



MyTherapyPal: A therapy progress and goal tracking application

Interim Report

**TU856
BSc in Computer Science**

Shane Buckley

C20703429

Supervisor: Eoin Rogers

School of Computer Science
Technological University, Dublin

20/11/2023

Abstract

The aim of this project is to create a comprehensive cross-platform mobile and rich web application aimed at providing psychotherapists, counsellors and their clients with a platform and tools to facilitate tracking therapy progress as its primary focus. The application will be designed to have a user account system with login and authentication, a note taking system that will have a summarization and sentiment analysis feature using artificial intelligence for client users, a note expansion feature for therapist users who may take notes during therapy in shorthand, therapist listings and map, mood tracking system, CBT (cognitive behavioural therapy) techniques, task assignment from therapists to clients, secure text chat, secure video calling and a payments system for clients to pay their therapists.

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

*Shane Buckley*_____

Shane Buckley

20/11/2023

Acknowledgements

I would like to thank my supervisor Eoin Rogers for providing guidance and feedback throughout the interim stage of my final year project.

I would also like to thank my Mother and especially my partner Grace, for their support and help with this project, they have provided me with an abundance of knowledge in the mental health field and this project would not be possible without them.

And finally, I would like to thank my family for their continued support and encouragement.

Table of Contents

TABLE OF FIGURES	8
1. INTRODUCTION	9
1.1. PROJECT BACKGROUND	9
1.2. PROJECT DESCRIPTION	9
1.3. PROJECT AIMS AND OBJECTIVES	10
1.4. PROJECT SCOPE.....	10
1.5. THESIS ROADMAP	11
2. LITERATURE REVIEW	12
2.1. INTRODUCTION	12
2.2. ALTERNATIVE EXISTING MENTAL HEALTH ASSISTANCE APPLICATIONS	12
2.2.1. <i>Quenza</i>	12
2.2.2. <i>Daylio</i>	13
2.2.3. <i>Moodnotes</i>	14
2.2.4. <i>eMoods</i>	15
2.2.5. <i>Theratrak</i>	16
2.3. TECHNOLOGIES RESEARCHED & SELECTION	17
2.3.1 <i>Operating Systems</i>	17
2.3.1.1 Android	17
2.3.1.2 iOS	18
2.3.1.3 Windows	18
2.3.1.4 MacOS	18
2.3.2 <i>Databases</i>	19
2.3.2.1 Firebase Cloud Firestore	19
2.3.2.2 MariaDB.....	20
2.3.2.3 PostGreSQL.....	20
2.3.2.4 Selection (Firebase Cloud Firestore)	20
2.3.3 <i>Frontend Framework</i>	21
2.3.3.1 React Native	21
2.3.3.2 Xamarin	21
2.3.3.3 Selection (Flutter)	21
2.3.4 <i>Backend Framework</i>	22
2.3.4.1 Django with Python	22
2.3.4.2Ruby on Rails	22
2.3.4.3 Firebase	22
2.3.4.4 Selection (Firebase)	22
2.4. OTHER RESEARCH YOU'VE DONE	23
2.4.1 <i>Mental Health & Psychotherapy</i>	23
2.4.1.1 Mobile technology for mental health assessment	23
2.4.1.2 Do mental health mobile apps work: evidence and recommendations for designing high-efficacy mental health mobile apps by Pooja Chandrashekar	23
2.4.1.3 Cognitive-Behavioural Therapy (CBT)	24
2.4.1.4 Computer-Assisted CBT and Mobile Apps for Depression and Anxiety	24
2.4.2 <i>Machine Learning</i>	25
2.4.2.1 Natural Language Processing Models.....	25
2.4.2.2 Sentiment Analysis	25
2.4.2.3 CBT Chatbot with Natural Language Processing Models	26
2.4.2.4 Mental Health Datasets for AI research	26
2.5. EXISTING FINAL YEAR PROJECTS	26
2.6. CONCLUSIONS	27
3. SYSTEM DESIGN	28

3.1. INTRODUCTION.....	28
3.2. SOFTWARE METHODOLOGY	28
3.2.1 Agile	28
3.2.2 Waterfall	29
3.2.3 DevOps	29
3.2.6 Selection (Agile & Devops)	30
3.3. OVERVIEW OF SYSTEM	31
3.3.1 Description of System's Operation	31
3.3.2 System Requirements	33
3.4. DESIGN DOCUMENTS.....	34
3.4.1 Class Diagram	34
3.4.2 Firestore Schema Diagram	35
3.4.3 Use Case Diagram	36
3.4.4 Sequence Diagram	37
3.5. CONCLUSIONS	37
4. TESTING AND EVALUATION	38
4.1. INTRODUCTION.....	38
4.2. PLAN FOR TESTING	38
4.3. PLAN FOR EVALUATION	39
4.3.1 Nielsens Heuristic Evaluation	39
4.3.3 User Surveys and Feedback	39
4.3.5 Compliance Testing	40
4.3.6 Scalability Testing	40
4.3.7 Reliability Testing	41
4.3.8 Accessibility Testing	41
4.4. CONCLUSIONS	41
5. PROTOTYPE DEVELOPMENT	42
5.1. INTRODUCTION.....	42
5.2. PROTOTYPE DEVELOPMENT	42
5.2.1 Frontend Application.....	42
5.2.1.1 Login	42
5.2.1.2 Authentication Service	44
5.2.1.3 Registration	45
5.2.1.4 Account Homepage	46
5.2.2 Backend Services	47
5.2.3.1 Firebase Authentication	47
5.2.3.2 Cloud Firestore	47
5.2.3.3 AI Emotional Sentiment API	47
5.3. PROTOTYPE SCREENSHOTS.....	49
5.3.1 Login Screen	49
5.3.2 Registration Screen	50
5.3.3 Account Dashboard Screen	51
5.4. CONCLUSIONS	52
6. ISSUES AND FUTURE WORK	52
6.1. INTRODUCTION.....	52
6.2. ISSUES AND RISKS	52
6.2.1 Confidentiality	52
6.2.2 Natural Language Processing	52
6.2.3 Time Constraints	53
6.3. PLANS AND FUTURE WORK.....	53

6.3.1 System Development.....	53
6.3.2 CBT Chatbot Research.....	53
6.3.1. GANTT Chart	54
BIBLIOGRAPHY	55

Table of Figures

Figure 1 - Quenza Mobile App Screenshots (3)	13
Figure 2 - Daylio Mobile App Screenshots (5).....	14
Figure 3 - Moodnotes Mobile App Screenshots (7)	15
Figure 4 - eMoods Mobile App Screenshots (9).....	16
Figure 5 - Theratrak App Screenshots (11)	17
Figure 6 - Desktop Operating System Market Share Worldwide (16)	19
Figure 7 - Cloud Firestore Data Organization (17)	20
Figure 8 - Agile Development Workflow(41)	29
Figure 9 - DevOps lifecycle (52)	30
Figure 10 - Agile DevOps lifecycle (52).....	31
Figure 11 - High Level Architecture Map	32
Figure 12 - Class Diagram.....	34
Figure 13 – Cloud Firestore Database Schema Diagram.....	35
Figure 14 - Use Case Diagram	36
Figure 15 - Sequence Diagram	37
Figure 16 - Nielsens 10 Heuristics (55).....	39
Figure 17 - Login Class Code Snippet	43
Figure 18 - Authentication Service Code Snippet	44
Figure 19 - RegisterAccount Class Code Snippet	45
Figure 20 - Account Homepage Code Snippet	46
Figure 21 - Emotional Sentiment API Code Snippet.....	47
Figure 22 - Emotional Sentiment Prediction Code Snippet	48
Figure 23 - Login Page Screenshot.....	49
Figure 24 - Registration Page Screenshot	50
Figure 25 - Account Dashboard Screenshot.....	51
Figure 26 - Interim GANTT Chart.....	54
Figure 27 - Final GANTT Chart.....	54

1. Introduction

1.1. Project Background

In a 2018 publication from the National Library of Medicine, it is noted that the potential of mobile apps in providing effective mental health interventions is considerable. With a shortage of psychiatrists globally and limited access to mental health care in rural areas, these apps have emerged as a practical solution to address the treatment gap in mental health. The utilization of technology has the capacity to revolutionize the delivery and accessibility of mental health treatment, emphasizing the need for the combined mobilization of science, regulation, and design to facilitate this transformative process. (1)

The concept for MyTherapyPal emerged from insightful conversations with two significant figures in my life—my mother, a seasoned psychotherapist, and my partner, an expert in psychology. These discussions shed light on the crucial need for clients to actively monitor their progress and work towards their goals within and beyond traditional therapy sessions. While traditional therapy undeniably has its merits, maintaining client engagement between appointments can be challenging. MyTherapyPal steps in to bridge this gap, providing a dynamic, interactive platform that empowers clients to take charge of their mental health journey and cultivate a personal sense of responsibility.

The timeliness and importance of this project are underscored by the growing demand for accessible mental health care and the widespread prevalence of mental health challenges. In our modern, tech-savvy world, where smartphones are practically an extension of ourselves, the development of a therapy app like MyTherapyPal feels not just relevant but essential. It's a response to the evolving needs of individuals seeking effective and user-friendly mental health support.

1.2. Project Description

The MyTherapyPal project sets out to serve as a solution for the need for improved client engagement between therapy sessions and to utilize modern technology to improve patient well-being and outcomes. The application will be accessible through multiple platforms such as mobile (Android & iOS), web and desktop (Windows & MacOS) to ensure that the service is accessible to as many users as possible. The project aims to achieve this goal by using the frontend development framework Flutter, which is a frontend UI framework developed by Google that natively compiles applications for mobile, web and desktop from a single codebase written in Dart programming language.

The application will provide features such as user accounts with secure authentication and account creation, progress tracking by way of journaling/note taking functionality tied to a calendar, day to day mood tracking, which the user will self-report, therapist listings with extensive information on therapists available in locality, mapping function to find and display local therapists, secure end-to-end encrypted chat for communication between client and therapist, secure video/phone calling for virtual therapy sessions, payment processing so clients can pay their therapists through the application, the ability for therapists to review clients notes, assign tasks and provide feedback,

natural language processing models for two features, the first being text summarization and classification from clients notes so that a therapist can see a short summary of their clients weekly notes and any inferences from those notes that the NLP model may detect, and the other feature being a note expansion feature for therapists who may write notes in short hand during therapy sessions.

The application will need to have a strong emphasis on security, especially in the realm of healthcare where maintaining the strict confidentiality is essential. By prioritizing security measures, the goal is not only to safeguard sensitive medical information but also to ensure compliance with GDPR regulations. This includes robust encryption protocols, access controls and authentication software to create a robust defence against any potential threats.

There will be an emphasis on a user-centric approach, prioritizing the user experience, incorporating feedback, and evaluating accessibility and ease of use. Evaluation will be conducted by providing working prototypes to mental healthcare professionals to test and provide feedback through the use of google forms to discover any flaws in the user experience.

1.3. Project Aims and Objectives

Overall aim and some milestones along the way to achieve the aim

- Build a cross-platform application that is accessible and user-friendly.
- Progress tracking features (daily notes, mood tracking).
- User accounts.
- Therapist listings.
- Therapist Map (find local therapists).
- Secure text chat.
- Secure video/phone calling (virtual therapy sessions).
- Payment processing.
- Natural language processing models making use of AI technology to summarize client notes and expand therapist notes.

1.4. Project Scope

This project does not intend to replace traditional therapy, instead, it complements and enhances the therapeutic process by addressing the challenge of maintaining client engagement between sessions. MyTherapyPal is not a diagnostic tool and does not aim to replace professional medical advice. It does not seek to replace the role of therapists but rather empower them with tools for more effective communication and engagement with their clients.

While the application incorporates natural language processing models, it is not a substitute for the nuanced understanding and empathy provided by human therapists. MyTherapyPal does not compromise on security and confidentiality, ensuring the protection of sensitive medical information, but it is not a comprehensive cybersecurity system beyond the scope of healthcare data. The project is not a one-size-fits-all solution, acknowledging that mental health needs are diverse, and individualized therapeutic approaches remain crucial.

MyTherapyPal is not static, it recognizes the need for continuous improvement based on user feedback but is not designed to replace the ongoing evolution and adaptability inherent in the field of mental health care.

1.5. Thesis Roadmap

The following report starts with literature review which provides a comprehensive survey of the existing research and technologies relevant to the MyTherapyPal project, establishing the context and justifying the necessity of the project. The next chapter is about the system design, and it outlines the architectural design of the MyTherapyPal system, detailing software and hardware component selection, their interactions, and the design rationales. The testing and evaluation chapter presents the methodologies used for testing the system's functionality and performance, along with a thorough evaluation of the test results.

This is followed by the prototype development section which describes the process of developing a working prototype of the MyTherapyPal system and the implementation of a primitive proof-of-concept for the artificial intelligence features of the application along with the challenges encountered. The final chapter focuses on the limitations and challenges faced during the course of the interim stage of this project and proposes potential improvements and directions for future work.

2. Literature Review

2.1. Introduction

In this chapter, we delve into a comprehensive exploration of existing solutions and research within the realm of mental health interventions and technology applications. The purpose is to gain valuable insights and inform the development of MyTherapyPal, recognizing the significance of building upon established knowledge and experiences.

By critically examining the current landscape, we aim to identify gaps, successes, and potential pitfalls in the existing literature. This exploration serves as a foundation for shaping the innovative features and functionalities of our application. In navigating the expansive terrain of mental health technology, we seek to understand the strengths and limitations of prior endeavours, providing a context for the unique contributions MyTherapyPal aspires to make.

Through this literature review, we aim to not only acknowledge the strides made in the field but also to pinpoint areas where our project can offer novel solutions and foster advancements in mental health care technology.

2.2. Alternative Existing Mental Health Assistance Applications

Some alternative existing applications that focus on mental health and in particular progress and mood tracking I have found include:

- Quenza – A digital health solution that helps therapists administer assessments and track patient progress over time.
- Daylio – A mood tracker and micro-diary with statistics and charts for mood trends.
- Moodnotes – A thought journal and mood diary with insights into mood patterns and incorporating CBT (Cognitive Behavioural Therapy) principles.
- eMoods – Mood tracking application for people with bipolar disorder.
- Theratrak – A digital platform enabling allied health therapists to track participant progress in between face-to-face sessions.

