University of Essex

Department of Computer Sciences and Electronic Engineering

CE901 DISSERTATION

The Effectiveness of Exercise in Parkinsons Disease

Omkar R Kharkar

2205233

Supervisor: **Dr Katerina Bourazeri**

August 28, 2023
Colchester Campus, Essex

CHAPTER 1

Abstract

Parkinson's disease (PD) is a long-term degenerative disorder affecting the nervous system. It is characterized by motor symptoms such as tremors, rigidity, bradykinesia, and postural instability. This report examines current research on the effectiveness of exercise in improving symptoms and slowing the progression of PD in patients. Regular physical activity can significantly enhance motor function, gait, balance, flexibility, and quality of life. Aerobic exercise can help improve cardiovascular health and endurance, while strength training can enhance muscle strength and motor function. Balance and coordination exercises can reduce fall risk and improve mobility, and flexibility training can increase range of motion and reduce rigidity. The report also highlights evidence suggesting that exercise boosts dopamine levels, neuroplasticity, and neuroprotection, which counteract PD pathology. Key studies show that exercise should be an integral part of PD management, though further research is needed to determine the optimal parameters like frequency, intensity, and type of exercise. Additionally, the report describes the development of a mobile app that provides customized video exercise guides to assist PD patients. It includes details on design components, technical architecture, testing, and future enhancements. In conclusion, studies confirm that exercise is a promising intervention to manage symptoms and slow the decline of PD when tailored to individual needs.

Contents

List of Figures

List of Tables

CHAPTER 2

Introduction

2.1 Introduction

Parkinson's disease is a chronic and progressive neurodegenerative condition that affects a considerable number of people around the world. This ailment results in a decline in motor function and a plethora of incapacitating symptoms. Although manifestations can differ between individuals, initial signs often include subtle changes in movement, such as slight tremors, rigidity, postural instability or difficulty with fine motor skills. Activities like aerobic exercises, strength training, balance exercises, and stretching routines are the types of physical exercise that are most effective for Parkinson's disease sufferers. Physical exercise has been shown to facilitate improvements in gait, balance, flexibility, and motor coordination, providing a viable means of improving the overall quality of life of Parkinson's disease sufferers. Individualized exercise programs for Parkinson's disease can take into account factors such as age, health, stage of the disease, specific symptoms, and personal goals. For example, someone with Parkinson's disease may struggle with balance and gait disturbances, while another individual may experience stiffness and rigidity. These programs can be tailored to address the unique needs and abilities of individuals with Parkinson's disease, leading to improved motor function, balance, coordination, and flexibility. Healthcare professionals can assess an individual's abilities and develop tailored exercise plans.

The signs and symptoms of Parkinson's disease can vary from person to person, but common early symptoms include:

1. Tremor: A shaking or trembling in one hand, arm, leg, or jaw.
2. Rigidity: Stiffness or resistance to movement in the limbs and trunk.
3. Bradykinesia: Slow movements and decreased ability to initiate voluntary action.
4. Postural instability: Difficulty with balance and coordination, leading to an increased risk of falling.

5. Impairment of fine motor skills: Difficulty with tasks that require fine motor coordination, such as buttoning a shirt or writing.

When it comes to Parkinson's disease, there is no cure, but research has shown that exercise can be highly effective in improving symptoms and slowing down its progression. Exercise offers numerous benefits such as enhancing gait, balance, flexibility, and motor coordination, ultimately leading to a better quality of life for those living with Parkinson's.

Therefore, we have created a mobile application with twelve screens. The first two screens are splash screens, followed by a welcome screen where users can choose to log in or sign up. These screens enable users to access the app's various features, including Warmup Exercises, Brain Games, and different courses related to symptoms, once they have logged in. If you're having trouble finding a specific course on the homepage, there are various options available to help you. You can click on "See All" to view all available exercises. Once you find your desired course, simply click on it to open a new page where you'll find a list of all exercises. Once you click on the exercise, it will transfer you to a new page where it demonstrates how to do the exercise and provides some helpful tips. When you visit the homepage, you'll see a button with three arrows. This button will take you to the bottom navigator, which will assist you in navigating two new features. The first feature is called "Ask Me!" where you can ask any question related to symptoms, exercise, and more. The second feature is the application's Setting page.

2.2 Background

A study published in the journal "Movement Disorders" discovered that individuals with Parkinson's who engaged in regular exercise experienced better motor function, balance, and overall quality of life compared to those who didn't exercise regularly[**canning2015exercise**]. Another study published in "Neurology" indicated that high-intensity treadmill training significantly improved gait and balance for individuals with Parkinson's disease [**mehrholz2015treadmill**]. Additionally, walking, cycling, yoga, and Pilates have all proven to be effective exercises for people with Parkinson's, as recommended by the Parkinson's Foundation.

Different types of exercise have shown significant benefits for individuals with Parkinson's disease, including low-impact exercises like walking, cycling, pilates, and yoga. These exercises can be tailored to meet individual needs and can be performed at a pace that suits them best.

Recent research findings reveal that exercise positively impacts the overall quality of life for those with Parkinson's disease. A study in the Journal of Parkinson's Disease demonstrated that

exercise can enhance balance, and gait speed, reduce falls, and improve cardiovascular health for individuals with Parkinson's [**schenkman2018effect**]. Furthermore, a study published in the Journal of Neurology found that yoga has a positive influence on balance and functional mobility in individuals with Parkinson's disease [**wang2018influence**].

The ultimate goal of this project is to develop a user-friendly app that provides personalized exercise routines for individuals with Parkinson's disease based on their specific symptoms. By tailoring the exercises to target the specific motor function affected by the disease, this app has the potential to significantly improve their quality of life and help maintain their independence for a longer period.

2.3 Problem Statement

A lack of agreement exists on the best kind, frequency, and intensity of exercise for people with Parkinson's disease (PD), despite mounting evidence of the potential advantages of exercise for this population. The processes by which exercise may affect the signs and symptoms of PD and its development are also poorly understood. This knowledge gap has consequences for the creation of practical exercise therapies and the enhancement of PD patients' quality of life. What kinds of workouts can help slow down PD's symptoms and progression?

2.4 Project Scope

The purpose of this project is to develop a mobile application that provides customized exercise programs for people with Parkinson's disease. The app will allow users to select exercises tailored to their specific symptoms, with the goal of improving motor function, balance, coordination, mobility, and overall quality of life. Video guides will demonstrate proper form and technique for each exercise.

The app will include a library of exercise videos covering:

1. Brain exercises
2. Aerobic exercises
3. Balance exercises
4. Arm Strength training
5. Leg Strength training

6. Flexibility

7. Hand exercise

8. Hand shaking exercise

CHAPTER 3

Literature Review

There have been several studies on the effect of exercise on Parkinson's disease. These studies have shown that exercise can have a positive impact on the symptoms of Parkinson's disease, such as improving balance, gait, motor function, and overall physical function. The types of exercise that have shown promise include resistance training, balance and coordination training, aerobic exercise, and dance-based movements. Additionally, some studies have suggested that regular exercise may slow down the progression of Parkinson's disease, although more research is needed in this area. It is important to note that exercise should be tailored to the individual's needs and abilities and should be done under the supervision of a healthcare professional.

