**Technical Report**

Project Name DePause Appointment System

Student Name and Number Volodymyr Rosinskyi [x23252065](mailto:%78%32%33%32%35%32%30%365@%73t%75d%65%6et%2e%6e%63ir%6c%2e%69%65)

**Overview**

*This document is submitted in partial fulfillment of the project Module running on the Higher Diploma in Computing module delivered by Dr Yasantha Samarawickrama.*

Table of Contents

Project Overview 3

Requirements Specification Migration 3

technical specification 3

Self reflective analysis of the initial Gantt chart submitted 3

Challenges Encountered 3

Conclusion and Future Work 3

# Project Overview

This platform helps psychologists manage their schedules and connect with clients more efficiently. Users can browse psychologists, book appointments, and leave reviews.

Many psychologists rely on manual systems for booking and managing clients. This platform fills the gap by automating these processes.

Unlike costly alternatives like BetterHelp, this project focuses on simplicity and affordability for both users and psychologists.

Online mental health services are growing rapidly, with the global market expected to reach $10 billion by 2027. This platform is designed to cater to this demand.

# Requirements Specification Migration

Initially focused on booking appointments, the project expanded to include features like user reviews, role-based access, and detailed profile management for psychologists and admin.

Manual scheduling needed automation to save time for psychologists.

Testing revealed the necessity of role-based access for better security.

Stakeholder feedback emphasized the need for user-friendly profile customization.

Role-specific features, profile photo uploads, and integrated calendars make the platform much more practical and secure compared to its initial scope.

# Technical specification

# *Languages/Frameworks:*

# *Backend: Java + Spring Boot*

# *Frontend: React.js + Material UI*

# *Web: RESTful APIs handle communication between the frontend and backend using Axios.*

# *Database: PostgreSQL stores user, psychologist, and appointment data with optimized relationships for performance.*

# *Security:*

# *JWT secures user authentication.*

# *Role-based access controls protect sensitive data.*

# *HTTPS ensures secure communication.*

# *Data: Combines user-uploaded photos with structured database information.*

# *Deployment: Currently runs locally, with plans for AWS deployment using Docker.*

# *Additional Tools:*

# *UI Libraries: Material UI was chosen after testing other libraries for better visual design.*

# *Photo Storage: Amazon S3 was used to handle user profile photos securely.*

# *API Documentation: Swagger was implemented to document and test backend endpoints efficiently.*

# Self reflective analysis of the initial Gantt chart submitted

Integration: APIs didn't always match frontend expectations, requiring improved logging and documentation.

Database Design: Structuring data for psychologists, users, and appointments needed multiple revisions.

Errors: Debugging authentication took extra effort, resolved with enhanced error handling.

Performance: Pagination improved the speed of listing users and psychologists.

Design Challenges: A lot of time was spent on designing the interface and testing various UI libraries before finalizing Material UI.

Endpoints: Not all endpoints were fully integrated into the frontend due to time constraints.

Complex Features: Some backend features turned out to be much harder to implement on the frontend than expected.

# Conclusion and Future Work

# Spring Boot made backend development robust, while React + Material UI ensured a clean, responsive frontend.

# Proper role management and early testing are crucial for secure and functional systems. Swagger made API testing much smoother.

# Plan to rework some backend logic and fully integrate the remaining endpoints into the frontend. Additionally, I aim to implement messaging integrations with WhatsApp and Telegram to improve communication between psychologists and clients

# .References

Roles for entity

<https://www.youtube.com/watch?v=L8M_eXV0OVk>

<https://www.youtube.com/watch?v=mq5oUXcAXL4>

<https://stackoverflow.com/questions/49192618/using-roles-in-spring-boot-jpa-application>

//jwt tokens and expiration

<https://github.com/ali-bouali/spring-boot-3-jwt-security/blob/main/src/main/resources/application.yml>

<https://stackoverflow.com/questions/78805779/issue-with-parserbuilder-method-in-jjwt-library-for-jwt-token-validation>

<https://medium.com/@mtl98/avoid-using-integers-for-temporal-data-eg-age-and-use-dates-instead-d13a9743c2a1>

<https://stackoverflow.com/questions/6667243/using-enum-values-as-string-literals>

<https://ru.stackoverflow.com/questions/1128951/java-generics-%D0%92-%D1%87%D0%B5%D0%BC-%D1%80%D0%B0%D0%B7%D0%BD%D0%B8%D1%86%D0%B0-%D0%BC%D0%B5%D0%B6%D0%B4%D1%83-wildcard-%D0%B8-parameterized-typest>

<https://reflectoring.io/bean-validation-with-spring-boot/>

<https://medium.com/@AlexanderObregon/how-to-set-up-email-notifications-in-spring-boot-applications-b5a2574c5e8f>

<https://mui.com/material-ui/react-app-bar/#app-bar-with-responsive-menu>

<https://www.pexels.com/ru-ru/photo/4926674/>

<https://www.pexels.com/ru-ru/photo/4927361/>

<https://www.pexels.com/ru-ru/photo/4699273/>

<https://www.pexels.com/ru-ru/photo/4554279/>

<https://www.pexels.com/ru-ru/photo/2741701/>

<https://www.pexels.com/ru-ru/photo/3907595/>

<https://www.pexels.com/ru-ru/photo/4355256/>

<https://www.pexels.com/ru-ru/photo/4197932/>

https://www.pexels.com/ru-ru/photo/4052692/