2.2.1. Quenza

Quenza is a tool designed for professionals in fields like therapy, coaching, counselling, and social work. Its goal is to make it easy for them to create various engaging activities such as worksheets, exercises, and surveys, which can be shared digitally with their clients. These activities can be grouped into care pathways—timed series of interactive exercises sent to clients via email or in-app notifications with automated reminders. The client dashboard allows practitioners to keep track of client progress. One of the main objectives of the Quenza application is to keep clients involved between sessions, providing them with personalized homework, reflections, meditations, and

interventions. Other use cases include educating and onboarding clients, evaluating progress or session effectiveness, and offering inspiration.(2)

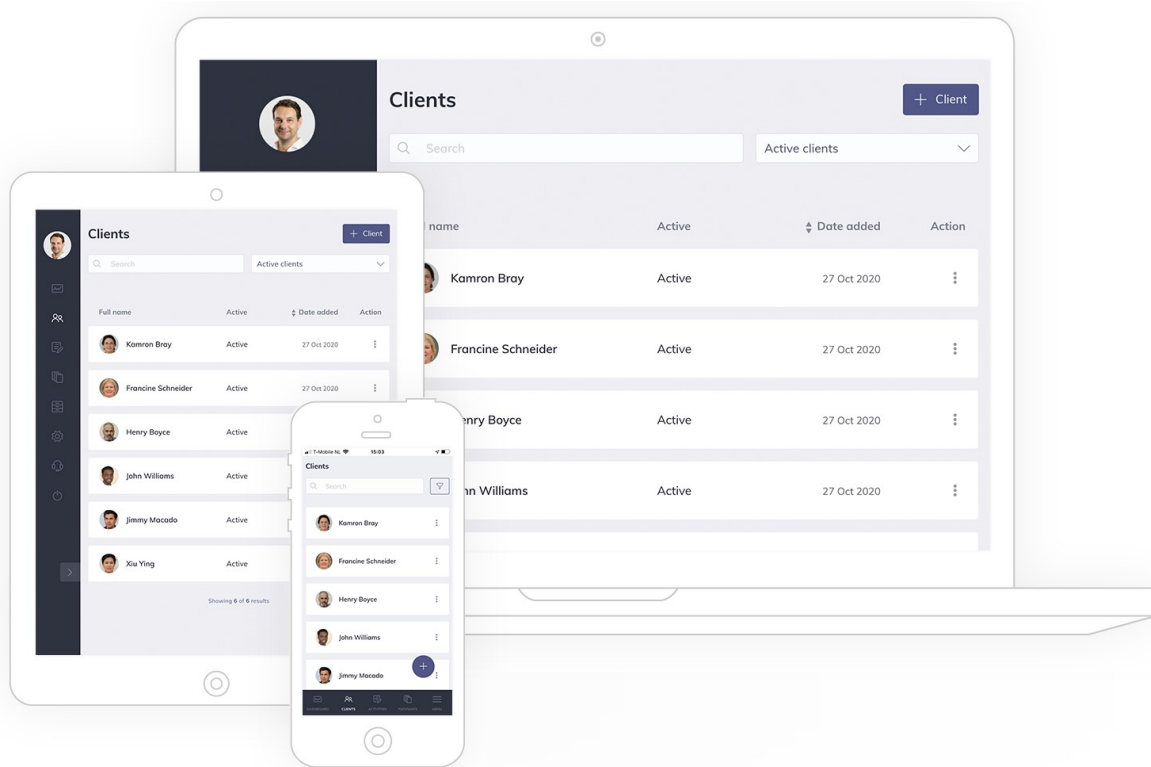


Figure 1 - Quenza Mobile App Screenshots (3)

2.2.2. Daylio

Daylio is a versatile micro-diary app designed to assist users in tracking various aspects of their lives effortlessly. Acting as a personal companion for goals such as fitness, mental health, nutrition, and gratitude, the app allows users to record daily activities and moods without the need for extensive typing. Its three guiding principles include promoting mindfulness for happiness and self-improvement, validating the impact of new habits, and providing a seamless, obstacle-free user experience.

Functionality-wise, Daylio operates on a simple two-step entry process, enabling users to pick their mood, add activities, and include optional notes. The app then compiles this data into statistics and a calendar, facilitating a better understanding of daily habits. Users can review their entries through charts and calendars, sharing them with friends if desired. The app offers features such as a large icon database for personalized activities, customizable moods using emojis, and in-depth statistics on a weekly, monthly, or yearly basis.

Daylio encourages the development of reflection as a daily habit, helping users discover what brings them happiness. With customizable colour themes, dark mode, and the option to set reminders, the app strives to make journaling a seamless and enjoyable experience. Privacy and security are prioritized, with data stored locally on the user's phone and optional backups to private cloud storage. The app's commitment to transparency ensures that users maintain control over their data, and measures such as PIN lock and encrypted channels for backups contribute to a secure journaling environment. (4)

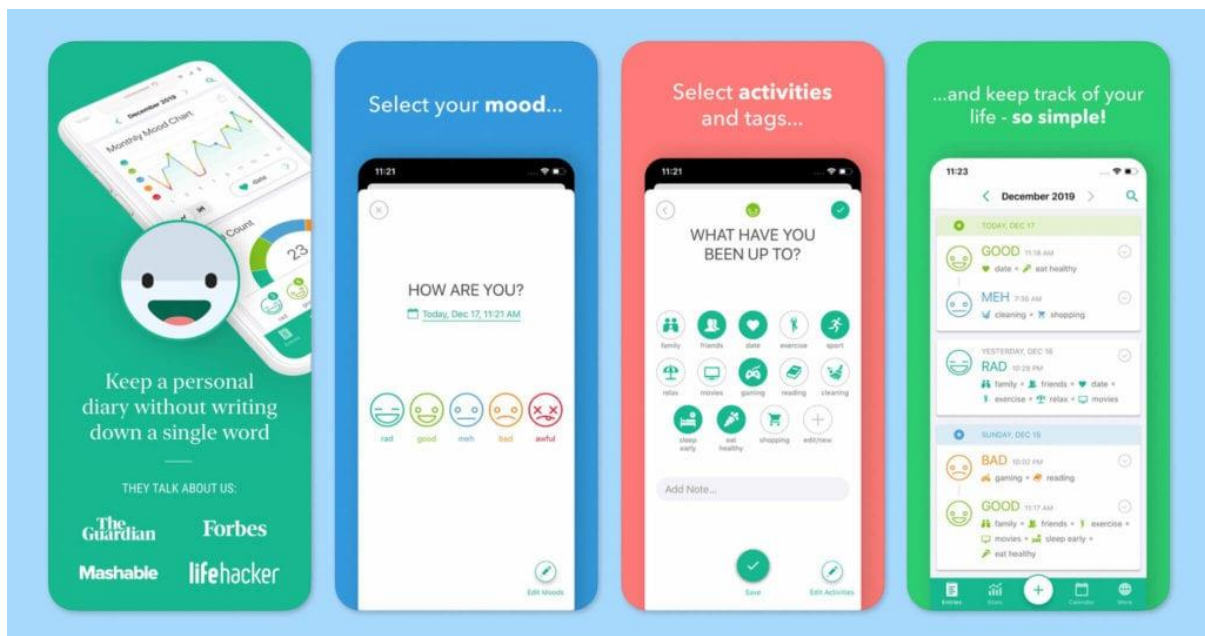


Figure 2 - Daylio Mobile App Screenshots (5)

2.2.3. Moodnotes

Moodnotes is a user-friendly mood tracker and journaling app designed to capture and enhance user moods while promoting positive thinking habits. Developed by design experts and clinical psychologists, the app draws on the scientifically supported principles of cognitive behaviour therapy (CBT) and positive psychology to empower users in tracking their moods, identifying triggers, and fostering well-being.

Key features of Moodnotes include the ability to track moods over time, automatic face scanning using the device camera, memory preservation through photos or images, and access to self-awareness articles crafted by mental health experts. The app facilitates the development of healthier thinking habits by educating users about common thinking traps and providing tools to avoid them. Additionally, Moodnotes aims to reduce anxiety, enhance well-being, and bring new, constructive perspectives to various situations.

Moodnotes Premium, available for members, unlocks additional features such as the ability to make unlimited entries per day, monitor personal statistics and mood insights, add notes to enrich entries, access the "Notable insights" section for a deeper understanding of moods, and discover more full-text articles for self-reflection. (6)



Figure 3 - Moodnotes Mobile App Screenshots (7)

2.2.4. eMoods

eMoods is a highly rated mood tracking app designed for individuals managing Bipolar, Anxiety, and Depression. The app prioritizes user privacy by exclusively storing data on the device and avoiding cloud storage or external connections without explicit consent. It facilitates easy tracking of daily mood variations, sleep patterns, medications, and other symptoms associated with mood disorders. Notably, eMoods allows users to track medication changes, providing insights for informed discussions with healthcare professionals.

An essential feature of eMoods is its capacity to generate monthly PDF reports for users to share with their healthcare providers. This aids in communication and collaboration to identify triggers or events contributing to mood episodes or relapses. The app also includes in-app graphs and printable charts, enabling users to visualize trends and patterns for a deeper understanding of their symptoms between appointments. Customization options, such as adding emojis or language characters to medication names and free-form notes, enhance user flexibility.

Privacy is a core focus, with eMoods ensuring that all logged information remains securely stored on the user's device. For those seeking additional functionalities, the optional Pro version offers features like tracking custom symptoms and storing multiple timestamped notes per day through auto-renewing subscriptions with transparent terms. The app's commitment to user satisfaction and flexibility is evident in its clear subscription model, allowing users to cancel at any time. In summary, eMoods serves as a comprehensive and privacy-conscious tool for effective mood tracking and management. (8)

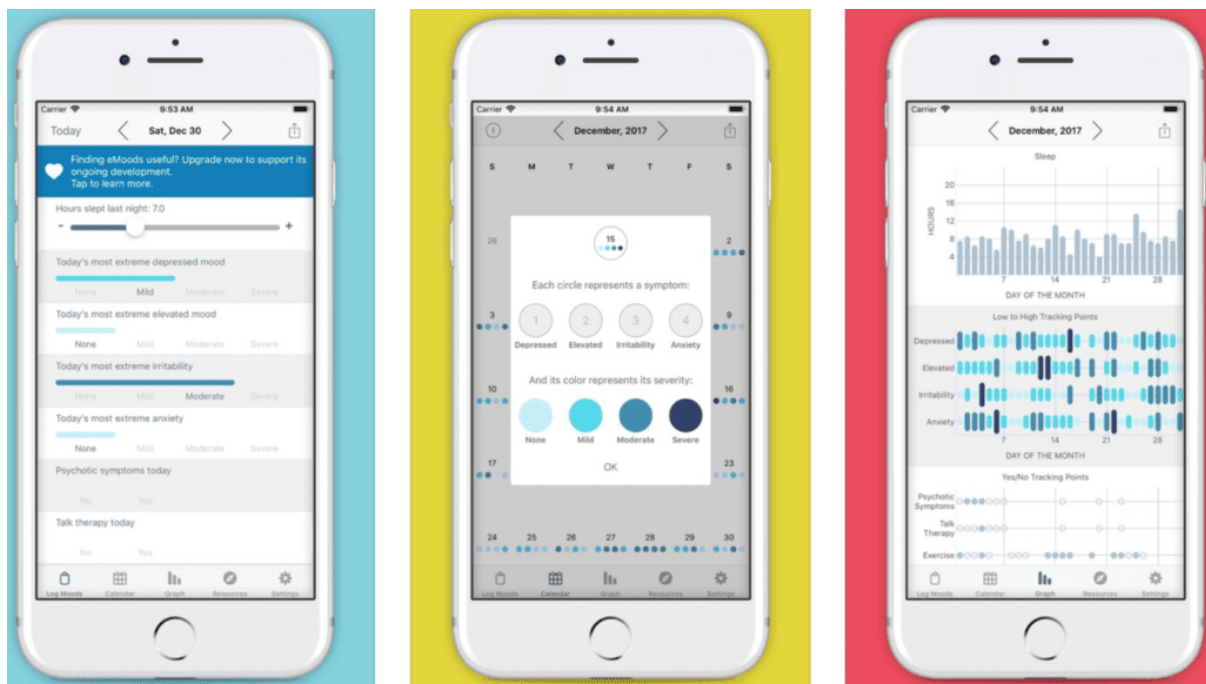


Figure 4 - eMoods Mobile App Screenshots (9)

2.2.5. Theratrak

Theratrak is a mobile app that allows occupational therapists to capture meaningful information from therapy sessions and create customised home programs for children. Therapists have access to a database of therapeutic activity scaffolds which they can easily access within a therapy session. Therapists then attach photos and videos to create highly personal home programs, individual to each client's needs. This allows the therapist to grade the activities to the child's abilities and give specific verbal and written notes about task completion.

Therapists can use this tool concurrently with therapy sessions, helping to reduce the amount of lost paperwork or forgotten material. Theratrak gives the patient digital access to the home program the therapist prescribes, so that they continue with their treatment, better informed about their program, enabling them to take charge of their health. (10)

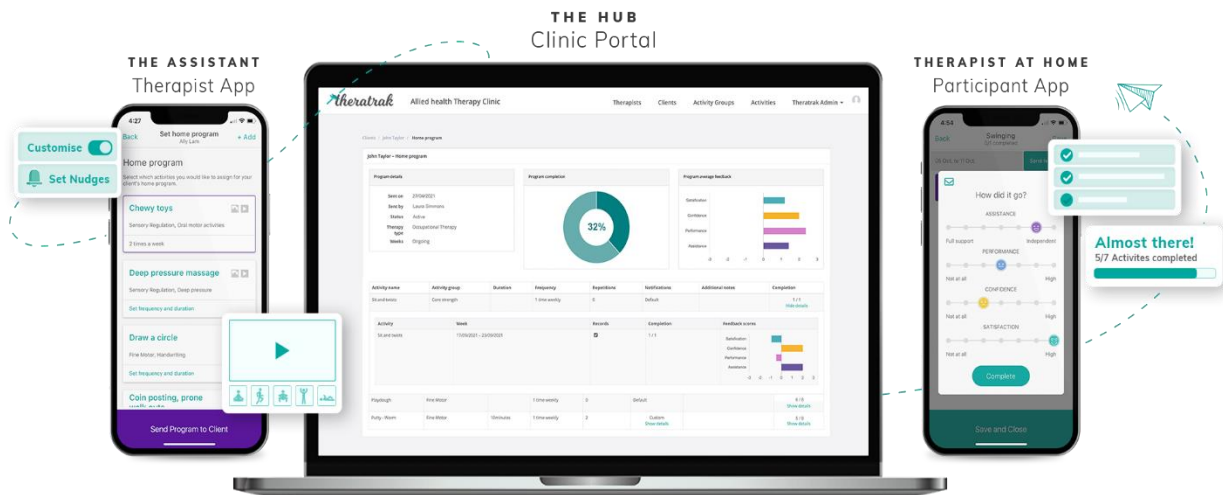


Figure 5 - Theratrak App Screenshots (11)

2.3. Technologies Researched & Selection

2.3.1 Operating Systems

In this section we will look at various operating systems that could be suitable as platforms for the application. A strong understanding of the target audience in terms of demographics and operating system popularity will weigh heavily in the operating system selection decision making process. Other factors that will need to be considered are the type of application we are developing, hardware considerations, development tools and testing strategies.

2.3.1.1 Android

Android is an open-source operating system developed by Google, widely used for mobile devices such as smartphones and tablets. Known for its versatility and robust app ecosystem, Android provides a user-friendly interface and serves as the foundation for a diverse range of electronic devices.

In choosing Android as one of the operating systems for MyTherapyPal, we prioritize its widespread adoption and extensive user base. Android's global prominence ensures that the app has the potential to reach millions of users. The decision to incorporate Android aligns with a broader strategy of catering to diverse user preferences, and by leveraging Flutter's cross-platform capabilities, the app is poised to deliver a consistent and high quality experience across multiple operating systems. This approach not only maximizes the app's accessibility but also underscores its adaptability in a multi-platform landscape. (12)

2.3.1.2 iOS

iOS is a mobile operating system developed by Apple Inc. It is the foundation that powers Apple's mobile devices, including iPhones, iPads, and iPod Touches. iOS is known for its user-friendly interface, security features, and seamless integration with other Apple products and services.