3.1 Evidence-based analysis of physical therapy in Parkinson's disease with recommendations for practice and research

A study by Keus et al. (2007) [**keus2007evidence**] analyzed data and established guidelines for physical therapy for Parkinson's disease, taking into account clinical knowledge, patient values, and global standards for guideline production. These guidelines serve as a basis for physical therapy in clinical practice and future research. Parkinson's disease is a neurodegenerative condition that affects cognitive, social, psychological, and physical well-being. Exercise programs may be beneficial in slowing down or preventing functional decline in patients. The study utilized mixed methods, including narrative, vote-counting, and random effects meta-analysis techniques, and reviewed 14 RCTs of moderate quality. Results indicated that exercise has positive effects on strength, balance, gait speed, and health-related quality of life. However, the efficacy of exercise in preventing falls or depression was inconclusive. The ideal exercise therapy for various stages of the disease is still a topic of concern (Goodwin et al., 2008).[**goodwin2008effectiveness**]

3.2 Exercise and Parkinson's disease

According to studies, regular exercise can help decrease the risk of developing Parkinson's disease (PD) and its associated complications. It can also mitigate other negative effects such as dyskinesia and wearing off. People with PD can benefit greatly from exercise, which can aid in reducing oxidative stress, repairing mitochondrial damage, and enhancing growth factor synthesis. Additionally, exercise is seen as a complementary approach to PD medications [**xu2019exercise**].

3.3 Randomized clinical trial of 3 types of physical exercise for patients with Parkinson's disease

Various studies have explored the potential of mobile apps and wearable technology in aiding individuals with Parkinson's disease during exercise. One such study, conducted by [**shulman2013randomized**], assessed the effectiveness of a smartphone app in encouraging physical activity among those with Parkinson's disease. The app offered tailored exercise plans, daily reminders, and instant feedback on performance. Results showed that participants who utilized the app experienced notable enhancements in their physical activity levels compared to those who did not use it.

3.4 A Smartphone-Based Tool for Assessing Parkinsonian Hand Tremor

A recent research study, conducted by [**kostikis2015smartphone**], investigated the effectiveness of a wearable device in monitoring and improving gait in individuals suffering from Parkinson's disease. This device offered instant feedback on gait performance and was able to track changes in gait patterns over time. The study concluded that the device was successful in enhancing gait parameters and reducing the risk of falls in Parkinson's patients.

3.5 The American College of Sports Medicine

The American College of Sports Medicine (ACSM) has published a comprehensive manual called The Guidelines for Exercise Testing and Prescription. This helpful guide provides

recommendations for developing safe and effective exercise programs for various populations, such as healthy adults, senior citizens, children, and individuals with chronic illnesses or disabilities. The tenth edition of the guidelines includes the latest information on exercise intensity, duration, and frequency, as well as suggestions for incorporating resistance training and other types of exercise into a complete fitness regimen.

3.6 ACSM's guidelines for exercise testing and prescription

The guidelines are based on the most recent research and best practices in exercise science. They stress the importance of customized, evidence-based interventions to improve health and fitness outcomes. Additionally, the guidelines offer specific advice for exercise testing and prescription based on individual requirements and goals[**american2013acsm**].

CHAPTER 4

Methodology

4.1 Technology

In this part of the section, we will discuss the technology & software that is been used in this project :

4.1.1 Choosing software for Android

Developing Android applications using Flutter with Visual Studio Code (VS Code)[**vscode**] offers a hassle-free and efficient development environment. VS Code's advanced features, such as its powerful code editor, integrated terminal, and numerous extensions, make it an ideal option for Flutter development. With the Flutter extension for VS Code, developers can access tools for code completion, debugging, and hot reloading, enabling them to write and test code with ease. VS Code's integration with Flutter's command-line tools simplifies project setup and management. Moreover, its compatibility with various programming languages and frameworks, as well as its active community support, ensures a seamless and productive development experience for creating high-quality Android applications with Flutter.

Figure 4.1: Visual Studio

4.1.2 Choosing software for the database

When it comes to choosing software for a database, MongoDB [**mongodb**] stands out as a strong and adaptable solution. As a NoSQL database, it offers numerous benefits that meet the needs of modern databases. Its schema-less structure allows for flexible data modelling, which can easily accommodate changes in application needs with minimal downtime. MongoDB's distributed nature supports horizontal scaling, allowing for effortless expansion as data demands increase. Its document-based architecture efficiently handles complex data structures, making it ideal for projects with semi-structured or unstructured data. Additionally, MongoDB's support for replication and high availability helps ensure data resilience and minimize the risk of data loss. Its query language, which is based on JSON-like documents, simplifies data retrieval and manipulation. All of these factors make MongoDB an ideal choice for database software, particularly for applications that need scalability, flexibility, and effective management of diverse data formats.

Figure 4.2: MongoDB Databases

4.2 Framework

For this project, we have used Dart and Flutter as a framework. Flutter is a framework for building applications, and Dart is the programming language used to write code for Flutter applications. Flutter is a UI (User Interface) software development kit (SDK) and framework developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. It allows developers to create visually attractive and high-performance applications with a reactive programming style. Dart, on the other hand, is the programming language used to write applications in Flutter. It is an object-oriented, class-based language with C-style syntax, and it is developed by Google as well. Dart is the language that Flutter is built upon and serves as the foundation for developing applications using the Flutter framework

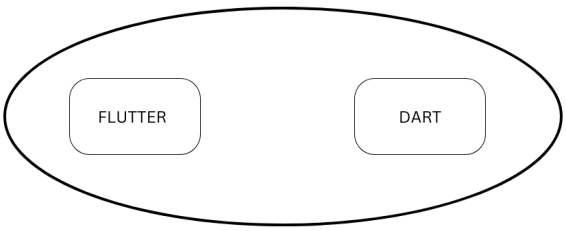


Figure 4.3: Flutter & Dart FrameWork

4.3 Dataset

Exercise is emerging as one of the most important treatments for managing symptoms and slowing progression of Parkinson's disease. As highlighted in the guide "Fitness Counts: A Body Guide to Parkinson's Disease," research increasingly shows that consistent physical activity can significantly improve motor and non-motor symptoms. Exercise essentially acts as medicine for both the body and brain in people with Parkinson's.

- Research shows exercise improves Parkinson's symptoms through promoting neuroplasticity - the brain's ability to rewire and compensate for damage.
- Aerobic exercise like walking, swimming, or dancing 3x a week for 30 minutes improves heart and lung health. Exercise at moderate to high intensity for maximum Parkinson's benefits. Calculate your target heart rate zone.
- Strength training 2-3x a week works all major muscle groups - core, legs, back, arms. Use weights, resistance bands, or body weight. It combats muscle loss and improves function.
- Balance exercises like tandem stance and agility drills help prevent falls. Vary movements and dualtask to challenge your brain as well. Do them 2-3x a week minimum.
- Stretching increases flexibility and fights rigidity. Target major muscle groups 2-3x a week. Perform static stretches held for 10-30 seconds. Focus on breathing.
- Skill-based exercises like Tai Chi, dance, yoga, and boxing improve coordination, mobility, and cognitive function through complex movements. Mix these with aerobic exercise.

