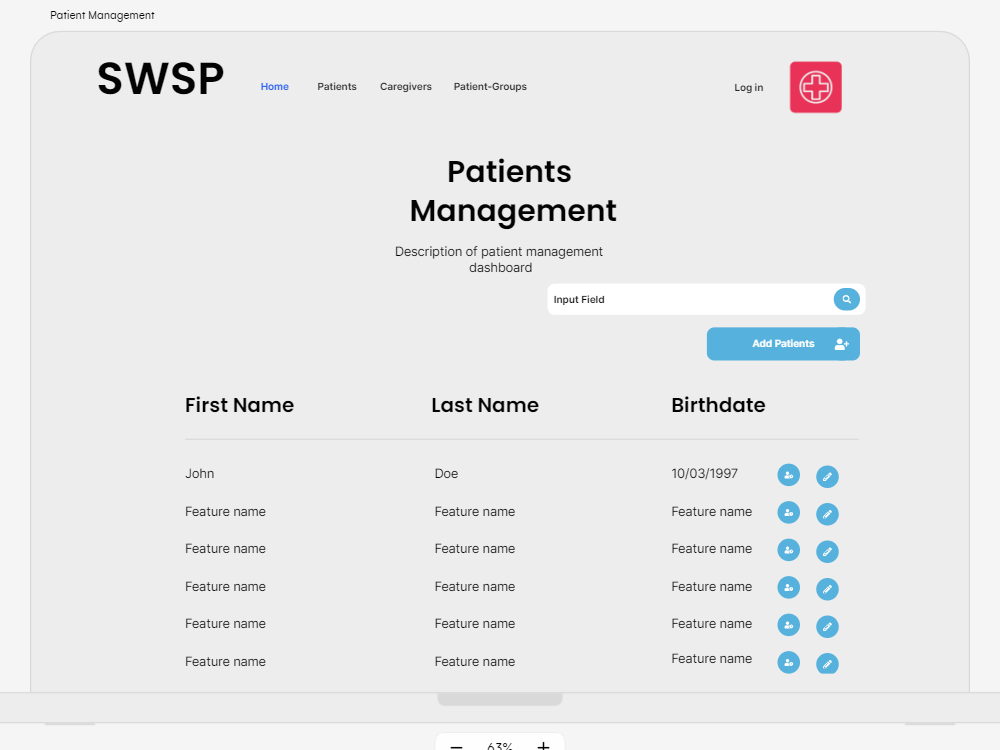
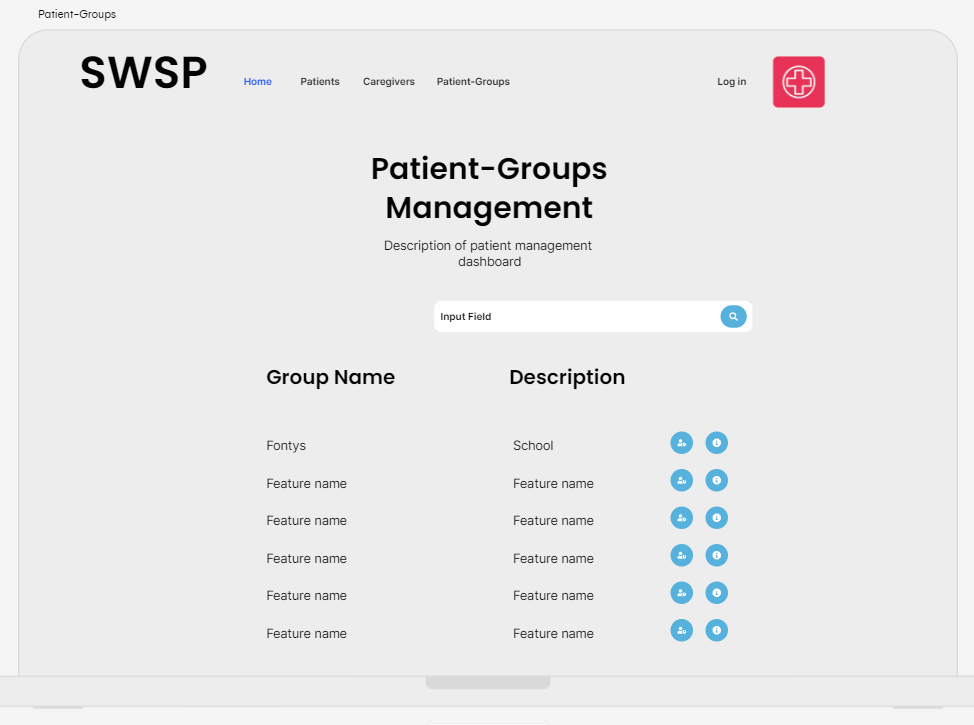
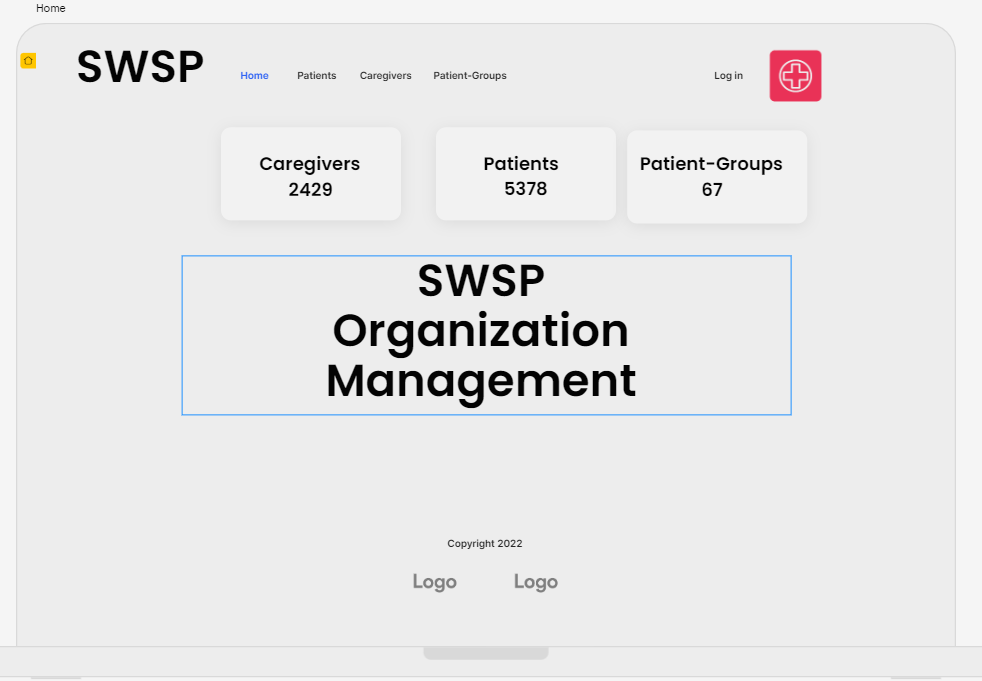