MyTherapyPal, being a healthcare-related application, could benefit from developing a version for iOS due to several reasons. Firstly, iOS has a reputation for providing a secure and controlled environment, ensuring the protection of sensitive health data. This aligns well with the privacy and confidentiality requirements often associated with healthcare applications. Apple's App Store is a well-established platform with a large user base. By releasing a version of MyTherapyPal on iOS, we gain access to a significant market share and potential users who prefer or exclusively use Apple devices. iOS devices are also known for their consistent hardware and software ecosystem. This uniformity makes it easier for developers to create and optimize applications for a standardized set of devices, resulting in a smoother and more reliable user experience. (13)

2.3.1.3 Windows

Windows is a family of operating systems developed by Microsoft. It is one of the most widely used operating systems for personal computers, ranging from desktops and laptops to tablets and hybrid devices. Windows provides a graphical user interface and supports a wide range of software applications. Developing a version of MyTherapyPal for the Windows platform can be a strategic decision for several reasons. Firstly, Windows has a significant market share in the personal computing space. By targeting this platform, you can reach a diverse user base that includes individuals using various types of Windows-powered devices. Windows is commonly used in professional and enterprise environments. This presents an opportunity for MyTherapyPal to cater to healthcare professionals, clinics, or organizations that rely on Windows-based systems for their daily operations. Compatibility with Windows can facilitate the integration of MyTherapyPal into existing healthcare workflows. (14)

2.3.1.4 MacOS

MacOS, formerly OS x, is an operating system developed by Apple Inc. for its line of Macintosh computers. It's built on a Unix-based foundation, providing a stable and secure environment for

users. MacOS offers a user-friendly interface, advanced graphics capabilities, and seamless integration with other Apple devices and services. While MacOS would have a significantly smaller market share over Windows, it is still the second most popular desktop operating system globally. Since we are using flutter, it makes it a very simple process to compile for this operating system too so the decision to include MacOS as a platform was an easy one. (15)

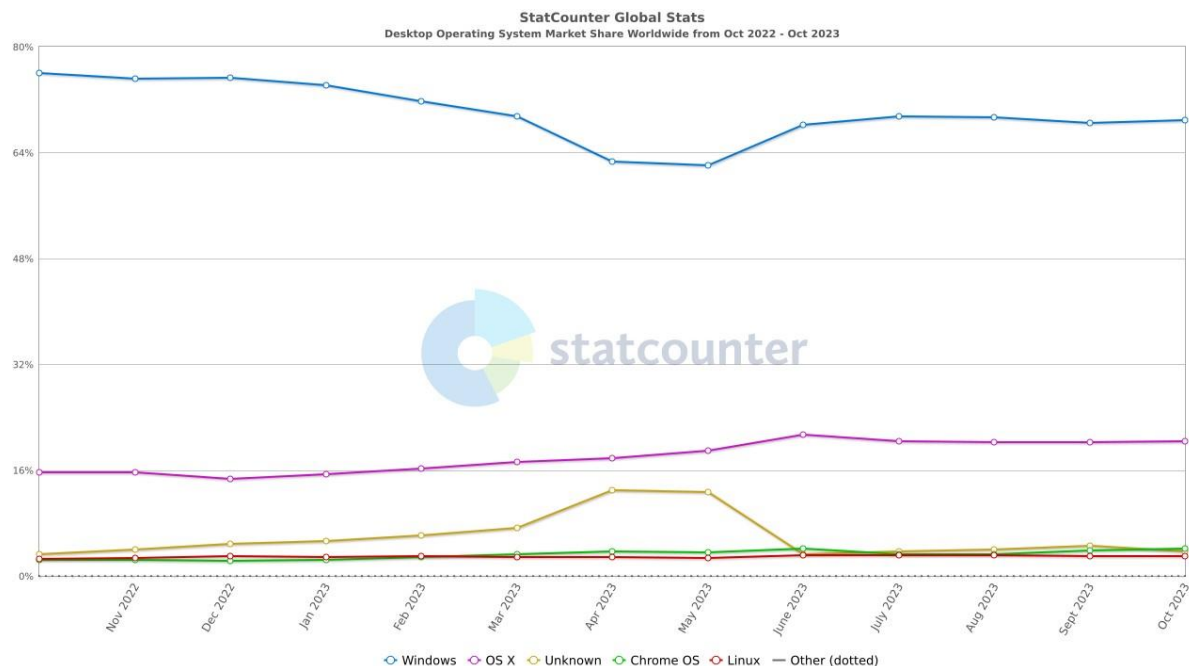


Figure 6 - Desktop Operating System Market Share Worldwide (16)

2.3.2 Databases

2.3.2.1 Firebase Cloud Firestore

Google's Cloud Firestore in Firebase is a NoSQL document based database stored in the cloud which facilitates the efficient storing, synchronization and querying of data at a global scale. The database is structured with collections and documents that can be used for the building of hierarchies to store related data and quickly retrieve necessary data using expressive queries. Queries are scalable with the size of the result set rather than the size of the data set to maximize scalability.

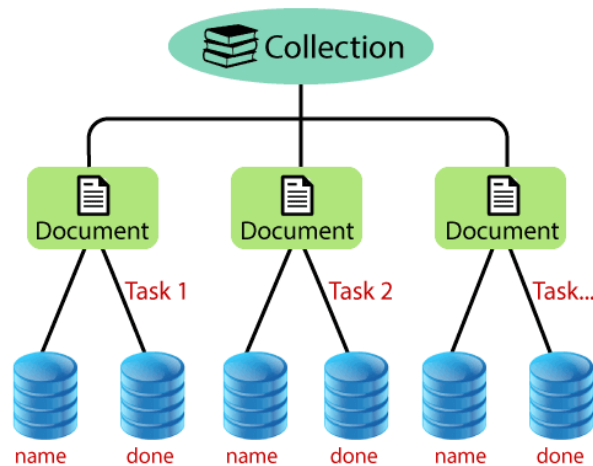


Figure 7 - Cloud Firestore Data Organization (17)

2.3.2.2 MariaDB

MariaDB is an open source fork of MySQL relational database management system (RDBMS). It is a very popular choice among developers as a backend database for a range of use cases and provides a powerful solution for the management of large volumes of data. It has strong features suited to transactional and analytical workloads. MariaDB also supports a wide range of storage engines which enables users to tailor the database to a large variety of use cases. MariaDB has an active community that is continuously developing, improving, and contributing to the software.

2.3.2.3 PostgreSQL

PostgreSQL is an open source object-relational database system with over 35 years of active development. An object-relational database (ORD) is a type of database management system (DBMS) that combines elements of both relational databases and object-oriented databases. In a traditional relational database, data is organized into tables with rows and columns, and relationships between tables are established through keys. On the other hand, object-oriented databases treat data as objects, each containing both data and the procedures that can operate on the data. Object-relational databases seek to bridge the gap between these two approaches by incorporating object-oriented features into a relational database.

2.3.2.4 Selection (Firebase Cloud Firestore)

When selecting a database solution, it is crucial to carefully consider the compatibility and seamless integration with all other components within the software system. Given our decision to adopt Flutter as the frontend technology, Firebase Cloud Firestore emerges as the logical choice for our database solution.

2.3.3 Frontend Framework

When faced with the task of selecting the frontend programming languages and frameworks for a cross-platform application, several options present themselves, each with its set of strengths and considerations.

2.3.3.1 *React Native*

React Native, leveraging JavaScript and React, is a widely embraced choice. Its appeal lies in the vast pool of web developers who can seamlessly transition to mobile app development. With a strong community backing, React Native supports efficient code sharing between platforms, making it a pragmatic choice for those looking to optimize development time. (18)

2.3.3.2 *Xamarin*

Xamarin, on the other hand, utilizes C# and .NET, providing an advantage for developers familiar with Microsoft technologies. The framework allows for tight integration with existing C# codebases and offers access to a broad range of native APIs. Xamarin's performance is notable, and its ability to deliver a native user experience has led to its wide use in enterprise. (19)

2.3.3.3 *Selection (Flutter)*

Ultimately the decision was made to use Flutter, developed by Google. Flutter takes a distinctive approach by employing Dart as its programming language. Dart is known for its simplicity and efficiency, providing a strong foundation for building scalable applications. Flutter's widget-based architecture ensures a consistent and visually appealing user interface across different platforms. The "hot reload" feature in Flutter significantly enhances the development workflow by enabling real-time code changes and rapid debugging.

One of the key considerations that swayed this decision towards Flutter was its commitment to native-like performance across both iOS and Android platforms. The framework's single codebase approach reduces development overhead, ensuring that changes made on one platform reflect seamlessly on the other. The support from the growing Flutter community and the availability of comprehensive documentation added to its appeal. While React Native and Xamarin have their merits, the decision to opt for Flutter was influenced by Dart's efficiency, Flutter's widget-based architecture for UI consistency, and the productivity gains offered by the "hot reload" feature. The framework's single codebase advantage and the support from a vibrant community aligned well with the project's goals, making Flutter the optimal choice for this cross-platform application. (20)

2.3.4 Backend Framework

When considering backend technologies for a cross-platform application, various options exist, each with distinct strengths. Popular choices include Express.js with Node.js, Django with Python, Spring Boot with Java, Ruby on Rails, and Firebase.

2.3.4.1 Django with Python

Django with Python: Django, powered by Python, is renowned for its simplicity and readability. Its "batteries-included" philosophy provides a comprehensive set of features out of the box, making it efficient for rapid development, especially for projects with complex data models. Django's built-in admin panel, ORM, and security features contribute to its popularity. (21)

2.3.4.2 Ruby on Rails

Ruby on Rails: Ruby on Rails is a web application framework based on the Ruby programming language. It is recognized for its developer-friendly syntax and convention over configuration principles, promoting rapid development. Ruby on Rails is often chosen for projects where simplicity and speed are paramount. (22)

2.3.4.3 Firebase

Firebase: Firebase, a comprehensive solution from Google, stands out for its full-stack capabilities. It includes services like Realtime Database, Authentication, and Cloud Functions. Firebase's seamless integration with Flutter, the chosen frontend framework, ensures synchronized data updates across devices in real-time, making it a powerful choice for cross-platform applications.

2.3.4.4 Selection (Firebase)

In this case, the decision tilted towards Firebase, a comprehensive solution from Google. Firebase not only stood out for its real-time database and secure authentication but also for its exceptional synergy with Flutter, the chosen frontend framework. The seamless integration between Firebase and Flutter will significantly streamline the development process, as both technologies are designed to work harmoniously together. Firebase's real-time features align well with Flutter's reactive nature, making it an ideal choice for ensuring synchronized data updates across devices in real-time.

The Firebase-Flutter combination offers a cohesive development experience, with Firebase providing a robust backend and Flutter delivering a consistent UI across platforms. This interoperability played a crucial role in the decision, as it facilitated a more efficient and unified development workflow for the cross-platform application. While options like Express.js, Django, Spring Boot and Ruby on Rails

are noteworthy, Firebase's exceptional compatibility with Flutter became a decisive factor. The combined strengths of Firebase and Flutter ensured a streamlined development process and a harmonious integration between the backend and frontend of the cross-platform application. (23)

2.4. Other Research you've done

2.4.1 Mental Health & Psychotherapy

2.4.1.1 Mobile technology for mental health assessment

The study "Mobile technology for mental health assessment" by Kien Hoa Ly & Gerhard Andersson (2016) sets out to establish how mobile technology can be used to improve the diagnostics, delivery, and outcome monitoring in mental health treatment.(24) The study states that until recently, mental health clinicians have had to rely solely on clinical judgement and patient self-reporting in order to diagnose illness and deliver optimal treatments, this is despite decades of research on the improvement and refinement of clinical observation.(24) One of the main issues with this style of diagnosis is that it relies upon retrospective recollection of a patient's symptoms and functioning during clinical interviews, the study states that these kinds of reports tend to be biased and inaccurate.(24)

When it comes to technology-based assessment methods, the widespread ownership of smartphones and wearable sensors allows for the collection of real-time physiological, social, emotional, and behavioural data with minimal client burden.(24) This data, including electronic self-reporting, performance measures, sensor data, and social media data, can inform treatment decisions and monitor response over time. Mobile software applications process and present this data to clinicians and clients, offering valuable insights into therapeutic interventions targeting physical activity, social connection, cognitive function, and symptom burden.(24)

The conclusion of this study goes on to say that the future of psychiatric assessment lies in the real-time collection of client mood and behavioural data through mobile platforms. This approach offers more efficient and timely care, allowing clinicians to track progress based on real-time information rather than relying on retrospective client reports.(24) Ongoing projects like ICT-4Depression showcase the potential of combined mobile data collection, utilizing various technologies to predict and monitor mental health states in real-time.(24) While some technologies are ready for use, addressing privacy and ethical concerns, as well as ensuring interoperability with electronic health records, is crucial for the widespread adoption of these innovative assessment methods.(24)

2.4.1.2 Do mental health mobile apps work: evidence and recommendations for designing high-efficacy mental health mobile apps by Pooja Chandrashekar

The study "Do mental health mobile apps work: evidence and recommendations for designing high-efficacy mental health mobile apps" by Pooja Chandrashekar (2018) starts out by proposing that

smartphone-based apps may expand access to mental health treatment and states that the number of mobile health apps that focus on mental health has increased at a rapid pace with a 2015 World Health Organization (WHO) survey of 15,000 mobile health applications revealing that 29% of those applications focused on mental health diagnoses, treatment or support.(1,25) The study also discusses the utility of mental health applications for psychological treatment, where there exists an abundance of mobile applications that target a wide range of psychological disorders and can vary in design and functionality. The study states that according to NIMH there are six categories it uses to classify mental health applications on functionality which are self-management, cognition improvement, skills training, social support, symptom tracking and passive data collection.(1,26)

2.4.1.3 Cognitive-Behavioural Therapy (CBT)

Cognitive Behavioural Therapy (CBT) is a psychological intervention that has proven effective in addressing a variety of issues such as depression, anxiety disorders, substance abuse issues, relationship problems, eating disorders, and severe mental illness. The fundamental premise of CBT is that psychological difficulties are partly rooted in unhelpful thought patterns and maladaptive behaviours. CBT assists individuals in managing and enhancing their mental health by dissecting complex issues into more manageable components. It operates on the principle that thoughts, emotions, physical sensations, and actions are all interconnected, and that negative thoughts and feelings can create a vicious cycle. By altering these negative cycles, CBT can help improve an individual's emotional state.(27,28)

One of the key aspects of CBT is empowering individuals to become their own therapists. Through in-session exercises and “homework” tasks, individuals are equipped with coping skills that enable them to modify their own thoughts, emotions, and behaviours. A wealth of research indicates that CBT results in significant improvements in functioning and quality of life. In many studies, CBT has been shown to be as effective as, or more effective than, other psychological therapies or psychiatric medications.(29)

2.4.1.4 Computer-Assisted CBT and Mobile Apps for Depression and Anxiety

The objective of this review was to evaluate the effectiveness of computer-assisted cognitive-behaviour therapy (CCBT) and mobile applications in delivering or augmenting treatment for depression and anxiety, and to provide recommendations for their clinical use.

Research has shown that CCBT is effective when used in conjunction with a clinician or other helping professionals. However, when used as standalone treatments, computer programs have typically shown lower efficacy or ineffectiveness. Numerous mobile apps have been developed claiming to be beneficial for depression and/or anxiety, but their evaluation and recommendation require considerable caution. Research on these mobile apps is still in its nascent stages.

Several well-established CCBT programs have undergone multiple randomized, controlled trials and have proven to be effective. These programs, which demonstrate adequate quality, security, and efficacy, are suitable for clinical practice. Mobile apps, with their easy portability and immediate access to coping strategies, may be useful for augmenting treatment. However, clinicians need to ensure that the apps they select for clinical use are reliable and maintain integrity.

2.4.2 Machine Learning

2.4.2.1 Natural Language Processing Models

During the course of the literature review we looked at several natural language processing models that would be suitable for sentiment analysis and our CBT chatbot. In the following sub sections we will discuss each notable model along with the merits, and the pitfalls of each model.