4.4 Mobile Application Development

The application for this project is called Parkinson Exercise (PE) which is built on the Flutter & dart platform from scratch using the libraries provided by Flutter and third-party developers. The app is fully developed The following sub-chapters will give a detailed overview of the methods and tools required to create an Android application and the workings of its inner structures. It is important to get an overview of the Android system as a whole before beginning the development process.

4.4.1 Flutter Environment and Tools set-up

Setting up the Flutter environment and tools is a crucial step for developers looking to create robust and efficient applications using the Flutter framework [**zotero-177**]. Flutter, a popular open-source UI software development toolkit, allows developers to build natively compiled applications for mobile, web, and desktop from a single codebase. To begin the development journey, one must set up the necessary environment and tools. According to the official Flutter documentation, this involves installing the Flutter SDK, configuring the development environment, and setting up an Integrated Development Environment (IDE) such as Android Studio or Visual Studio Code. By following the guidelines provided by Flutter, developers can ensure that they have a stable and optimized environment for building innovative cross-platform applications.

Figure 4.4: Flutter & Dart

Flutter includes a variety of tools and features to aid in the development process. Some of the key tools included in Flutter are:

- Flutter SDK: This is the core software development kit that provides the framework and tools needed to build Flutter applications.

- Dart Programming Language: Flutter uses Dart as its primary programming language. Dart is optimized for building user interfaces and offers features like hot reload, strong typing, and asynchronous programming.
- Flutter Engine: The engine is the runtime environment that executes Flutter applications. It provides low-level rendering, input, and more.
- Widgets: Flutter offers a comprehensive set of customizable UI widgets that can be composed to create complex user interfaces. Widgets are building blocks of the UI and can be structured hierarchically.
- Hot Reload: One of Flutter's standout features, hot reload allows developers to see the changes they make in the code immediately reflected in the app without losing the app's state.
- Material Design and Cupertino Widgets: Flutter provides widgets that follow Google's Material Design guidelines for Android apps and Apple's Cupertino design for iOS apps. These widgets ensure a consistent and platform-specific look and feel.
- Packages and Plugins: The Flutter community has created a vast collection of packages and plugins that extend the functionality of Flutter apps. These packages cover various domains like navigation, networking, state management, and more.
- DevTools: A suite of performance and debugging tools that help developers profile and optimize their Flutter applications.
- Layout System: Flutter uses a flexible layout system that allows developers to create complex layouts with ease. The system includes concepts like rows, columns, grids, and more.
- Animations: Flutter provides powerful tools for creating animations, allowing developers to create smooth and visually appealing transitions and effects.
- Internationalization and Localization: Flutter has built-in support for internationalizing and localizing apps, making it easier to create apps that can cater to users from different regions.
- Platform Channels: Flutter allows communication between the Dart code and native code using platform channels. This enables integration with native device features and functionalities.

4.4.2 Android Emulator

The Android Emulator [**Virtual**] is a fundamental tool for Android developers, providing a simulated environment to test and debug applications without the need for physical devices. This versatile tool, as highlighted in the official Android Developer documentation, allows developers to mimic various Android device configurations, screen sizes, and hardware specifications. It is an integral part of the Android development workflow, enabling developers to ensure that their applications are compatible with and perform well on a wide range of devices and scenarios.

It is also highly recommended to set up a physical Android device like a smartphone or tablet to test your app on real hardware once development progresses. The AVD is great for faster iteration during coding, but testing on a real device is essential before releasing your app to users. Together, the AVD and a real Android device will provide the full environment needed to build, iterate, and thoroughly test your Android application throughout the development process.

Figure 4.5: Android Emulator

4.4.3 Mobile Architecture

The Parkinson's Exercise (PE) application is designed to provide targeted exercises and resources for individuals with Parkinson's disease. To ensure a seamless user experience and efficient development, the application architecture is divided into several key components:

- User Interface Layer:

  Screens: Different screens for exercises, user profiles, progress tracking, and resources.
  Navigation: Navigation between screens using Flutter's navigation framework.

- Business Logic Layer:

  Exercise Logic: Handling exercise routines, tracking progress, and adapting difficulty levels.
  User Profile: Managing user profiles, preferences, and settings. Resource Management: Handling the presentation of educational resources and materials.

- Data Layer:

  Local Data Storage: Storing user profiles, exercise history, and settings using Flutter's local storage mechanisms.
  Remote Data: Fetching exercise data, user profiles, and resources from a remote server using APIs.

- External Integrations:

  Third-Party Libraries: Utilizing third-party Flutter libraries for enhanced functionality, UI components, and data management.
  External APIs: Connect to external APIs to retrieve exercise guidelines, medical information, and additional resources.

- App Architecture Patterns:

  MVVM (Model-View-ViewModel): Implementing the MVVM pattern to separate UI, business logic, and data management.
  State Management: Using Flutter's state management solutions like Provider, Bloc, or Riverpod to manage application state.

- Testing and Quality Assurance:

  Unit Tests: Writing unit tests for individual components and logic.
  Integration Tests: Conduct integration tests to ensure smooth interactions between different parts of the app.
  User Testing: Gathering feedback through user testing to improve user experience.

- Deployment and Distribution:

  App Stores: Deploying the application on Google Play Store and Apple App Store. Continuous Integration/Continuous Deployment (CI/CD): Automating the build and deployment process for smoother updates.

4.4.4 App Architecture Patterns

App architecture patterns are structured design methodologies that guide the organization and interaction of components within a software application. These patterns provide developers with guidelines on how to design, organize, and manage different parts of an application to achieve specific goals such as maintainability, scalability, and code reusability. They help developers address common challenges, improve code quality, and make the application more robust and maintainable over time.

Several popular application architecture patterns are widely used in software development, including:

1. MVC (Model-View-ControllerMVC (Model-View-Controller) is a programming model that consists of three components: MVC divides an application into three main components: the Model (data and business logic), the View (user interface), and the Controller (which mediates between the Model and View). While effective, MVC can sometimes lead to tight coupling between the components.

2. MVP (Model-View-Presenter): Similar to MVC, MVP separates the concerns of data, UI, and interaction. In MVP, the Presenter acts as a mediator between the Model and View, reducing their direct interaction.

3. MVVM (Model-View-ViewModel) : MVM is a pattern that consists of MVVM enhances separation of concerns by introducing the ViewModel. The ViewModel holds the presentation logic and exposes data for the View, eliminating the need for direct interaction between View and Model.

4. MVI (Model-View-Intent) : MVI is commonly used with reactive programming paradigms. It focuses on unidirectional data flow, where the View sends user intents to the Model through the Presenter. The Model processes the intents and updates the View accordingly.

5. Clean architecture: Clean Architecture emphasizes the separation of concerns through multiple layers: entities, use cases, and interface adapters. The goal is to achieve independence between business rules and external frameworks.