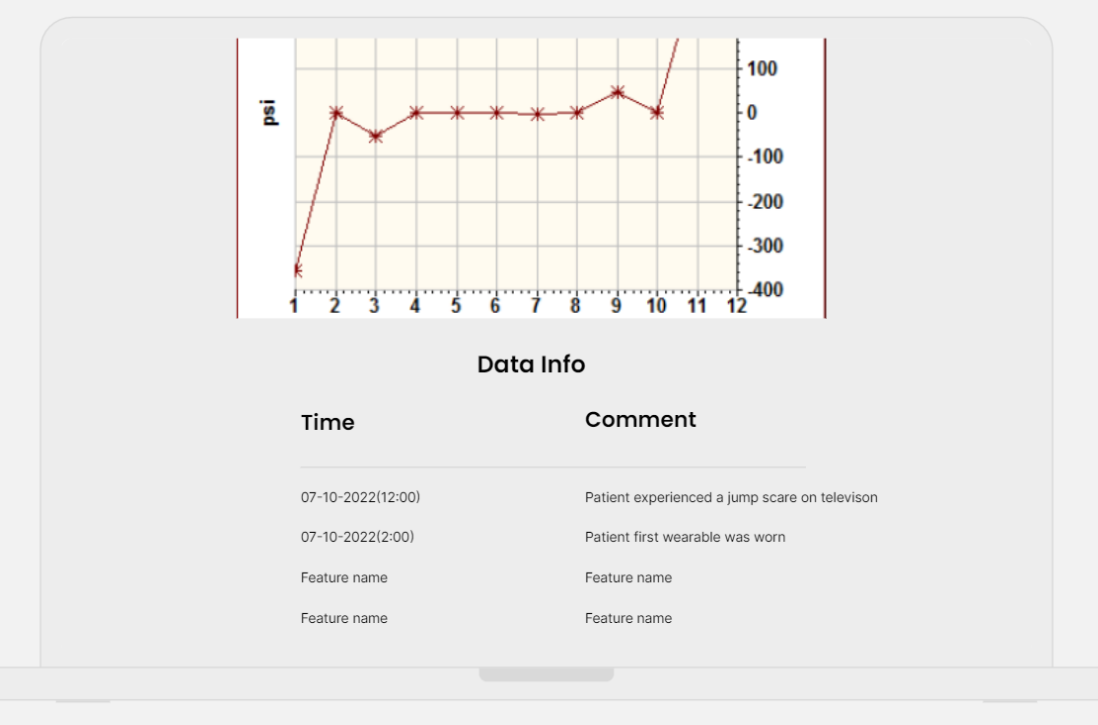
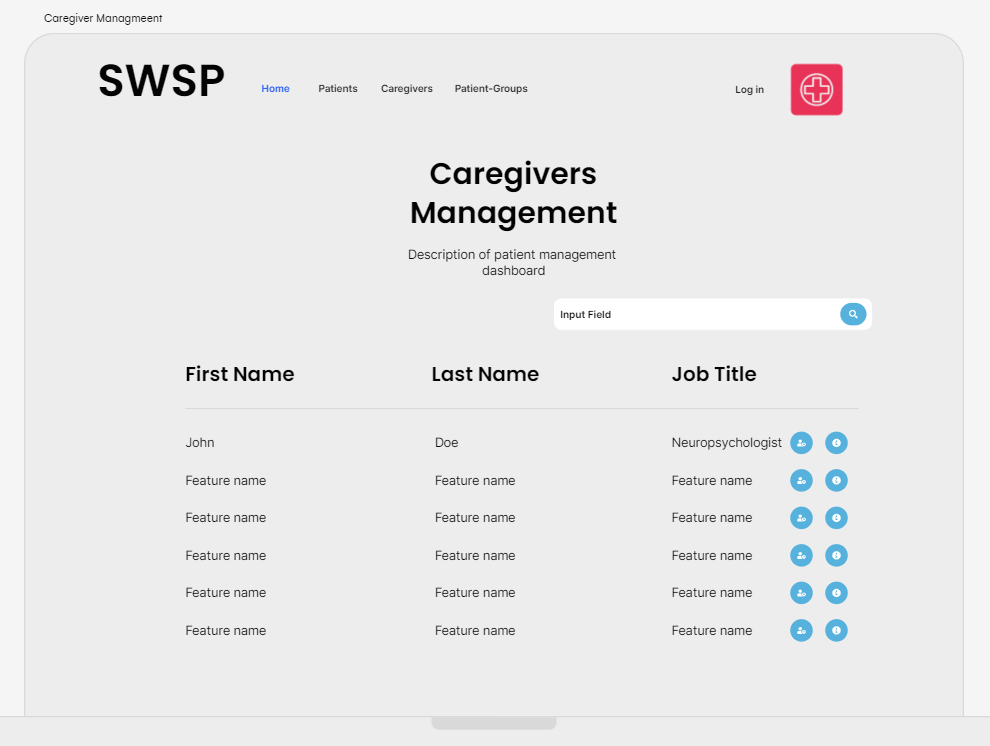
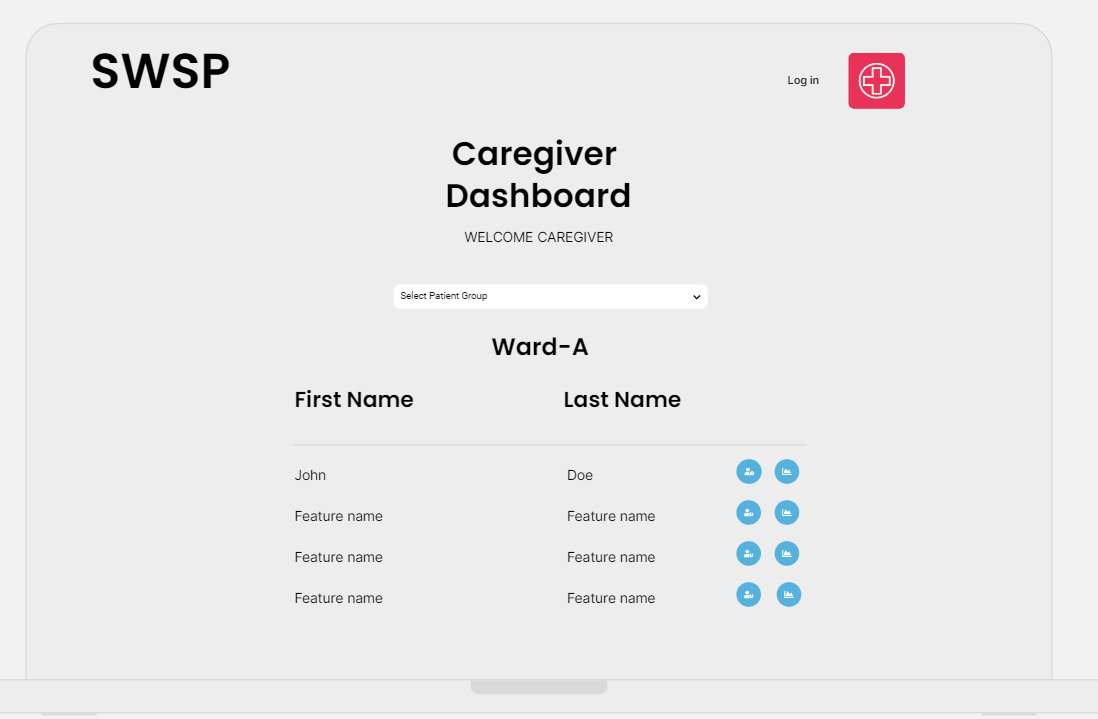
*HRV stands for Heart Rate Variability*