BERT, or Bidirectional Encoder Representations from Transformers, is an open source natural language processing (NLP) pre-training technique that was developed by Google in 2018(30). It's designed to understand the context of words in a sentence by considering the surrounding words. Unlike previous models that read text input from left to right or vice versa, BERT looks at the entire sentence to predict missing words. This bidirectional approach helps BERT grasp the nuances of language and context, making it particularly effective for various NLP tasks like question answering, sentiment analysis, and text prediction. It's based on the Transformer architecture, a type of neural network architecture that has proven very successful in NLP tasks.(31)

The LLaMA NLP model is a collection of foundation language models ranging from 7 to 65 billion parameters, which are trained on trillions of tokens from a publicly available dataset (32). It is designed to help researchers advance their work in natural language processing (NLP), which is a subfield of AI that deals with understanding and generating natural language(33). LLaMA stands for Large Language Model Meta AI, and it is an open-source initiative by Meta AI, a company that is part of Meta (formerly Facebook)(28). LLaMA works by taking a sequence of words as an input and predicting the next word to recursively generate text. It can handle 20 languages with Latin and Cyrillic alphabets, and it can be fine-tuned for various tasks, such as text summarization, question answering, chatbot and sentiment analysis(34).

Another model that was considered is ChatGPT, which is a large language model-based chatbot developed by OpenAI(35). It interacts in a conversational way, answering follow-up questions, admitting its mistakes, challenging incorrect premises, and rejecting inappropriate requests. It is built upon either GPT-3.5 or GPT-4, both of which are members of OpenAI's proprietary series of generative pre-trained transformer (GPT) models. It is fine-tuned for conversational applications using a combination of supervised and reinforcement learning techniques.(35)

When it comes to sentiment analysis, ChatGPT offers several benefits. Its ability to understand and generate human-like text makes it a flexible tool for sentiment analysis in various industries. It can analyse large amounts of text data quickly and efficiently. It can analyse the data context, considering domain-specific knowledge, industry jargon, and the unique relationships between businesses and customers. It can also generate synthetic text data with various sentiment labels, which can be used to augment existing training datasets or create new ones.(35)

2.4.2.2 Sentiment Analysis

Sentiment analysis is a powerful tool in the field of natural language processing that allows us to extract emotional context from text. It involves determining whether a piece of writing is positive, negative, or neutral. When applied to text summarization, sentiment analysis can provide a more nuanced summary by not only condensing the information but also capturing the emotional tone of

the text. This is particularly useful for the projects use cases where understanding the sentiment is as important as understanding the content. By integrating sentiment analysis, we can create summaries that maintain the original emotional context, providing a more comprehensive understanding of the text, providing therapists with deeper insight into the mental state of their clients.

2.4.2.3 CBT Chatbot with Natural Language Processing Models

BERT, a model developed by Google, is adept at understanding the context of user inputs in a chatbot, making it a powerful tool for such applications. However, it's worth noting that BERT only encodes and doesn't generate responses, necessitating additional components.(30) On the other hand, LLaMA can be fine-tuned on specific data like patient-doctor dialogues, enhancing its understanding of medical advice. This makes it potentially suitable for a CBT therapy chatbot, although information on its use in this context is limited.(33)

2.4.2.4 Mental Health Datasets for AI research

Mental health datasets, such as those derived from social media platforms like Twitter, can be invaluable resources for training natural language processing (NLP) models for emotional and sentiment analysis. These datasets often contain a wealth of information about individuals' emotional states and sentiments, expressed through their posts or tweets. By training NLP models on these datasets, we can teach them to recognize and understand the nuances of human emotion and sentiment. For instance, a model could learn to identify negative sentiments associated with depression or anxiety, or positive sentiments reflecting happiness or satisfaction. This could enable the development of more sophisticated tools for monitoring mental health trends, providing timely support, and contributing to the broader field of mental health research.

2.5. Existing Final Year Projects

During the course of this project, some past projects were researched and one in particular stood out as being closely related to the problem domain of this project. Suaimeas: A responsive web application for anxiety management and wellbeing by Ben McCormack (2023), which set out to investigate the best way to create an application that would allow users to use wellbeing and mindfulness techniques to better manage their anxiety. The outcome of the project was the development of a feature rich and responsive web application that is helping users manage their anxiety and allows mental health professionals to monitor their progress. The user management system and goal setting feature are particularly interesting, and from which our project may draw inspiration during the development stage.

Another past project that was researched is Anxiety Manager (incorporating Wearable Technology) by Katie Fitzgerald (2018). The projects aim was to design and develop a mobile application that would give individuals who suffer from anxiety with a new tool to track their anxiety attacks and also

learn more about their anxiety. The system had an emphasis on accessibility and user friendliness to cater to users of different IT skill levels. The paper states that current solutions are outdated in their approach of tracking anxiety attacks and that with a wearable sensor to monitor changes in a user's physical symptoms along with a mobile application, the system would provide much more valuable insights into the user's anxiety and better help them to manage their anxiety.

In the conclusion of this paper, the author declares that the project's objectives were met by developing a solution for mobile in Android that provides an easy way for users to track their anxiety with wearable technology. The user-friendly approach and accessibility considerations are excellent sources of inspiration for the development of MyTherapyPal.

2.6. Conclusions

The literature review has provided a comprehensive overview of the existing mental health assistance applications, the technologies researched and selected for the development of the project, the other research done on mental health and psychotherapy, and the machine learning techniques that will be applied to the project.

There are various alternative existing mental health assistance applications that offer different features and functionalities, such as Quenza, Daylio, Moodnotes, eMoods, and Theratrak. However, none of them provide a personalised and interactive chatbot that can deliver cognitive behavioural therapy (CBT) sessions to the users based on their mood and preferences.

The technologies researched and selected for the project are cross-platform Android, iOS, Web, Windows and MacOS as the operating systems, Firebase as the database and the backend framework, and Flutter as the frontend framework. These technologies were chosen based on their advantages, compatibility, popularity, and suitability for the project requirements and objectives.

The other research done on mental health and psychotherapy has provided a background knowledge and understanding of the concepts, theories, and practices of CBT, as well as the benefits and challenges of delivering CBT online. The literature review has also explored the ethical and legal issues related to the development and use of mental health assistance applications, such as data protection, confidentiality, consent, and duty of care. Additionally, the literature review has conducted a systematic review of the existing studies on the use of mobile apps to provide mental health treatment, and has evaluated their effectiveness, usability, and acceptability among the users and the therapists.

The machine learning techniques that will be applied to the project are natural language processing (NLP) and sentiment analysis. These techniques will enable the chatbot to understand and generate natural language, as well as to analyse the mood and emotions of the users based on their inputs. The literature review has also discussed the methods, tools, and datasets that will be used for implementing and evaluating these techniques.

The literature review has identified some gaps and limitations in the existing literature, such as the lack of empirical evidence on the effectiveness and user satisfaction of mental health assistance applications, the scarcity of studies on the use of machine learning for CBT chatbots, and the need for more research on the ethical and legal implications of online CBT. These gaps and limitations suggest some areas for future research and improvement.

The literature review has also established the significance and originality of the project, as it aims to develop a novel and innovative mental health assistance application that can provide personalised and interactive CBT sessions to the users through a chatbot powered by machine learning. The project will contribute to the advancement of knowledge and practice in the fields of mental health, psychotherapy, and computer science.

3. System Design

3.1. Introduction

In this chapter, we will discuss the design of the system and the research conducted on the relevant types of software development methodologies, design patterns and selections for this project. The system architecture will be explained and presented with the use of a high level architecture diagram, class diagrams, sequence diagrams and corresponding use cases.

3.2. Software Methodology

Software methodologies are structured approaches to software development that guide the planning, design, implementation, testing, and delivery of software products. Software methodologies help developers to organize their work, manage their resources, communicate with their stakeholders, and ensure the quality and reliability of their software. Different software methodologies have different advantages and disadvantages, depending on the nature, scope, and complexity of the software project. Some of the most popular software methodologies are Agile, DevOps, Waterfall, and Feature Driven Development.(37–41)

3.2.1 Agile

Agile is a software development methodology that emphasizes iterative, incremental, and collaborative delivery of working software. Agile aims to respond to changing customer needs and feedback, and to deliver value faster and more frequently. Agile is based on the Agile Manifesto, which outlines four values and 12 principles for Agile software development(42). Agile is not a single, fixed approach, but rather a family of frameworks and practices that share the same philosophy and values. Some of the most common Agile frameworks are Scrum, Kanban, Extreme Programming (XP), and Feature Driven Development (FDD)(43–45).

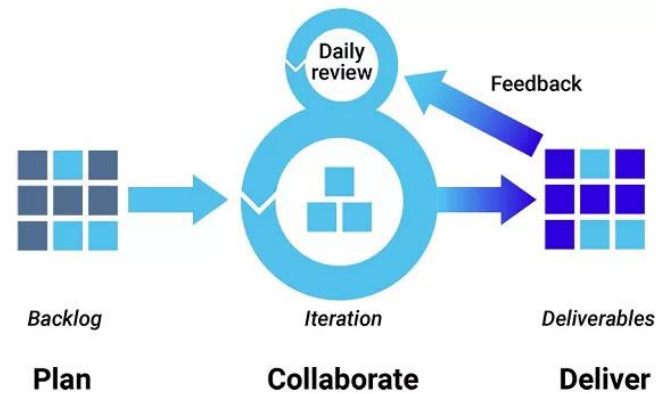


Figure 8 - Agile Development Workflow(41)

Agile software development is a customer-centric and iterative approach to software delivery. It starts with breaking down the project into user stories, which are small and specific descriptions of the desired features and functionalities(43). The user stories are then prioritized, estimated, and assigned to sprints, which are short and fixed periods of time for developing and testing the software. The team follows the chosen Agile framework and practices, such as daily stand-ups, pair programming, test-driven development, and continuous integration, to collaborate and communicate effectively with the customer and other stakeholders(44). At the end of each sprint, the team delivers a working product increment and demonstrates it to get feedback and suggestions for improvement. The team also conducts a retrospective meeting to review their performance and identify areas of improvement for the next sprint(45).

3.2.2 Waterfall

The waterfall methodology is a linear and sequential approach to project management that was first proposed by Winston Royce in 1970(46). It consists of five phases: requirements, design, implementation, verification, and maintenance(47). Each phase must be completed before moving on to the next one, and there is no overlap or iteration between them(48). The waterfall methodology is mainly used in software development projects where the objectives and specifications are clearly defined and stable from the beginning(49). It is also suitable for projects that require a lot of predictability and reliability(50).

3.2.3 DevOps

DevOps is a set of practices, tools, and a cultural philosophy that automate and integrate the processes between software development and IT teams. It emphasizes team empowerment, cross-team communication and collaboration, and technology automation(51).



Figure 9 - DevOps lifecycle (52)

DevOps aims to bring development and operations together to complete software development faster and more efficiently. DevOps is centred around CI/CD which stands for continuous integration and continuous deployment which introduces automation for continuously testing and deploying an application.(52)

3.2.6 Selection (Agile & Devops)

Agile and DevOps were chosen as the software methodologies for the MyTherapyPal project due to their ability to facilitate rapid, iterative development and deployment. Agile methodology, with its emphasis on iterative development, frequent delivery, and continuous feedback, aligns well with the project's need for adaptability and responsiveness to changing requirements. DevOps, on the other hand, complements Agile by bringing in practices like continuous integration, continuous delivery, and infrastructure as code, which help in faster and reliable delivery of features and automated testing.

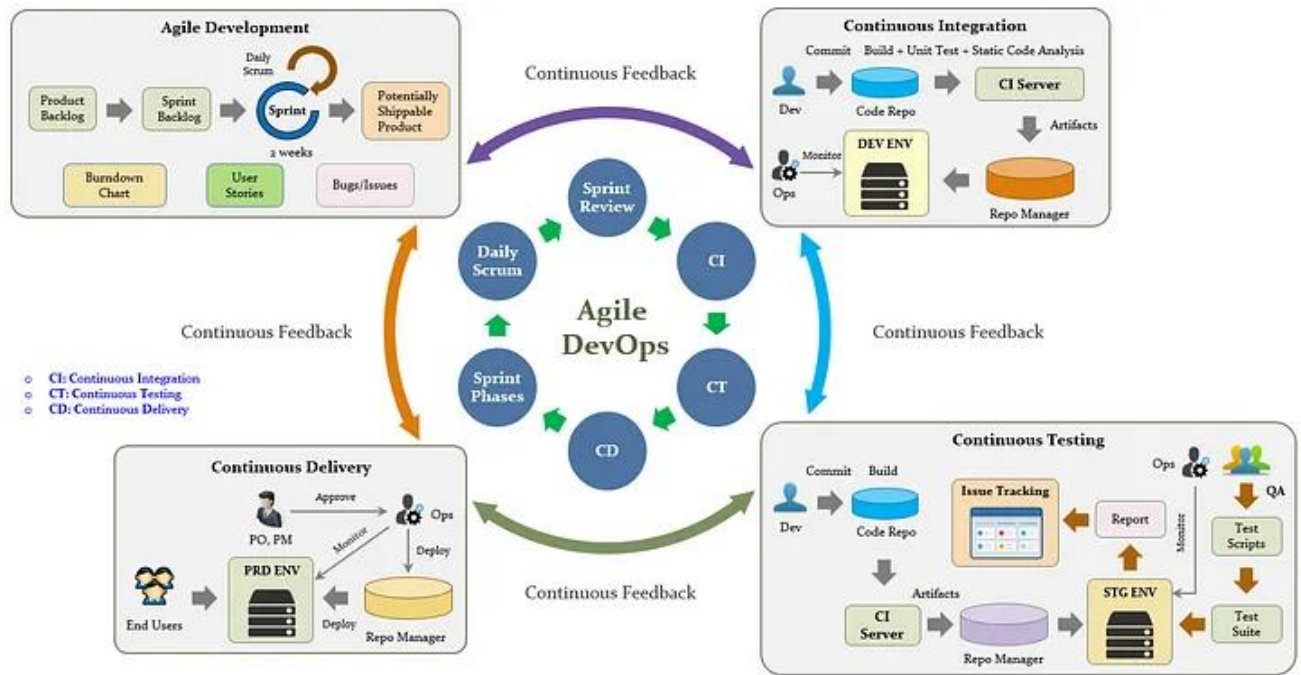


Figure 10 - Agile DevOps lifecycle (52)

3.3. Overview of System

3.3.1 Description of System's Operation

The system will consist of three parts, a frontend UI powered by Flutter SDK, a backend powered by the serverless cloud service Firebase, and a REST API to run and serve our AI processing program. The backend can be further broken down into several microservices to provide functionality for user authentication, user management, user data storage, natural language processing, payment processing and the secure call/chat system.

The frontend will be a modern and well-designed UI incorporating accessibility and user-friendly design. On initial startup of the app on mobile will display a short, animated splash screen while any necessary classes are generated following with the user being presented with a login page. While on desktop the idea is to have a landing page with information about what MyTherapyPal is and links to login to or register an account.