6. Hexagonal architecture (ports and adapters): This pattern centers on creating an application with an inner core of business logic (ports) surrounded by external interfaces (adapters). This approach supports flexibility and adaptability.

7. Redux (state management): Redux manages an application's state in a single immutable store. Actions trigger state changes, and the store maintains the application's state tree. This pattern is particularly effective for managing complex application states.

8. Microservices architecture: This pattern structures an application as a collection of small, independent services communicating over APIs. Each service handles a specific functionality, making scaling and maintenance more manageable.

App architecture patterns offer developers frameworks for designing modular, maintainable, and scalable applications. The choice of pattern depends on factors such as the project's complexity, the development team's familiarity with the pattern, and the application's specific goals. It's important to choose a pattern that aligns with the project's requirements and adapt it to best fit the application's needs.

Ultimately, adopting a solid app architecture pattern can significantly contribute to the success of software projects by promoting code quality, ease of development, and long-term maintainability.

4.5 Database

A database is a computerised record-keeping system used to store large amounts of information or collections of computer files in order. In this project, we are using MongoDB [**mongodb**] as the database. MongoDB is a prominent NoSQL database management system designed for handling unstructured or semi-structured data. Unlike traditional relational databases, MongoDB offers a flexible and scalable approach to data storage and retrieval. It is particularly well suited for applications with dynamic and evolving data structures, such as mobile applications.

4.5.1 Logical Database Requirements

The backend system consists of Three tables: *Patient*, *Doctor*, and *Medical History*. The relationships between the tables and their descriptions are shown in Table 4.1.

Table	Description
Patient	Information relating to the user. **Patient ID** : The Patient ID will be used to log in and as well as use for Sign Up. **Patient Name** : The Patient name will be used for Sign Up.
Doctor	Information relating to the Doctor. **Doctor ID** : The Doctor's ID will be used for Sign Up. **Doctor Name** : The Doctor's Name will be used for Sign Up.
Medical History	**Gender** : This data will be stored for medical records. **Age** : This data will be stored for medical records. **Sugar** : This data will be stored for medical records. **BP** : This data will be stored for medical records. **Height** : This data will be stored for medical records. **Weight** : This data will be stored for medical records. **Symptom's** : This data will be stored for medical records.

Table 4.1: Description of Database Tables

4.6 Widget

In a Flutter app, widgets [**flutterwidget**] serve as the fundamental building blocks of the user interface. Every visual element displayed on the app is a widget, ranging from buttons, images, layout containers, and even the app itself. Immutable by nature, widgets are objects that describe a portion of the UI. Upon creation, widgets are configured by passing arguments to their constructor. Once constructed, a widget's properties cannot be modified. Instead, a new widget instance is created with new values.

A widgetś build() method returns other widgets that describe what should be displayed. Widgets are inserted into the widget tree and grouped by parent and child widgets in their respective build methods. Developers can utilize pre-built widget classes such as Text, Image, Row, and Column provided in the Flutter framework. Custom widgets can also be created by extending either Stateless Widget or Statefull Widget. These custom widgets allow for the encapsulation of UI widgets for reuse. The use of widgets facilitates the composition of UIs in a declarative manner. The widget tree is automatically rebuilt when the state changes, thanks to Flutter's reactive programming model. This efficient feature allows for UIs to react to changes seamlessly.

Now, I'm going to discuss what type of widget has been used while making this application.

Here are some common Flutter widgets that you can use to create material design-style apps:

4.6.1 Scaffold

This widget provides a basic structure for your app, including an app bar, a floating action button, a bottom navigation bar, and a content area.

4.6.2 SafeArea

Use this widget to ensure that your content is always visible and not obstructed by system-provided areas like notches, status bars, or navigation bars.

4.6.3 MaterialApp

This widget sets up a material design-specific application, providing features like a theme, navigation, and internationalization support.

4.6.4 Container

This rectangular layout element can contain other widgets and has properties like padding, margin, color, decoration, and alignment.

4.6.5 Column

Arrange your widgets vertically in a single column with this widget. It's useful for creating lists, forms, or any scenario where you need to stack widgets on top of each other.

4.6.6 Row

Arrange your widgets horizontally in a single row with this widget. It's used for laying out widgets side by side.

4.6.7 Positioned

This widget allows you to position a child widget within a 'Stack' using explicit coordinates, which is useful for layering widgets on top of each other.

4.6.8 Padding

Add space around your child widget with this widget. It's used to control the amount of padding between the child widget and its surrounding elements.

4.6.9 Center

This widget centers its child both horizontally and vertically within the available space. It's often used to center a single widget within its parent.

4.6.10 Stack

Overlay multiple widgets on top of each other with this widget. It's commonly used for creating complex layouts.

4.6.11 Text

Display a piece of text on the screen with this widget. It's customizable with various properties like font size, color, alignment, and more.

4.6.12 InkWell

Create interactive ink splashes when the user taps on this widget. It's often used for creating clickable elements that provide visual feedback when pressed.

4.6.13 GestureDetector

Recognize various gestures like taps, drags, and long presses with this widget. It's used to make any widget interactive and respond to user input.

4.6.14 SingleChildScrollView

Allow your child to scroll vertically when the content is too large to fit within the visible area with this widget. It's useful for situations where you have limited screen space but need to display more content.

4.7 External Library

1 cupertino_icons - Provides standard iOS-style icons for material design apps. Enables iOS aesthetic and conventions.

2 google_fonts - Allows easy inclusion of Google fonts in a Flutter app. Provides access to a large font collection.

3 Lottie - Renders After Effects animations natively in Flutter. Enables complex animations and motions.

4 flutter_svg - Draws SVG files in Flutter widgets. Allows vector images and icons in the app.

5 flutter_spinkit - Collection of loading indicators animated widgets for Flutter. Easy loading spinners.

6 HTTP Makes HTTP requests in Flutter. Enables networking and web API integration.

7 provider - State management for Flutter. Allows access to data across widget tree.

8 animated_text_kit - Animated text widgets for Flutter. Creates fancy animated text effects.

9 video_player - Video playback widget for Flutter. Enables video files in the app.

10 curved_navigation_bar - Customizable curved navigation bar widget. Provides animated nav bar.

11 mongo_dart - MongoDB driver for Dart and Flutter. Enables MongoDB integration.

12 path_provider - Finds commonly used locations on the filesystem. Helps read/write files.

13 ion icons - Icon font for web and mobile apps. Provides icons for material design.

14 carousel_slider - Carousel slider widget for Flutter. Enables image and card carousels.

15 smooth_page_indicator - Page indicator for page views with smooth animation. Works with carousels.

16 qr_code_scanner - QR code scanner plugin for Flutter. Provides camera-based QR scanning.

17 font_awesome_flutter - Font Awesome icon font for Flutter. Large icon collection.

18 flutter_custom_clippers - Custom clippers for Flutter. Lets you clip widgets in creative ways.

19 html_unescape - Unescapes HTML entities in strings. Decodes HTML-encoded text.

20 auto_size_ text - Automatically sizes text to fit containers. Adapts text size to fit.

