**Future Implementations**

Stress Wearables



|  |  |
| --- | --- |
| **Date** | 16-01-2023 |
| **Version** | 0.1 |
| **Status** | Concept |
| **Author** | Victoria Fong, Milan Koster, Michael Osuntuyi, Marinda Boshoff |

Version

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Amendments** | **Status** |
| 0.1 | 11-01-2023 | Michael Osuntuyi | Added section “Introduction”, “Future Implementations”, “Organisation Dashboard Visual bug” & subsection “Unused Code” under the “Unused Files” section. | Concept |
| 0.1 | 11-01-2023 | Milan Koster | Added subsection “SQL Injection” under the “Unused Files” section. | Concept |
| 0.1 | 16-01-2023 | Victoria Fong | * Initial setup/layout of document, table of contents, glossary, list of abbrevations * Added subsection “Real-Time data visualization” under the “Future Implementations” section. * Added subsection "FHP Students Suggestions” under the “Future Implementations” section. | Concept |
| 0.1 | 16-01-2023 | Milan Koster | Added subsection “User Stories” under the “Feature Section” section. | Concept |

Table of Contents

[Glossary 4](#_Toc125318867)

[List of Abbreviations 4](#_Toc125318868)

[Introduction 5](#_Toc125318869)

[1. Future Implementations 6](#_Toc125318870)

[1.1 User Stories 6](#_Toc125318871)

[1.2 FHP Student Suggestions 6](#_Toc125318872)

[1.3 Software Features 6](#_Toc125318873)

[1.4 Caregiver dashboard 7](#_Toc125318874)

[2. Software Improvements 8](#_Toc125318875)

[2.1 Cleaning Up Code 8](#_Toc125318876)

[2.1.1 Unused code 8](#_Toc125318877)

[2.1.2 User service 8](#_Toc125318878)

[2.1.3 Stress algorithm service 8](#_Toc125318879)

[2.2 SQL Injection 9](#_Toc125318880)

[2.3 Docker Volumes 9](#_Toc125318881)

[2.4 Add Raw Data Services 9](#_Toc125318882)

[2.5 User Service Roles 9](#_Toc125318883)

[2.6 Status Codes over Exceptions 9](#_Toc125318884)

Glossary

|  |  |
| --- | --- |
| Word/Phrase | Definition |
| Heart Rate Variability | Heart rate variability is where the amount of time between your heartbeats fluctuates slightly. |

List of Abbreviations

|  |  |  |
| --- | --- | --- |
| Word/Phrase | Meaning | Page |
| FHP Students | The FHP students is an abbreviation for Fontys Allied Health Professionals Students. | 7 |

Introduction

The purpose of this document is to inform the reader about the possible future implementations that can be done for the Streaming Wearables and Stress Measurement Platform (SWSP) project. The section ‘Future Implementations’ refer to features that can be added to the software and the section ‘Software Improvements’ refer to software related implementations that can be done to improve the platform software or developer experience.

# Future Implementations

## User Stories

The user stories of the project can be found under [Requirements](https://stichtingfontys.sharepoint.com/:w:/r/sites/StressmeasurementPlatform-General-SoftwareAdvance/Gedeelde%20documenten/General%20-%20Software%20Advance/Documentation/Analysis/Requirements.docx?d=w5c70ac48d86a4aca927198721d75ff8d&csf=1&web=1&e=KhmJ78) under analysis. The document shows which documents were considered and what portion of these user stories have been implemented in the current iteration of the project.

## FHP Student Suggestions

This section outlines the suggestions and advice given by the FHP students after reviewing the software platform.

There is a document based on the user feedback received from the FHP students to the usability of the platform and what features can be added to the dashboard as a caregiver, user & physiotherapist perspective.

This document can be found in the reading guide under the Future Implementations section named “FHP Students Feedback Caregiver Dashboard (‘FHP Test Cases’ folder)”

## Software Features

This subchapter what software features have a high priority of being implemented next.

1. Real Time Data Visualisation. Currently we are able to mock the data for the heart-rate variability in the graph of the web application, but we have not yet made a connection to the wearables for live data.
2. Connecting to Wearable APIs. We have done research into implementing this functionality where needed resources can be looked at for any ideas to how to get started. The research document is referenced in the reading guide under the Research section named “Wearable API Research”.
3. Patient Dashboard. A mobile and desktop application for patients and/or their immediately family to view and interact with their data.
4. Two Factor Authentication. Allow users to log in using 2Fa.

## Caregiver dashboard

This subchapter describes suggested improvements to the caregiver dashboard.

1. Add notifications of stressed patients. This should make it easier for caregivers to know which patients to look at.
2. Add labels to graph axis and normalise values. This should make it clearer what data is being visualised. Add a ‘green – orange – red’ indicator for different levels.
3. Add feedback to the graph datapoints. The feedback should be shown on hover.
4. Show the author of the feedback.
5. In feedback, give users the option to indicate whether stress was positive or negative. Indicate whether comments are positive or negative using colours.
6. Add navigation buttons within the platform, such as an obvious indicator to return to the homepage.
7. When returning to the homepage, retain the previously viewed patient group.
8. Include patient info on their profile.
9. Include a profile picture for patients and/or caregivers.

# Software Improvements

## Cleaning Up Code

### Unused code

* We did not manage to use the Kubernetes folder files in all services.

### User service

* Commented out code in the Patient Group and User controllers can be deleted.
* NATS code is commented out but should be fixed to work.

### Stress algorithm service

* This service is entirely unused and wasn’t touched by our team.

## SQL Injection

One of the existing security flaws that is present in the application is SQL injection. This is more prevalent in the Stress Data Service, which makes use of a time series database named [InfluxDB](https://www.influxdata.com/) and [Flux](https://docs.influxdata.com/influxdb/cloud/query-data/get-started/) queries. According to the documentation, one method of preventing this would be to use [parameterised Flux queries](https://docs.influxdata.com/influxdb/cloud/query-data/parameterized-queries/).

## Docker Volumes

Docker volumes can be used to retain data when services and shut down and restarted. This would make the developer flow significantly easier, as it is currently necessary to re-register the organisation and create patients as well as patient groups each time this happens.

## Add Raw Data Services

Add a Raw data service to process the data from wearables. The Stress Data Service stores the processed data, which should be received from another (non-existent) microservice.

## User Service Roles

The user service contains two roles, which are stored as a string. These roles are ‘Patient’ and ‘Caregiver’. Instead, an Enum should be used, or the roles should be made flexible.

## Status Codes over Exceptions

Occasionally exceptions are thrown in the User service. Instead, status codes should be used to return an appropriate response.