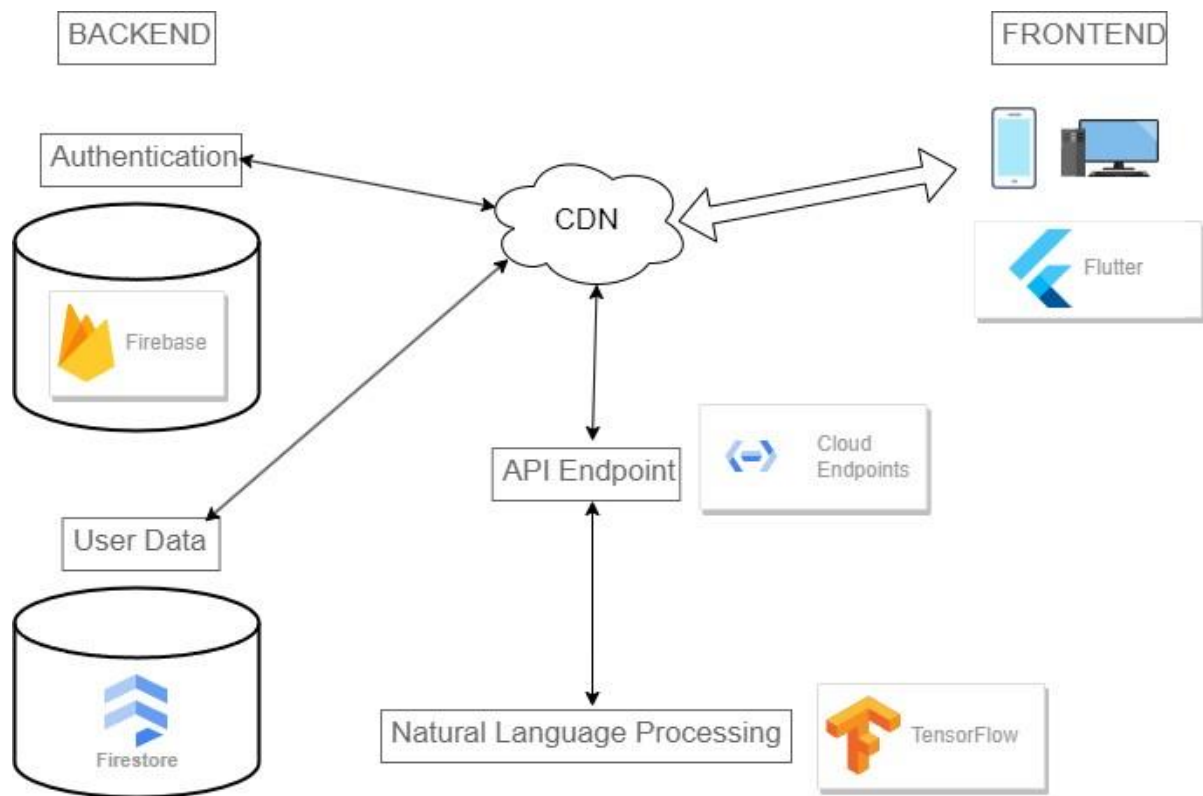


Figure 11 - High Level Architecture Map

If we come back to the mobile user experience where the user has landed on the login page, they will have the option to login or if they do not have an account, they can click the register new account button and it will take them to the registration screen. Once an account has been created the user will need to confirm their email address before being able to login. Once a user has logged in on mobile, they will be presented with a dashboard, and depending on the type of user/account (patient, therapist, admin) will be presented with different options.

The patient user type will be presented with a menu containing options to create notes in their calendar, submit the mood they are feeling that day, view assignments (such as from their therapist), view therapist listings and map, links to resources for better mental health management and account management settings. The therapist account will be presented with the options for note taking, client management (including appointment scheduling, messaging, and voice/video calling, adding and removal of clients, assign tasks to clients, review client notes, etc.), publish a listing and account management settings.

Secure messaging, voice and video calling will be handled by Agora SDK, which is an SDK with features for implementing secure text chat, voice calling, video calling, live streaming, and broadcast streaming. The SDK includes a secure authentication feature using tokens and uses built-in AES encryption to ensure strong end-to-end security.

3.3.2 System Requirements

The system requirements for the project are as follows:

- The system should be able to run on multiple platforms, such as Android, iOS, web, Windows, and macOS, without compromising the user experience, performance, or functionality. The system should have a consistent look and feel across all platforms and should be able to access the native features and APIs of each platform. The system should use Flutter as the frontend UI framework and Dart as the programming language.
- A daily journal/note taking feature where the user can write a note about their day, how they are feeling on a particular day, and how they are managing their mental health.
- A daily mood tracking feature where the user can select an icon and an emotion to log their mood at a specific time.
- A user account feature where the user can register an account and login to a dashboard which will present the user with a menu system to select the different features and functionalities. The system should use Firebase Authentication SDK to handle the creation of user accounts, and storage of basic information required for login and authentication. The system should also use Firebase Cloud Firestore NoSQL database to store other user data securely.
- A therapist listing feature where the user can find a therapist in their locality, view their profile, and contact them through the system. The system should also include geospatial information and maps to visualize the available therapists in the user's area.
- A messaging, voice and video calling feature where the user can communicate with their therapist securely and confidentially. The system should use Agora, a third-party microservice that provides real-time communication APIs for text, audio, and video.
- A payment feature where the user can pay their therapist for virtual or in-person therapy sessions, and where therapists can withdraw the funds in their account.
- A text summarization feature that uses natural language processing (NLP) to provide a summary of the user's notes and mood logs over a period of time. The system should also use machine learning to perform sentiment analysis and reveal deeper insights into the user's state of mind and moods throughout the time period.
- Therapist account tools to add/remove clients, assign tasks, post a listing, take notes, review client note summaries.
- An AI chatbot that specializes in cognitive-behavioural therapy (CBT) that uses artificial intelligence to provide CBT to users who have mental health issues such as anxiety, depression, or stress.

3.4. Design Documents

3.4.1 Class Diagram

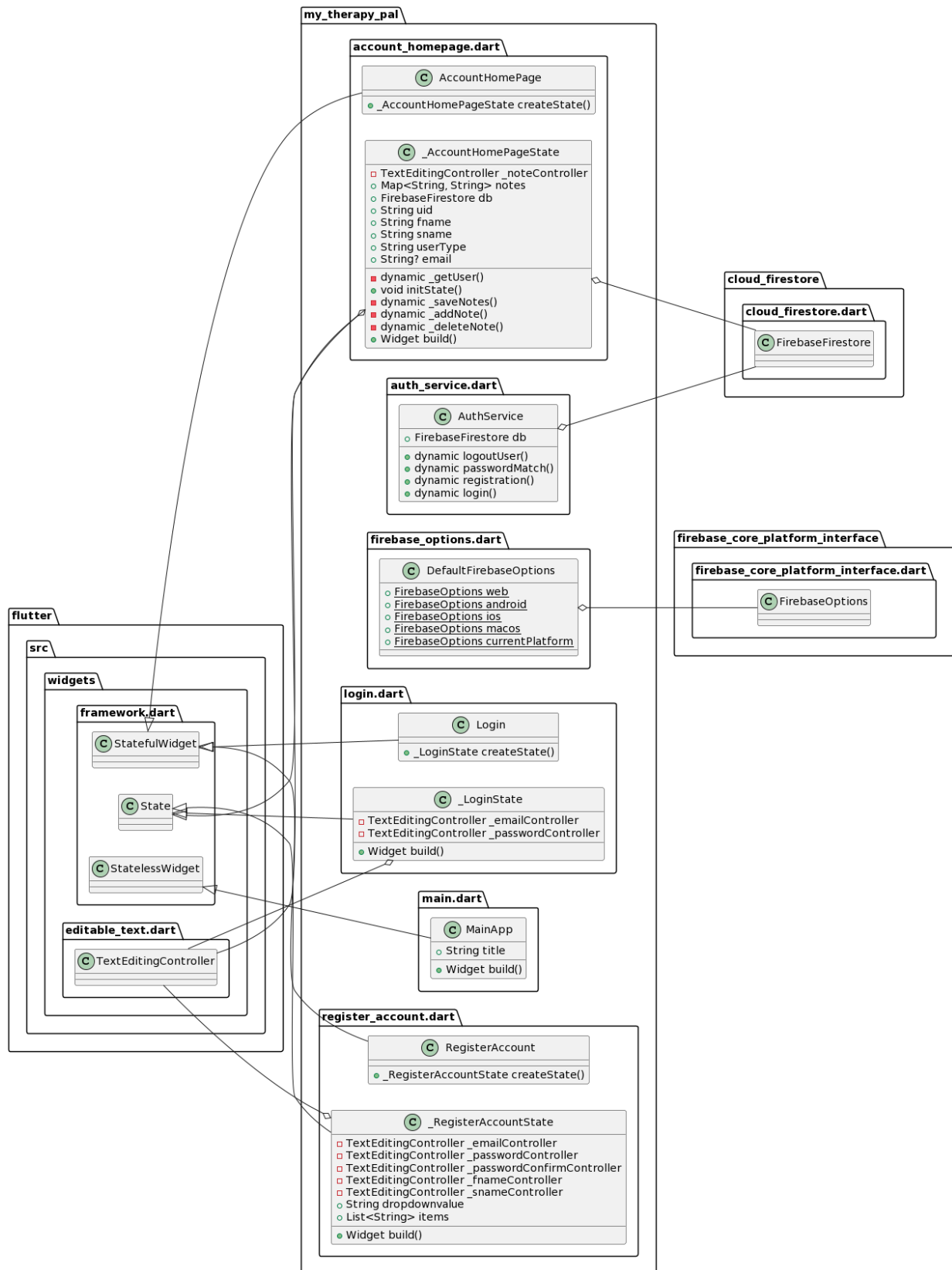


Figure 12 - Class Diagram

3.4.2 Firestore Schema Diagram

doc			
▼ __collections__	req	object	
• ▼ notes		object	
• • ▼ note_id		object	
• • • note		string	
• • • uuid		string	
• • • ▼ timestamp		object	
• • • • __datatype__		string	
• • • • ▼ value		object	
• • • • • _seconds		integer	
• • • • • _nanoseconds		integer	
• • • ▼ __collections__	req	object	
• ▼ profiles		object	
• • ▼ uuid		object	
• • • fname		string	
• • • sname		string	
• • • userType		string	
• • • ▼ __collections__	req	object	

Figure 13 – Cloud Firestore Database Schema Diagram

3.4.3 Use Case Diagram

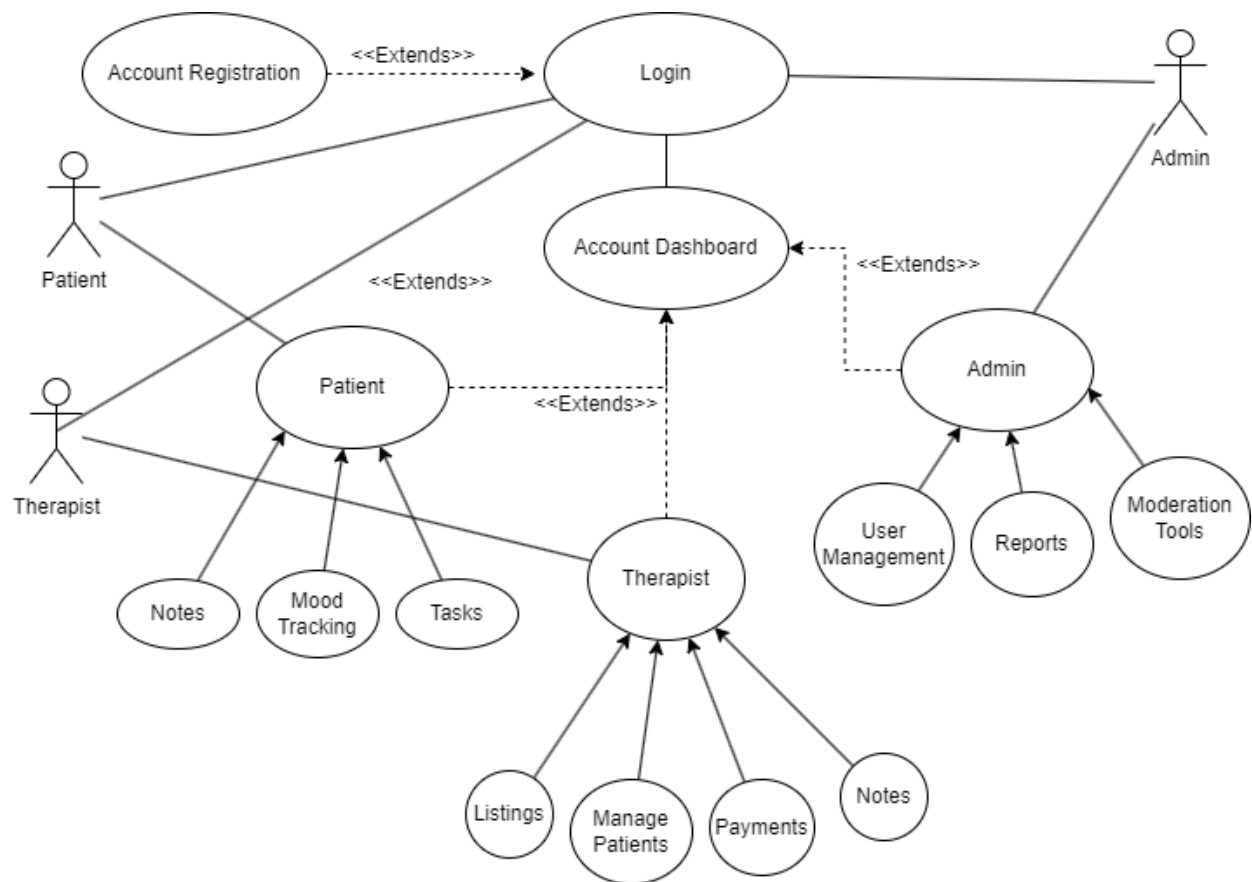


Figure 14 - Use Case Diagram

3.4.4 Sequence Diagram

The sequence diagram presented in figure 19 depicts the user flow of the current prototype where the user will first register an account, then will proceed to login, and finally create a note on their dashboard, which is automatically passed to the emotion detection API to return an emotion sentiment.

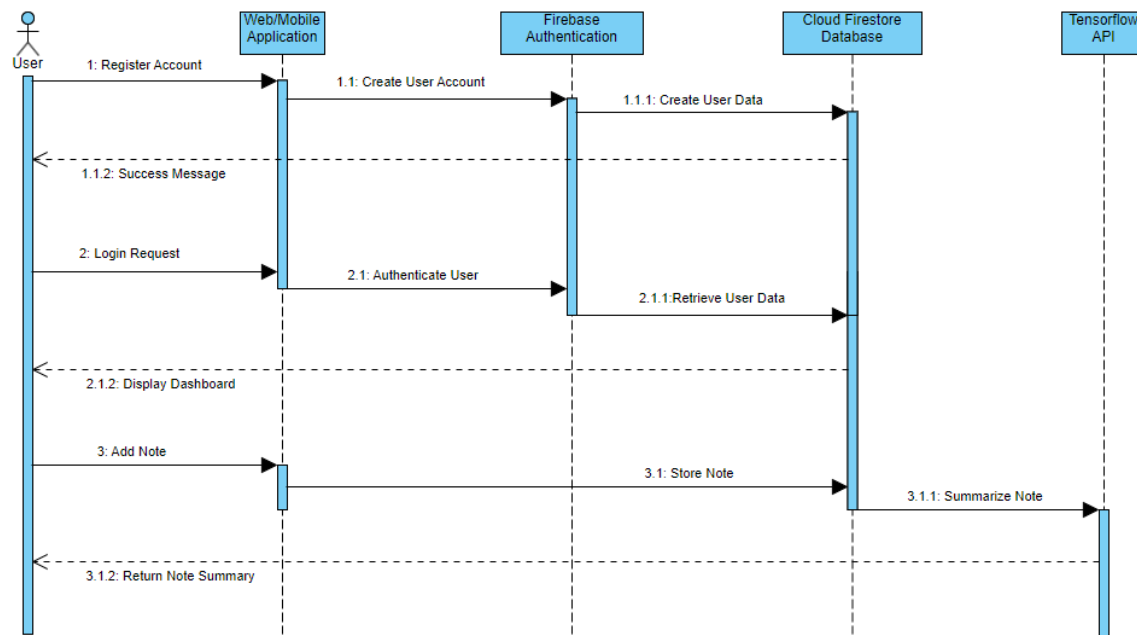


Figure 15 - Sequence Diagram

3.5. Conclusions

In conclusion, we have discussed the overview of the system design and the relevant development methodology research for this project. We defined and reviewed the software development methodologies, design patterns and selections that will be used to guide the development process. We also illustrated the system architecture and the interactions between the components using various diagrams and use cases. The next chapter will present the testing and evaluation plans for the system.