CHAPTER 5

Design & Implementation

5.1 MongoDB

Throughout the application codebase, the constant. dart file defines common constants, such as the MongoDB connection URL, collection name, and main user ID. Defining these values as constants is helpful in avoiding hardcoding them in multiple places. It also provides a single source of truth for shared data elements. However, it is important to follow certain best practices when using constants.

For instance, it is not secure to directly embed sensitive data like usernames and passwords in the code. Instead, such values should be fetched at runtime from secure storage such as a secrets manager. Also, exposing the MongoDB collection name as a raw constant tightly couples the code to a specific collection. It is better to pass the collection name as a parameter to methods that need it. This way, the collection can be easily changed without updating every usage across files.

To enhance maintainability, constants should be named in an intuitive, self-documenting way. Units and data types should also be indicated for additional clarity. In general, constants effectively avoid duplication and improve maintainability by centralizing shared values. However, care should be taken not to inadvertently expose secure data or reduce decoupling. By following best practices, constants can enhance the quality and security of the codebase.

The MongoDbModel.dart file contains the MongoDbModel class that defines the structure of documents in the MongoDB collection used by the application. This class encapsulates various fields such as id, patientId, patientName, doctorId, and doctorName based on the data domain requirements of the app. Thus, it models the document schema in a reusable class. The MongoDbModel also provides fromJson() and toJson() methods to serialize instances of the model class to and from JSON format. This seamless integration with the MongoDB Dart driver ensures easy storage and retrieval of documents from the database.

Although the model maps the document structure correctly, it can benefit from some enhancements. It is crucial to add essential validations to ensure data integrity and business

rules. For instance, fields such as patient-Id or name could be mandatory or formatted in specific ways. It is also recommended to unit test the model by validating required fields and serializing to JSON. As domain requirements change, the model may also need to evolve and add new fields.

Overall, the MongoDbModel allows the representation of MongoDB documents in a concrete class instead of using free-form Map structures. This adds type safety and encapsulates the domain data format. Incorporating validations and testing will improve the model's robustness and reliability for the app's database operations.

The core logic for integrating with the MongoDB database used by the application is contained in the mongodb.dart file. This file establishes connectivity, provides access to the specified collection, and executes CRUD operations. The MongoDart driver is leveraged to connect to the Atlas MongoDB server using the provided connection URL. The db.open() method initializes the network connection. The userCollection object provides access to the collection specified in the constants. Methods like insert() and find() execute database operations to insert documents and fetch records from the collection. findOne() can query a single document by a field value. Although the basic database operations are implemented, improvements can be made to exception handling, validation, and logging. Robust error checking and retries should be added to handle network failures. Input validation ensures that bad data does not enter the database. Proper logging will aid in troubleshooting issues. The methods return a mix of Futures and direct data. A consistent asynchronous approach should be followed using Future-based returns. Connection pooling, security, and migrations can also be incorporated. Overall, this file forms the backbone to connect with MongoDB and perform operations. However, it has significant room for improvement by following best practices around security, DevOps, and reliability. A well-implemented database layer ensures that the app reliably stores and retrieves the data it needs to function properly.

The mongodbQuerry.dart file contains the MongoDbQuerry class, which represents query objects that are used to query the MongoDB database collection. This class includes the necessary fields and filters required to execute a query operation on the database. Currently, the MongoDbQuerry has only basic fields such as id and patientId. However, it should be expanded to include all possible fields that may be required to construct diverse query filters, such as patientName, doctorId, symptomType, exerciseType, and so on.

The query model comes equipped with toJson() and fromJson() methods that allow for the serialization of MongoDbQuerry objects to and from JSON format. This makes it possible to seamlessly integrate the query objects with the MongoDB Dart driver for execution. A robust

query model is critical for dynamically constructing and representing the diverse queries that may be needed in an application. The current implementation, with only id and patient-ID, limits the flexibility and reusability of the queries. Expanding MongoDbQuerry with more fields and utilizing it consistently for queries promotes code reuse. It also allows for the encapsulation of query construction logic in one place, rather than scattering it throughout the codebase.

Overall, the query model plays an integral role in interacting with the database and should be enhanced to support extensible and feature-rich query capabilities. This will improve maintainability, flexibility, and separation of concerns in the app's database access layer.

5.2 Application Architecture

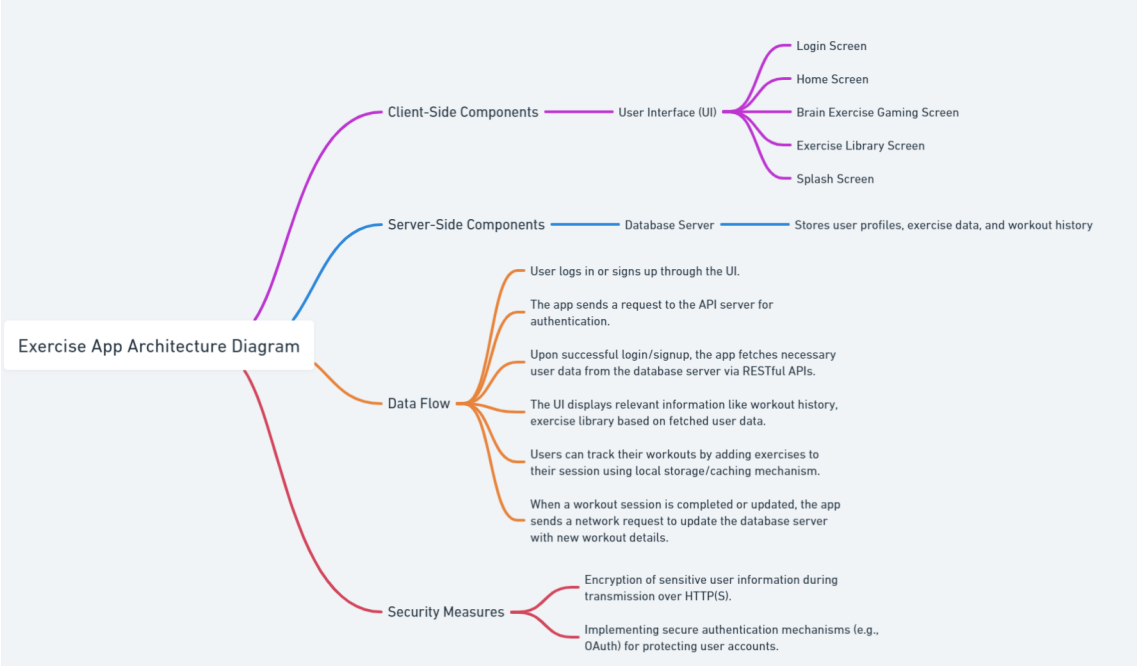


Figure 5.1: Architecture of Application

The Parkinson's Exercise app is built using the Model-View-ViewModel (MVVM) architecture pattern. This design ensures that the data, user interface, and business logic layers remain separate. The Model layer contains the data entities such as Exercise, User, Progress, etc. and methods to retrieve and update this data. This can be done from a local database or remote server. The View layer represents the UI components and screens that the user interacts with, such as ExerciseListScreen, ExerciseDetailScreen, and SettingsScreen. The ViewModel acts as a bridge between the Model and the View. It prepares the data from the Model into a suitable form that can be easily displayed in the UI. The ViewModel also contains presentation logic for UI behavior in response to user input and exposes commands that the View can invoke in response to user actions. This separation of responsibilities in MVVM makes the code cleaner and easier to maintain. The unidirectional data flow from Model to ViewModel to View enables effortless UI updates whenever underlying data changes. Overall, the MVVM architecture provides the Parkinson's Exercise app with a modular and testable code structure that can accommodate future growth and extension.