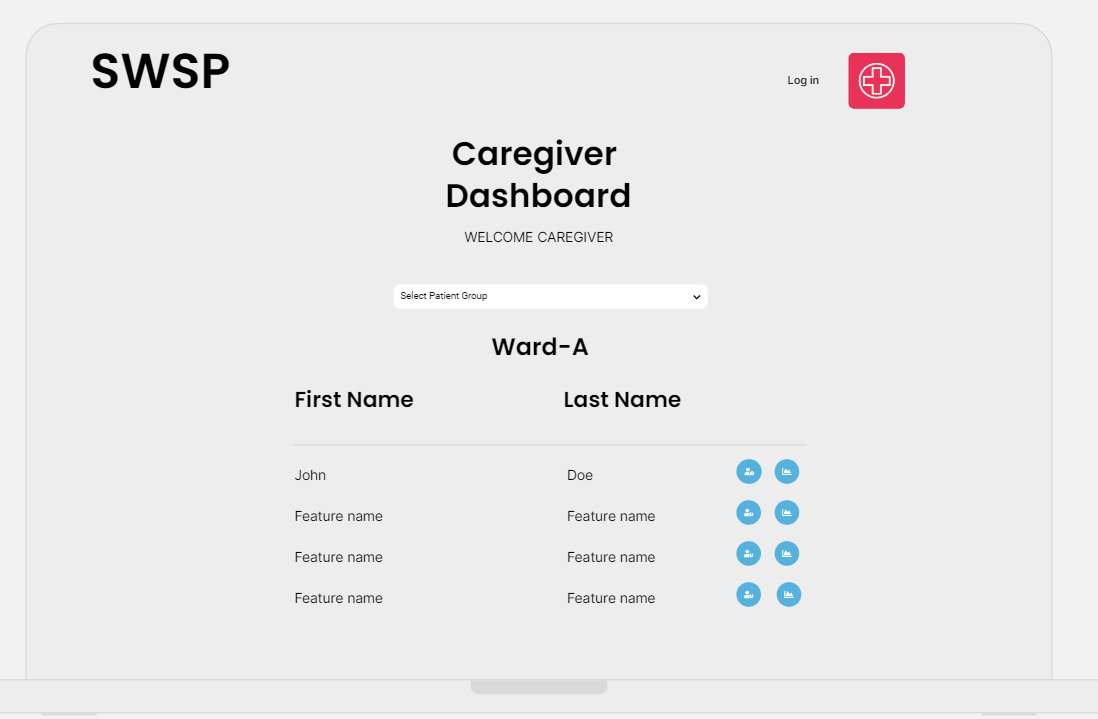
|  |  |  |
| --- | --- | --- |
| Resource | HTTP | Description |
| /stressed/belowValue | Get | **Not implemented**. Gets all recently stressed patients. |

# Frontend

## UI Mockups







## Final Designs