4. Testing and Evaluation

4.1. Introduction

Testing and evaluation play a crucial role in the software development process, in which steps are taken to ensure that the application will function as intended and to iron out as many issues as possible in order to produce a polished and refined software system. In some software development methodologies, testing is integrated as part of the development process in a continuous manner, such as in DevOps methodology of continuous integration and continuous delivery (CI/CD).

Testing can be extremely time consuming when performed manually or ad-hoc, which may be ok for smaller projects but for larger systems and for scalability, automated testing is crucial to the success of the system. There are various forms of testing strategies such as unit testing, in which individual components of a software system are tested in isolation with automated tests, but this is just one of many options when planning for a testing strategy and we will discuss in more depth the different testing options in the following sections.

4.2. Plan for Testing

There is an abundance of options that need to be examined when it comes to testing strategies in software development, with some that are more suitable for our application than others, and certain aspects of testing need strong consideration to ensure good performance and solid security measures. Various strategies were explored including unit testing, integration testing, functional testing, regression testing, user acceptance testing (UAT), security testing, and usability testing.

With flutter there are 3 categories for which automated testing falls, unit tests, widget tests (i.e. component test) and integration tests. According to flutter documentation, a well-tested application will have many unit and widget tests which are then tracked by code coverage (a software engineering concept which is defined as a percentage measure of the degree to which the source code of a program is executed when a particular set of tests are executed), along with enough integration tests to cover all important use cases.(54)

Unit testing in Flutter is implemented by adding the test package from Dart as a dependency, creating a test file for each dart file to be tested, for example app.dart would have app_test.dart, then create a test class within the new test file and write a test for the class or method unit we would like to check. Tests can then be run using the “flutter test” command in Visual Studio Code or relevant IDE. These tests can then be automated using a testing pipeline that will be triggered any time new code is merged with the development branch. This pipeline could test and then automatically build and publish the application if all tests pass.

Testing the application will involve a combination of the testing features provided by Flutter SDK for unit & integration testing combined with automation pipelines, along with security testing and usability testing.

4.3. Plan for Evaluation

4.3.1 Nielsens Heuristic Evaluation

Based on Jakob Nielsen's usability heuristics, it involves expert evaluators assessing the user interface against a set of 10 predefined usability principles which are:

1. Visibility of system status – Keeping users informed about what is happening.
2. Match between system and the real world – Matching the user's language based on region, follow real-world conventions.
3. User control and freedom – Provide easy ways to undo actions and escape from errors.
4. Consistency and standards – Follow platform conventions, maintain uniformity.
5. Error prevention – Anticipate user errors and prevent them where possible.
6. Recognition rather than recall – Minimize users memory load, make actions and options visible.
7. Flexibility and efficiency of use – Cater to both novice and expert users.
8. Aesthetic and minimalist design – Present only essential information, no extraneous details.
9. Help users recognize, diagnose, and recover from errors – Offer clear, concise error messages and guidance.
10. Help and documentation – Provide relevant, accessible information to assist users.

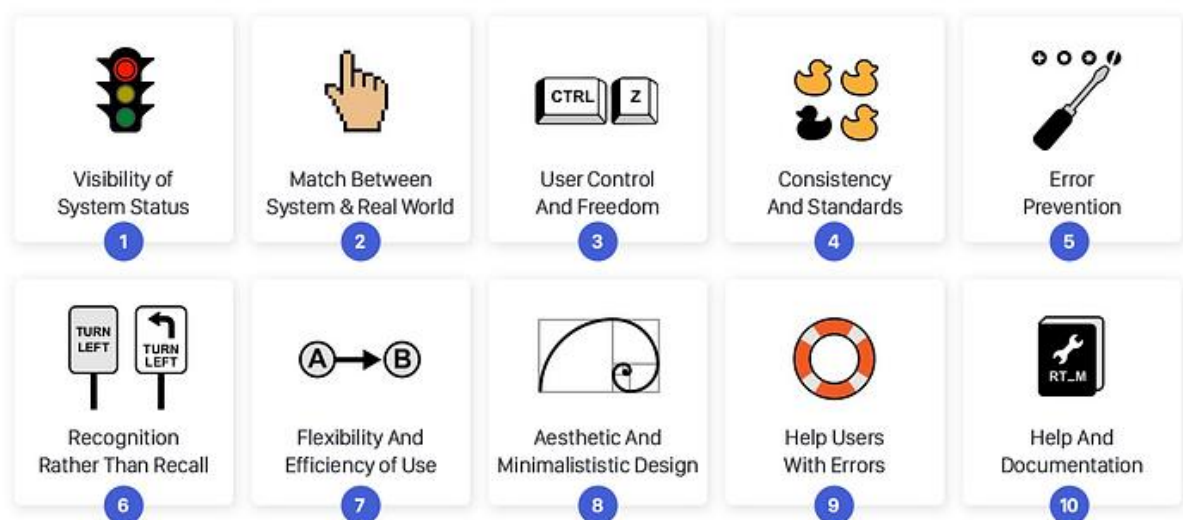


Figure 16 - Nielsens 10 Heuristics (55)

4.3.3 User Surveys and Feedback

There are many options available when it comes to evaluating the application through user surveys and general feedback. One way that user feedback can be collected would be in-app surveys in which short questionnaires are presented to users for collecting feedback and would provide lead to

large amounts of specific and targeted feedback helping to identify ways to improve the user experience.

Another way would be to use targeted surveys in which actionable insights can be obtained by targeting specific user segments and focusing on particular aspects of the user experience. A survey will be conducted during the later stages of this project in which an early version of the application will be distributed to professional and volunteer users and will be asked to complete a short survey provided through Google forms which will be used to evaluate the user experience, provide insight from domain professionals, and identify issues that may be overlooked during the development stages.

4.3.5 Compliance Testing

The compliance testing process for the evaluation of MyTherapyPal will be an extremely important one as the application will need to be GDPR compliant and must comply with security standards in relation to patient record confidentiality. In the process of conducting compliance testing for MyTherapyPal, an understanding of the General Data Protection Regulation (GDPR) will be acquired first, as it is the governing body for the processing of personal data within the EU. An audit will then be carried out to identify and categorize the data held by MyTherapyPal, which includes therapy session notes, invoices, and appointments. The legal basis for recording and storing this information is likely to be “legitimate interests”, which aligns with what clients would reasonably expect.

Measures such as using strong encryption and following cyber security best practices will be taken to ensure data protection by securely storing patient records and guarding against accidental disclosures. Penetration tests will be conducted to ensure all necessary security controls are implemented. Confidentiality will be maintained by ensuring that patient records are securely stored and not accessible to unauthorized individuals. Lastly, patients will be informed about the kind of information being held about them, how and why it might be shared, and with whom. It’s important to note that these are guidelines and consultation with a legal expert or data protection officer is recommended to ensure full compliance with GDPR and other relevant regulations.

4.3.6 Scalability Testing

Scalability testing is a non-functional testing methodology that verifies if an application can scale up or scale down in relation to specific performance measuring attributes such as the number of users of the system or the amount compute resources required to handle requests(56). It measures the performance of a system or network when the number of user requests are scaled up or down. The purpose of scalability testing is to ensure that the system can handle a projected increase in user traffic, data volume, transaction counts frequency, etc. It tests the system’s ability to meet growing needs. It is also referred to as performance testing, as it is focused on the behaviour of the application when deployed to a larger system or tested under excess load. In software engineering, scalability testing is used to measure at what point the application stops scaling and identify the reason behind it.(57)

4.3.7 Reliability Testing

Reliability testing is a popular evaluation technique in software development that assesses the ability of a software system to perform consistently and reliably under specific conditions for an extended period. It is a field of software testing that relates to testing a software's ability to function, given environmental conditions, for a particular amount of time.(58)

4.3.8 Accessibility Testing

Accessibility testing is a type of software testing that evaluates how user-friendly and functional a digital product is for people with disabilities. It tests whether people with various impairments, such as visual, auditory, motor, or cognitive, can use and access the product with ease. This testing is essential in ensuring that an application is inclusive and provides equal opportunities for all users(59).

The process of accessibility testing involves understanding accessibility guidelines, defining the scope of testing, and then assessing the inclusivity of an application or feature. The main objective is to identify and eliminate any barriers that might hinder people with disabilities from using the application or feature effectively. For instance, testing a mobile application would involve verifying if the app can be navigated and used by people with visual impairments by using voice-over technology for screen readers and providing sufficient colour contrast for those with colour blindness (60).

Accessibility testing is important because it helps to ensure that software applications are inclusive and usable by everyone, regardless of abilities. This is important for several reasons: such as legal compliance, social responsibility, benefits to businesses, and a design with users at the centre. In conclusion, accessibility testing has a key role in ensuring that a software application or system is accessible to each user, regardless of their abilities(60).

4.4. Conclusions

Testing and evaluation are crucial in the software development process, ensuring the application functions as intended and addressing issues to produce a polished system. Automated testing is essential for larger systems and scalability, and various testing strategies are explored, including unit testing, integration testing, functional testing, regression testing, user acceptance testing (UAT), security testing, and usability testing. A well-tested application will have many unit and widget tests, tracked by code coverage, along with enough integration tests to cover all important use cases.

User surveys and feedback are essential tools for evaluating applications, such as in-app surveys or targeted surveys. In-app surveys provide specific and targeted feedback, while targeted surveys focus on specific user segments. Feature analysis examines each feature's functionality, usability, and impact on the overall user experience, providing valuable insights for improvement. Compliance testing is crucial for MyTherapyPal, as it must be GDPR compliant and comply with security standards related to patient record confidentiality. An audit will be conducted to identify and categorize data held by the application, ensuring data protection through secure storage and

guarding against accidental disclosures. Penetration tests will be conducted to ensure security controls are implemented, and patients are informed about the information being held about them.

5. Prototype Development

5.1. Introduction

A prototype application was developed in Flutter with some basic functionality such as login/logout with user authentication, registration with full account creation in Firebase and a primitive account dashboard with a basic note taking app which stores a user's notes in Firebase and have functionality to add & delete notes. The centre piece of this proof-of-concept prototype is the BERT model that has been trained for emotion detection using sentiment analysis on a dataset of tweets and emotional context, a python flask RESTful API was developed that takes text as input, processes it through the trained model and returns an emotion detected by the model based on the content of the text. The proof-of-concept was selected in an early stage of this project, but new requirements of higher complexity were realised at a very late point in the interim stage which would have superseded the current proof-of-concept, namely, the AI CBT chatbot.

5.2. Prototype Development

5.2.1 Frontend Application

The frontend prototype application was developed using Flutter in Dart programming language and consists of a login screen, login/logout functionality, account registration screen, account creation functionality and account dashboard screen with note taking system.

5.2.1.1 Login

The login class is where the login screen and login functionality are handled. Widgets are used to display the login screen, create text inputs for username and password, along with a submit button and a link to register a new account.

```

class Login extends StatefulWidget {
  const Login({Key? key}) : super(key: key);

  @override
  _LoginState createState() => _LoginState();
}

class _LoginState extends State<Login> {
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          const MainApp().title,
          style: const TextStyle(color: Colors.white),
        ), // Text
      ), // AppBar
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.start,
          children: [
            SizedBox(
              height: MediaQuery.of(context).size.height / 10,
            ), // SizedBox
            const Text(
              'Login',
              style: TextStyle(color: Colors.black, fontSize: 28),
            ), // Text
            SizedBox(
              height: MediaQuery.of(context).size.height / 20,
            ), // SizedBox
            SizedBox(
              width: MediaQuery.of(context).size.width / 2,
              child: TextField(
                controller: _emailController,
                decoration: const InputDecoration(hintText: 'Email'),
              ), // TextField
            ), // SizedBox
            const SizedBox(
              height: 30.0,
            ), // SizedBox
            SizedBox(
              width: MediaQuery.of(context).size.width / 2,
              child: TextField(
                controller: _passwordController,
                obscureText: true,
                decoration: const InputDecoration(
                  hintText: 'Password',
                ), // InputDecoration
              ), // TextField
            ), // SizedBox
            const SizedBox(
              height: 30.0,
            ), // SizedBox
            ElevatedButton(
              onPressed: () async {
                final message = await AuthService().login(
                  email: _emailController.text,
                  password: _passwordController.text,
                );
                if (message!.contains('Success')) {
                  Navigator.of(context).pushReplacement(
                    MaterialPageRoute(
                      builder: (context) => const AccountHomePage(), // MaterialPageRoute
                    );
                )
                ScaffoldMessenger.of(context).showSnackBar(
                  SnackBar(
                    content: Text(message),
                  ), // SnackBar
                );
              },
              child: const Text('Login', style: TextStyle(color: Colors.white)),
            ), // ElevatedButton
            const SizedBox(
              height: 30.0,
            ), // SizedBox
            TextButton(
              onPressed: () {
                Navigator.of(context).push(
                  MaterialPageRoute(
                    builder: (context) => const RegisterAccount(),
                  ), // MaterialPageRoute
                );
              },
              child: const Text('Create New Account'),
            ), // TextButton
          ],
        ), // Column
      ), // Center
    ); // Scaffold
  }

```

Figure 17 - Login Class Code Snippet

5.2.1.2 Authentication Service

The authentication service class will contain the methods necessary for handling user authentication and interactions with Firebase authentication when registering new accounts or logging users in/out.

```
auth_service.dart > AuthService > registration
import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart';

class AuthService {

  var db = FirebaseFirestore.instance;

  // Logout a user in firebase
  logoutUser() async {
    await FirebaseAuth.instance.signOut();
  }

  // Password match check
  passwordMatch(pwd1, pwd2) {
    if (pwd1 == pwd2) {
      return true;
    } else {
      return false;
    }
  }

  Future<String?> registration({
    required String email,
    required String password,
    required String passwordConfirm,
    required String fname,
    required String sname,
    required String userType,
  }) async {
    if (!passwordMatch(password, passwordConfirm)) {
      return 'Passwords do not match';
    } else if (password.length < 6) {
      return 'Password must be at least 6 characters';
    }
  }
}
```

Figure 18 - Authentication Service Code Snippet

5.2.1.3 Registration

The registration class will provide a widget to display the registration screen, along with the form fields for collecting the necessary information to create a new user account.

```
class RegisterAccount extends StatefulWidget {
  const RegisterAccount({Key? key}) : super(key: key);

  @override
  RegisterAccountState createState() => _RegisterAccountState();
}

class _RegisterAccountState extends State<RegisterAccount> {
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final TextEditingController _passwordConfirmController = TextEditingController();
  final TextEditingController _fnameController = TextEditingController();
  final TextEditingController _snameController = TextEditingController();

  // Initial Selected Value
  String dropdownvalue = '';

  // List of items in our dropdown menu
  var items = [
    '',
    'Patient',
    'Therapist',
    'Admin',
  ];
};
```

Figure 19 - RegisterAccount Class Code Snippet

5.2.1.4 Account Homepage

The account homepage class will provide a widget to display the account dashboard screen, which contains the note taking system. Notes are displayed in a ListView with a subtitle to display the result of the emotional sentiment analysis.

```
55  _saveNotes(newNote) async {
56    try {
57      final myNewDoc = await db.collection("notes").add({
58        "uuid": uid,
59        "note": newNote,
60        "timestamp": Timestamp.now()
61      });
62      return myNewDoc.id.toString();
63    } catch (e) {
64      print(e);
65      return null;
66    }
67  }
68
69  _addNote() async {
70    String newNote = _noteController.text;
71    if (newNote.isNotEmpty) {
72      final newNoteId = await _saveNotes(newNote);
73      if (newNoteId != null) {
74        setState(() {
75          notes[newNoteId] = newNote;
76        });
77        _noteController.clear();
78      }
79    }
80  }
81
82  deleteNote(String noteId) {
83    db.collection("notes").where("uuid", isEqualTo: uid).get().then(
84      (querySnapshot) {
85        for (var docSnapshot in querySnapshot.docs) {
86          if (docSnapshot.id == noteId) {
87            docSnapshot.reference.delete();
88          }
89        }
90      },
91      onError: (e) => print("Error completing: $e"),
92    );
93    setState(() {
94      notes.remove(noteId);
95    });
96  }
```

Figure 20 - Account Homepage Code Snippet

5.2.2 Backend Services

5.2.3.1 Firebase Authentication

Firebase authentication is an SDK used to handle user account creation and login authentication. In this prototype, users are authenticated with email and password but will implement other sign-in methods in the final application such as with Google or Facebook. Firebase authentication securely stores the basic information required to handle and authenticate users such as email, password, phone, display name, unique user id, account creation timestamp, last login timestamp, and a profile picture.