5.3 Application Design

Our mobile application consists of twelve distinct screens, designed to offer users an intuitive pathway to explore the app's diverse functionalities. The first two screens display the app's splash screen and lead to a welcoming interface where users can choose to log in or register. Once logged in, users can seamlessly access the app's Warmup Exercises, Brain Games, and a range of courses addressing various symptoms.

To enhance user experience, the homepage offers multiple avenues for effortless navigation. If users have difficulties locating a specific course, they can select the "See All" option to view a comprehensive list of available exercises. Upon identifying the desired course, a simple click will transition users to a dedicated page containing a comprehensive list of all exercises pertaining to that course. Selecting an exercise will redirect users to an instructional page, complete with demonstration videos and insightful tips.

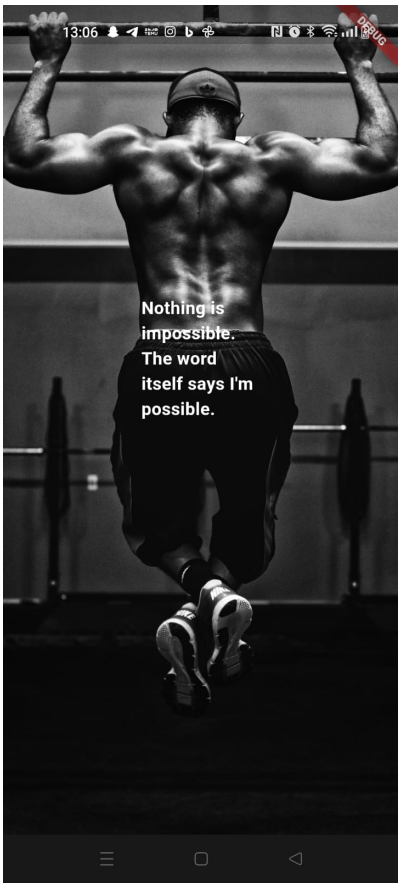
For seamless navigation towards additional functionalities, a conspicuous button featuring three arrows is positioned on the homepage. This button acts as a conduit to the bottom navigation menu, facilitating access to two new key features. The first feature, "Ask Me!", serves as a platform for users to pose queries related to symptoms, exercises, and related topics. The second feature corresponds to the application's settings, allowing users to customize their preferences and configurations.

Our mobile application design seamlessly integrates twelve unique screens, providing users with an intuitive pathway to explore its diverse offerings. Through strategically placed options and menus, users can effortlessly access courses, exercises, and interactive features while enjoying a user-friendly experience.

5.3.1 Splash-Screen

The splash screen is the first screen that appears when an Android application is launched. It displays an animation that introduces the app's motion upon launch. The splash screen acts as a window and covers an activity. It brings standard design elements to every app launch, but it's also customizable to maintain the app's unique branding.

(a) Splash-Screen 1



(b) Splash-Screen 2

Figure 5.2: Splash - Screen

5.3.2 Welcome Screen

Welcome to the app! On this screen, you will be presented with two choices.

- Login Screen
- Signup Screen

If you want to log in, click on the "login" option, and it will direct you to the login page. On the another hand, selecting the "sign up" option will take you to the sign-up page.

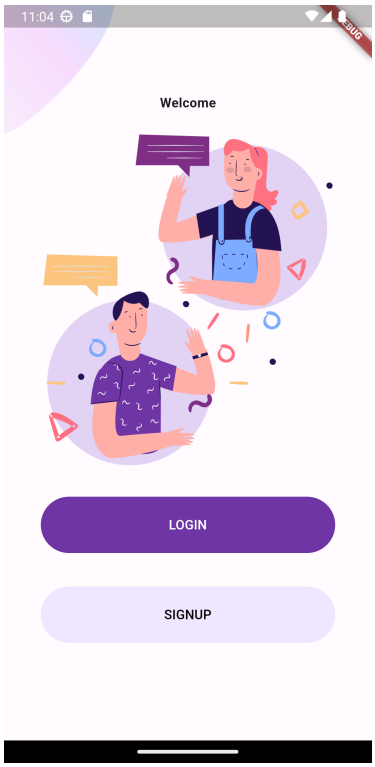


Figure 5.3: Welcome Screen

5.3.3 Login Screen

A login page is a screen that appears when you need to authenticate yourself before accessing an application's features. It collects your user credentials, like your username and password. In our case, the Patient ID is the login credential we use to authenticate with the database. If your credentials match with the database, you will be directed to the home page. Otherwise, you will be sent to the sign-up page.

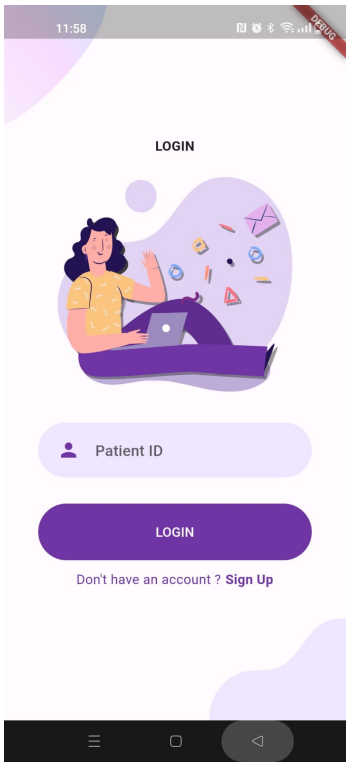
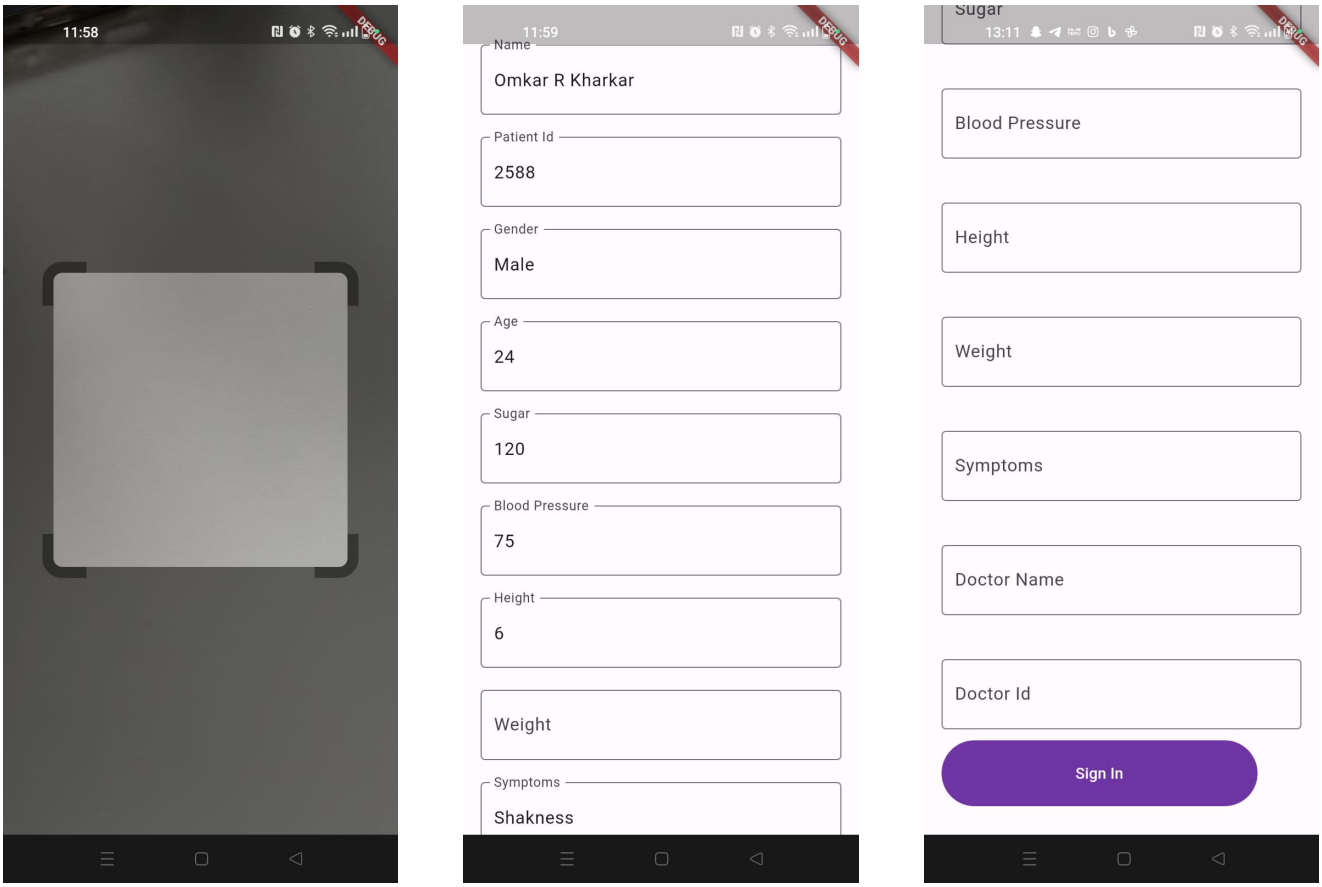


Figure 5.4: Login Screen

5.3.4 Signup Screen

When using an Android application, sometimes a user is required to create an account in order to access certain features. This is done to collect user information. In the case of our app, which is designed for people with Parkinson's disease, we have made changes to the signup process. Instead of manually filling out details, we have implemented a system where a barcode scanner is used. This is similar to when a doctor creates a report file for a patient and prints a barcode sticker containing all relevant data. When a user clicks on the signup page, the barcode scanner will be activated and the user can scan the barcode to automatically populate their details. To see what information will be displayed, please refer to table 4.1.



(a) Scanner (b) Signup Details 1 (c) Signup Details 2

Figure 5.5: Splash - Screen

5.3.5 Homepage Screen

Once you have successfully logged in, you will be taken to the Homepage/Dashboard of the application. The app will greet you with a message such as "Good Evening" and you will see a rounded icon next to it. Clicking on this icon will open a bottom sheet where you will have two options:

a. Ask me!

b. Settings.

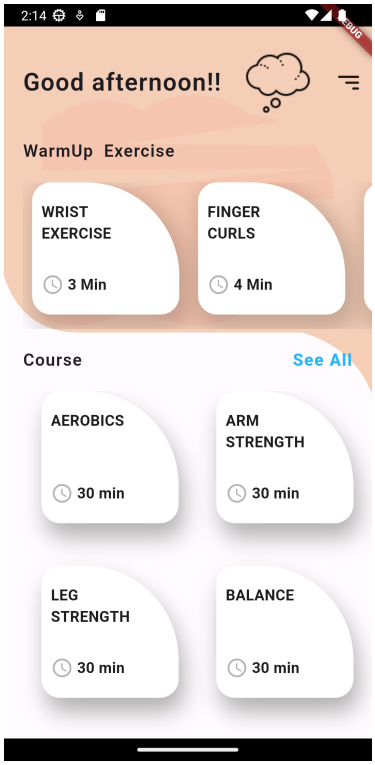


Figure 5.6: Home Screen

Warmup Exercise

Afterwards, you will be presented with some Warmup exercise options. Warmup exercises are essential to complete before any other exercise. There are four different types of warmup exercises to choose from.

Course Exercise

Additionally, there is a course option available, which displays only two different exercise sections. If you would like to see more exercises, you can click on the "See All" option. This will take you to a new screen where you can view all available courses.

5.3.6 Navigation Drawer Screen

In Android apps, a navigation drawer is a crucial feature that is commonly used. It is a UI panel that emerges from the side of the screen to reveal the primary menu options for an app. The navigation drawer makes it easy for users to access critical actions such as changing their profile, modifying app settings, and more. By providing a dedicated navigation surface, the drawer enables users to move seamlessly between various sections and features of an app. Although the drawer typically slides in from the edges of the screen, in this application, we have changed it so that it appears from the bottom side of the screen instead.

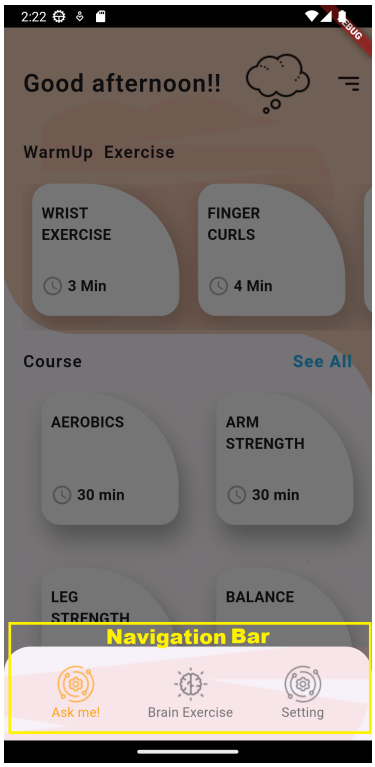


Figure 5.7: Navigation Drawer Screen 2

You may have noticed that there are now two new screens available: "Ask Me!" and "Settings". Let's take a closer look at each one.

Ask Me! Screen

First, let's talk about "Ask Me!". This feature allows you to easily connect with a doctor and ask any questions or address any concerns you may have about your exercise routine. It functions similarly to a messaging app.