5.2.3.2 Cloud Firestore

Cloud Firestore is our NoSQL data storage for all other user data, including note data and extra profile information. Note data is stored in a collection labelled “notes” and stores note text, timestamp and unique user id. Extra profile information is stored in a collection labelled “profiles” and stores first name, last name, and the users account type (patient, therapist, administrator).

5.2.3.3 AI Emotional Sentiment API

The emotional sentiment feature will be provided through an API which is being implemented through the use of Python Flask, which is a web framework that allows us to create and run lightweight web applications, such as a basic REST API. For the prototype, this REST API is provided over localhost at port 5000. This small python application will contain the methods for training and fine-tuning the BERT model on our tweet sentiment dataset, then taking text as input through POST calls to the API, running the text through our trained model and returning a predicted emotion.

```
emotional_sentiment_analysis > app.py > ...
1  from flask import Flask, request, jsonify
2  from flask_cors import CORS # Import the CORS module
3  from predict import predict_emotion
4
5  app = Flask(__name__)
6  CORS(app) # Enable CORS for all routes
7
8  @app.route('/predict', methods=['POST'])
9  def predict():
10     data = request.get_json()
11     input_text = data['text']
12     predicted_emotion = predict_emotion(input_text)
13     return jsonify({"predicted_emotion": predicted_emotion})
14
15 if __name__ == '__main__':
16     app.run(port=5000)
17
18
```

Figure 21 - Emotional Sentiment API Code Snippet

```

emotional_sentiment_analysis > predict.py > ...
8
9 # Check if the saved model exists
10 if not os.path.exists("emotion_model"):
11     # Run the train script to train the model
12     train_model()
13     print("Model trained successfully.")
14
15 # Load the trained model
16 model = tf.keras.models.load_model("emotion_model")
17
18 # Load tokenizer
19 model_ckpt = "distilbert-base-uncased"
20 tokenizer = AutoTokenizer.from_pretrained(model_ckpt)
21
22 # Load label encoder
23 label_encoder = joblib.load('label_encoder.joblib')
24
25 def predict_emotion(input_text):
26     tokens = tokenizer(input_text, padding=True, truncation=True, return_tensors="tf")
27     inputs = {
28         'input_ids': tokens['input_ids'],
29         'attention_mask': tokens['attention_mask']
30     }
31
32     # Make prediction
33     predictions = model.predict(inputs)
34
35     # Extract logits from the predictions dictionary
36     logits = predictions['logits']
37
38     # Determine the predicted label
39     predicted_label = int(tf.argmax(logits, axis=-1)[0])
40
41     # Convert the predicted label to emotion
42     predicted_emotion = label_encoder.inverse_transform([predicted_label])[0]
43
44     return predicted_emotion
45
46 # Create an argument parser
47 parser = argparse.ArgumentParser()
48 parser.add_argument("--text", type=str, help="Input text for emotion prediction")
49
50 # Parse the command line arguments
51 args = parser.parse_args()
52
53 # Get the input text from the command line
54 input_text = args.text if args.text else "No input text provided"
55

```

Figure 22 - Emotional Sentiment Prediction Code Snippet

5.3. Prototype Screenshots

5.3.1 Login Screen

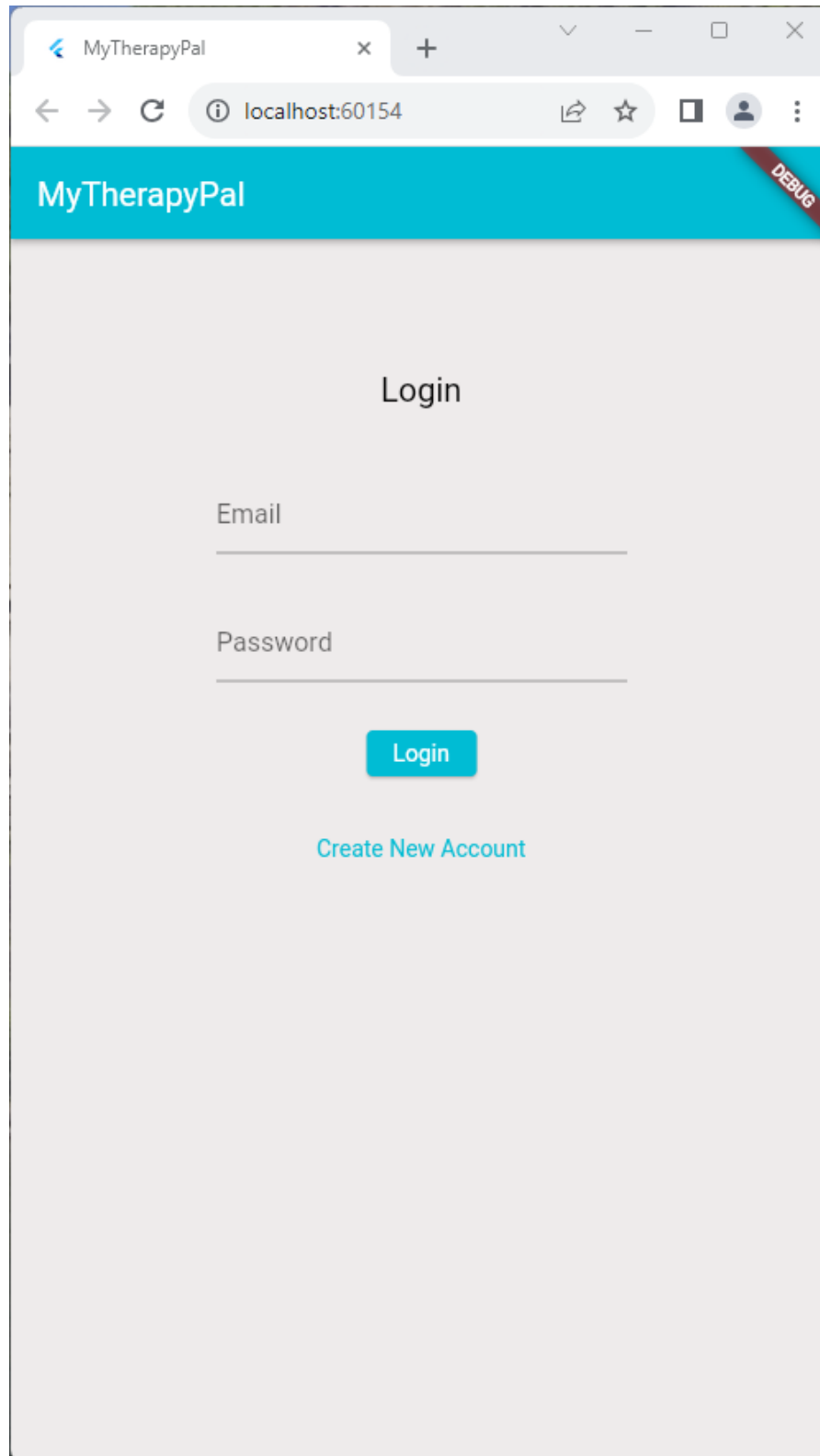


Figure 23 - Login Page Screenshot

5.3.2 Registration Screen

MyTherapyPal

localhost:60154

← MyTherapyPal

DEBUG

Sign Up

Email

Password

Confirm Password

First Name

Last Name

Please select your user type:

Patient

Create Account

Figure 24 - Registration Page Screenshot

5.3.3 Account Dashboard Screen

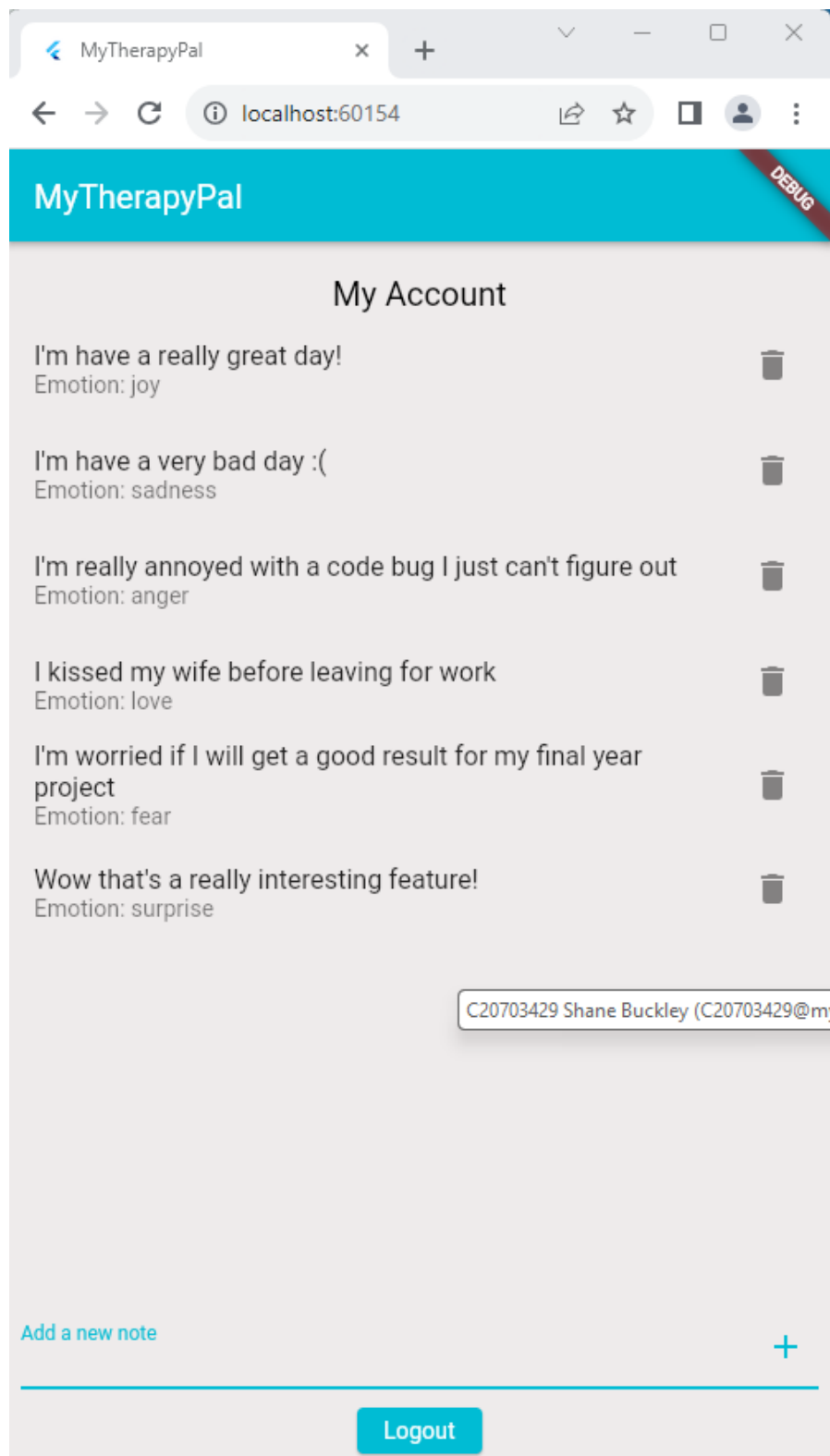


Figure 25 - Account Dashboard Screenshot

5.4. Conclusions

In conclusion, a prototype was successfully implemented and demonstrated the proof-of-concept of AI emotional sentiment analysis of a text input within a basic user interface. The prototype showcased the potential for accurately analysing and interpreting emotions in text, opening up the possibilities for the future of this project. This successful implementation lays the foundation for further advancements and refinement of AI-based functionality going forward.

6. Issues and Future Work

6.1. Introduction

This section aims to review and discuss the issues faced over the course of the interim stage of the project, and the ongoing risks and to outline a roadmap for future development during the final stage of the project.

6.2. Issues and Risks

6.2.1 Confidentiality

Maintaining strict confidentiality when controlling, processing and storing private patient medical information is one of the most important issues in a project of this nature and careful consideration needs to be given to the security of this data through every step of the development process.

6.2.2 Natural Language Processing

There is an ongoing risk in relation to natural language processing, machine learning, and AI in the future of this project due to the complexity of the technology, which may rise far beyond the scope of this project. Issues may also arise with time constraints in relation to this complexity risk. Every step will be taken to ensure that the final solution developed remains within the scope of the project and that continuous consideration will be made to mitigate this ongoing risk wherever possible.

Additionally, as the technology evolves, new challenges and limitations may emerge that were not initially anticipated. Therefore, it is crucial to regularly reassess and adapt the project's approach to address any potential risks that may arise from these advancements. Constant monitoring of the rapid advancements in the field of AI will be essential to stay updated on the latest developments and mitigate any unforeseen challenges.

6.2.3 Time Constraints

One of the main risks and issues in relation to time constraints while developing an application for a final year project is the possibility of scope creep, which is the uncontrolled expansion of the project's requirements and features over time. Scope creep can result from unclear or changing expectations, lack of communication, poor planning, or unrealistic deadlines.

To avoid or mitigate scope creep, it is important to define the project's scope clearly, prioritize the essential features, communicate frequently with the project supervisor and the end users, and monitor the progress and feedback regularly.

6.3. Plans and Future Work

6.3.1 System Development

Going forward, the project will progress in accordance with the timeline defined in Figure 27, which shows the estimated duration and deadlines for each stage of the project. The next stage of the project will be to plan the final design of the app, which will involve creating wireframes, mock-ups, and prototypes to illustrate the user interface and the functionality of the app. The final design will be based on the user requirements, the feedback from the supervisor and the potential users, and the best practices for cross-platform application development.

After the final design is approved, the implementation stage will begin, which will consist of coding, testing, and debugging the app using the selected tools and technologies. The implementation stage will follow an agile methodology, which will allow for iterative development and continuous improvement of the app.

6.3.2 CBT Chatbot Research

The development of an AI CBT chatbot that can provide cognitive behavioural therapy to users who suffer from mental health issues will be continued in the next stage of this project. A suitable large language model (LLM) that can be trained on CBT, and generate natural and empathetic responses for the chatbot is one of the main challenges of this project. Language models that can be hosted securely, such as LLaMA 2 or Vicuna will be explored and compared in terms of their performance and feasibility.

6.3.1. GANTT Chart

Interim Submission

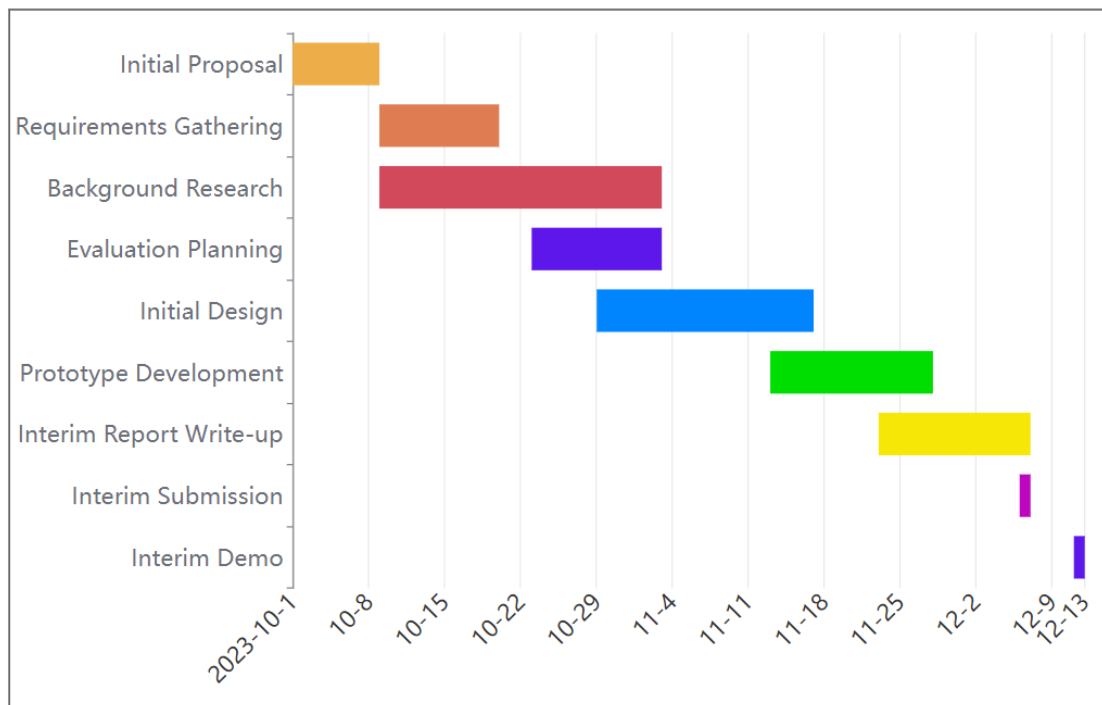


Figure 26 - Interim GANTT Chart

Final Submission

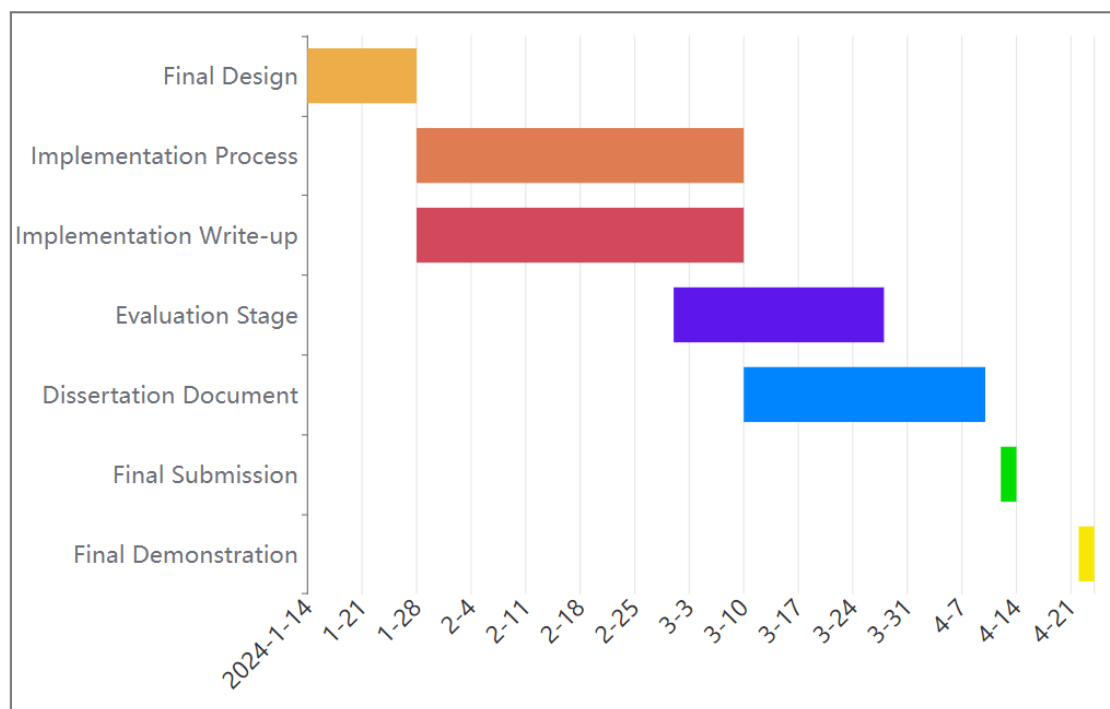


Figure 27 - Final GANTT Chart

Bibliography

1. Chandrashekar P. Do mental health mobile apps work: evidence and recommendations for designing high-efficacy mental health mobile apps. *mHealth*. 2018 Mar 23;4:6.
2. Quenza - Apps on Google Play [Internet]. [cited 2023 Nov 28]. Available from: <https://play.google.com/store/apps/details?id=com.quenza.app&hl=en>
3. Quenza™ - Craft Your Care - #1 Online Tool For Coaches [Internet]. [cited 2023 Dec 3]. Available from: <https://quenza.com/>
4. About - Daylio Knowledge Base [Internet]. [cited 2023 Nov 28]. Available from: <https://faq.daylio.net/category/23-about>
5. AHEAD [Internet]. AHEAD; [cited 2023 Dec 3]. Daylio - Wellbeing App. Available from: <https://ahead.ie/daylio>
6. App Store [Internet]. 2021 [cited 2023 Nov 28]. Moodnotes - Mood Tracker. Available from: <https://apps.apple.com/us/app/moodnotes-mood-tracker/id1019230398>
7. One Mind PsyberGuide [Internet]. [cited 2023 Dec 3]. Moodnotes - CBT & Mood Tracker. Available from: <https://onemindpsyberguide.org/apps/moodnotes-cbt-mood-tracker/>
8. App Store [Internet]. 2023 [cited 2023 Nov 28]. eMoods Bipolar Mood Tracker. Available from: <https://apps.apple.com/us/app/emoods-bipolar-mood-tracker/id1184456130>
9. One Mind PsyberGuide [Internet]. [cited 2023 Dec 3]. eMoods Bipolar Mood Tracker. Available from: <https://onemindpsyberguide.org/apps/emoods-bipolar-mood-tracker/>
10. Theratrak – Apps on Google Play [Internet]. [cited 2023 Nov 28]. Available from: https://play.google.com/store/apps/details?id=co.theratrak.Theratrak&hl=en_IE
11. Theratrak [Internet]. [cited 2023 Dec 3]. How Theratrak Works for Therapists & Allied Health Professionals. Available from: <https://www.theratrak.co/for-therapists>
12. Android | Definition, History, & Facts | Britannica [Internet]. 2023 [cited 2023 Nov 30]. Available from: <https://www.britannica.com/technology/Android-operating-system>
13. IOS | Apple, Updates, Software, & Origin | Britannica [Internet]. 2023 [cited 2023 Nov 30]. Available from: <https://www.britannica.com/topic/iOS>
14. What is Windows? [Internet]. [cited 2023 Nov 30]. Available from: <https://www.computerhope.com/jargon/w/windows.htm>
15. Statista [Internet]. [cited 2023 Nov 30]. Desktop operating system market share 2013-2023. Available from: <https://www.statista.com/statistics/218089/global-market-share-of-windows-7/>
16. StatCounter Global Stats [Internet]. [cited 2023 Nov 30]. Desktop Operating System Market Share Worldwide. Available from: <https://gs.statcounter.com/os-market-share/desktop/worldwide>
17. www.javatpoint.com [Internet]. [cited 2023 Nov 30]. Firebase Data Organization in Firestore - Javatpoint. Available from: <https://www.javatpoint.com/firebase-data-organization-in-firestore>

18. React Native · Learn once, write anywhere [Internet]. [cited 2023 Nov 28]. Available from: <https://reactnative.dev/>
19. Microsoft [Internet]. [cited 2023 Nov 28]. Xamarin | Open-source mobile app platform for .NET. Available from: <https://dotnet.microsoft.com/en-us/apps/xamarin>
20. Flutter - Build apps for any screen [Internet]. [cited 2023 Nov 28]. Available from: <https://flutter.dev/>
21. Django Project [Internet]. [cited 2023 Nov 28]. Django. Available from: <https://www.djangoproject.com/>
22. Ruby on Rails [Internet]. [cited 2023 Nov 28]. Ruby on Rails. Available from: <https://rubyonrails.org/>
23. Firebase [Internet]. [cited 2023 Nov 28]. Firebase | Google's Mobile and Web App Development Platform. Available from: <https://firebase.google.com/>
24. Areàn PA, Hoa Ly K, Andersson G. Mobile technology for mental health assessment. *Dialogues Clin Neurosci*. 2016 Jun 30;18(2):163–9.
25. Anthes E. Mental health: There's an app for that. *Nature*. 2016 Apr 1;532(7597):20–3.
26. Technology and the Future of Mental Health Treatment - National Institute of Mental Health (NIMH) [Internet]. [cited 2023 Dec 4]. Available from: <https://www.nimh.nih.gov/health/topics/technology-and-the-future-of-mental-health-treatment>
27. <https://www.apa.org> [Internet]. [cited 2023 Dec 5]. What is Cognitive Behavioral Therapy? Available from: <https://www.apa.org/ptsd-guideline/patients-and-families/cognitive-behavioral>
28. nhs.uk [Internet]. 2021 [cited 2023 Dec 5]. How it works - Cognitive behavioural therapy (CBT). Available from: <https://www.nhs.uk/mental-health/talking-therapies-medicine-treatments/talking-therapies-and-counselling/cognitive-behavioural-therapy-cbt/how-it-works/>
29. Cleveland Clinic [Internet]. [cited 2023 Dec 5]. Cognitive Behavioral Therapy (CBT): What It Is & Techniques. Available from: <https://my.clevelandclinic.org/health/treatments/21208-cognitive-behavioral-therapy-cbt>
30. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
31. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention Is All You Need [Internet]. arXiv; 2023 [cited 2023 Dec 3]. Available from: <http://arxiv.org/abs/1706.03762>
32. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux MA, Lacroix T, et al. LLaMA: Open and Efficient Foundation Language Models [Internet]. arXiv; 2023 [cited 2023 Dec 3]. Available from: <http://arxiv.org/abs/2302.13971>
33. Introducing LLaMA: A foundational, 65-billion-parameter language model [Internet]. [cited 2023 Dec 3]. Available from: <https://ai.meta.com/blog/large-language-model-llama-meta-ai/>
34. LLaMA [Internet]. [cited 2023 Dec 3]. Available from: https://huggingface.co/docs/transformers/main/en/model_doc/llama

35. Introducing ChatGPT [Internet]. [cited 2023 Dec 4]. Available from: <https://openai.com/blog/chatgpt>
36. Caulfield J. Scribbr. 2023 [cited 2023 Dec 4]. What Is ChatGPT? | Everything You Need to Know. Available from: <https://www.scribbr.com/ai-tools/what-is-chatgpt/>
37. 11 Software Development Methodologies (Plus How To Pick One) | Indeed.com [Internet]. [cited 2023 Dec 3]. Available from: <https://www.indeed.com/career-advice/career-development/software-development-methodologies>
38. Easy Agile [Internet]. [cited 2023 Dec 3]. 8 Software Development Methodologies Explained. Available from: <https://easyagile.com/blog/software-development-methodologies/>
39. What Is a Software Development Methodology? (With 11 Types) | Indeed.com Australia [Internet]. [cited 2023 Dec 3]. Available from: <https://au.indeed.com/career-advice/career-development/software-development-methodology>
40. Best Software Development Methodologies: Pros & Cons In 2022 [Internet]. 2022 [cited 2023 Dec 3]. Available from: <https://devtechnosys.com/insights/software-development-methodologies/>
41. Top 4 Software Development Methodologies | Synopsys Blog [Internet]. [cited 2023 Dec 3]. Available from: <https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies.html>
42. Atlassian. Atlassian. [cited 2023 Dec 3]. What is Agile? Available from: <https://www.atlassian.com/agile>
43. Asana. Asana. [cited 2023 Dec 3]. What Is Agile Methodology? (A Beginner's Guide) [2023] • Asana. Available from: <https://asana.com/resources/agile-methodology>
44. Springboard Blog [Internet]. 2020 [cited 2023 Dec 3]. What is the Agile Methodology in Software Development? Available from: <https://www.springboard.com/blog/software-engineering/agile-methodology-software-development/>
45. Altvater A. Stackify. 2023 [cited 2023 Dec 3]. What is Agile Methodology? Tools, Best Practices & More. Available from: <https://stackify.com/agile-methodology/>
46. 33 Managing the Development of Large Software Systems (1970). In: Ideas That Created the Future: Classic Papers of Computer Science [Internet]. MIT Press; 2020 [cited 2023 Dec 3]. p. 321–32. Available from: <https://ieeexplore.ieee.org/document/9357688>
47. Atlassian. Atlassian. [cited 2023 Dec 3]. Agile vs. waterfall project management. Available from: <https://www.atlassian.com/agile/project-management/project-management-intro>
48. Jackson D. Software Development Processes.
49. Team AC. Waterfall Methodology: Project Management | Adobe Workfront [Internet]. [cited 2023 Dec 3]. Available from: <https://business.adobe.com/blog/basics/waterfall>
50. Docslib [Internet]. [cited 2023 Dec 3]. Waterfall vs V-MODEL Vs AGILE: a COMPARATIVE STUDY on SDLC. Available from: <https://docslib.org/doc/6044506/waterfall-vs-v-model-vs-agile-a-comparative-study-on-sdlc>

51. Atlassian. Atlassian. [cited 2023 Dec 4]. What is DevOps? Available from: <https://www.atlassian.com/devops>
52. Raycad. DevOps methodology and process [Internet]. Medium. 2019 [cited 2023 Dec 4]. Available from: <https://medium.com/@raycad.seedotech/devops-methodology-and-process-dde388eb65bd>
53. Feature Driven Development (FDD) [Internet]. [cited 2023 Dec 4]. Available from: <https://www.productplan.com/glossary/feature-driven-development/>
54. Testing Flutter apps [Internet]. [cited 2023 Dec 4]. Available from: <https://docs.flutter.dev/testing/overview>
55. Hazlazuardi. How My Team Uses Nielsen's 10 Usability Heuristics in Our Application [Internet]. Medium. 2021 [cited 2023 Dec 1]. Available from: <https://medium.com/@hazlazuardi/how-my-team-uses-nielsens-10-usability-heuristics-in-our-application-865fef19220d>
56. What is Scalability Testing? Learn with Example [Internet]. 2023 [cited 2023 Dec 5]. Available from: <https://www.guru99.com/scalability-testing.html>
57. Software Testing Help [Internet]. 2023 [cited 2023 Dec 5]. What is Scalability Testing? How to Test the Scalability of an Application. Available from: <https://www.softwaretestinghelp.com/what-is-scalability-testing/>
58. Software Testing | Reliability Testing [Internet]. GeeksforGeeks. 2019 [cited 2023 Dec 5]. Available from: <https://www.geeksforgeeks.org/software-testing-reliability-testing/>
59. BrowserStack [Internet]. [cited 2023 Dec 5]. Accessibility Testing: An Essential Guide. Available from: <https://browserstack.wpengine.com/guide/accessibility-testing/>
60. What is Accessibility Testing? (Examples) [Internet]. 2023 [cited 2023 Dec 5]. Available from: <https://www.guru99.com/accessibility-testing.html>