Figure 5.8: Chat Screen

Brain Game Exercise Screen

Creating quiz games that cater specifically to individuals with Parkinson's disease is an excellent way to enhance their cognitive and communication abilities. These games offer an interactive and enjoyable way to stimulate various cognitive functions, practice verbal expression, and improve language skills. By offering a diverse range of quiz formats, including trivia, word association, memory challenges, and more, individuals with Parkinson's can maintain and even improve their mental agility while having fun. The customization of game difficulty levels and incorporation of accessible design elements ensure that these games are both enjoyable and beneficial for individuals at different stages of their Parkinson's journey.

Basically, there are two different types of games that we created. They are:

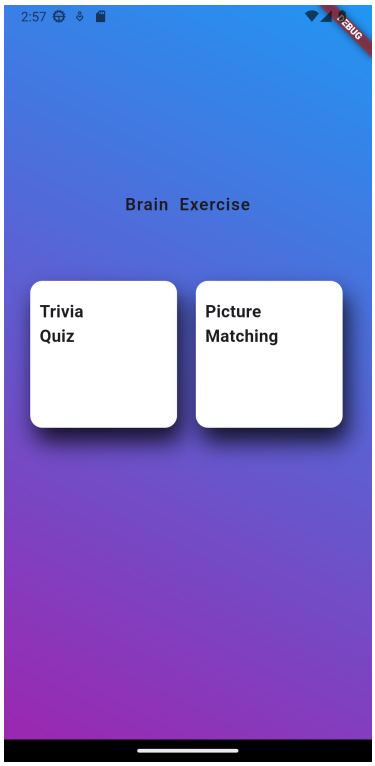


Figure 5.9: Brain Screen

- Trivia:

  The Trivia Quiz stands as a captivating and intellectually invigorating game that holds the potential to engage individuals in a dynamic exploration of knowledge across a spectrum of subjects. Participants are presented with a series of thought-provoking questions spanning various domains, including history, science, arts, and popular culture. This game not only serves as a splendid pastime but also encourages players to tap into their memory reserves, recall obscure facts, and exercise their cognitive prowess. As players eagerly strive to unravel the answers, their curiosity is piqued, and their cognitive agility

experiences a welcomed boost. The Trivia Quiz seamlessly intertwines entertainment with mental stimulation, making it an exceptional tool for enhancing cognitive function and fostering a delightful atmosphere of learning.

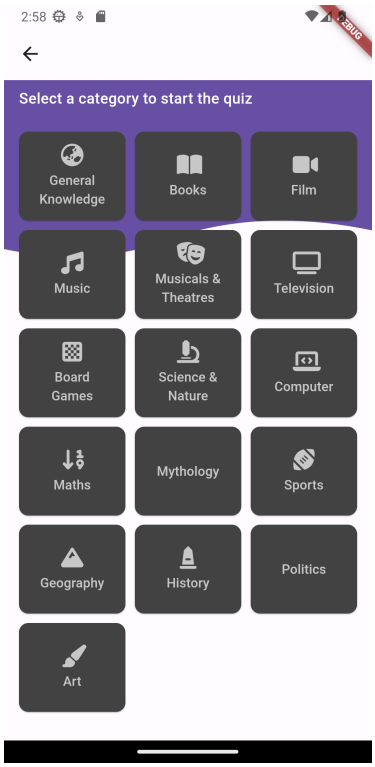
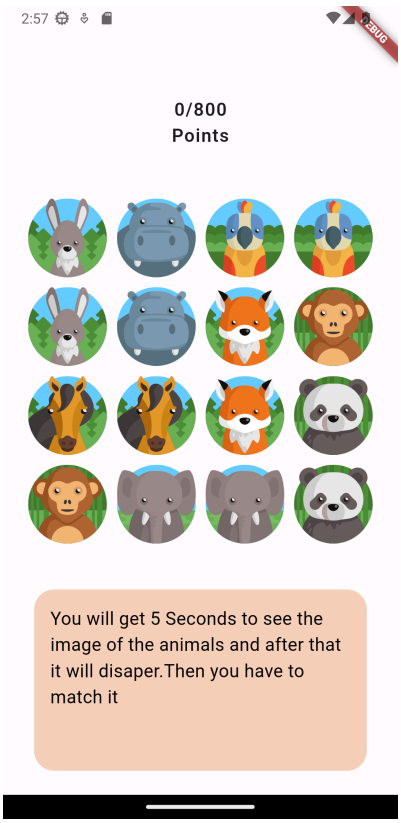


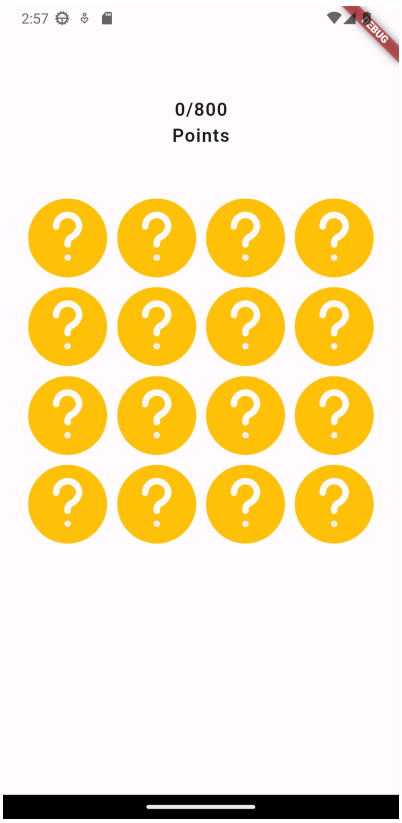
Figure 5.10: Trivia Screen

- Picture Matching:

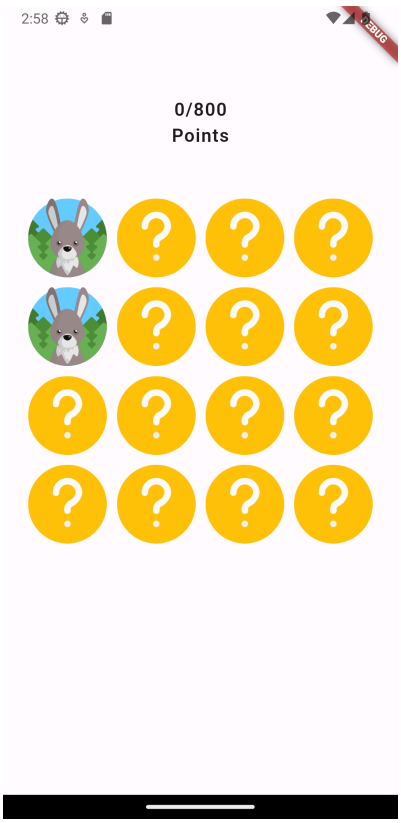
  The Memory Game presents an engaging and valuable opportunity for individuals to sharpen their memory and concentration skills while having fun. This captivating game involves the display of a sequence of images, numbers, or words for a brief period, challenging participants to memorize the sequence. As players strive to accurately recall the presented information, they exercise their short-term memory and enhance their focus and attention to detail. The Memory Game is a wonderful exercise for individuals looking to maintain or improve their cognitive abilities, and it offers an enjoyable way to challenge oneself and track progress over time. By fostering the connections within the brain responsible for memory retrieval, this game not only provides entertainment but also serves as an effective tool for cognitive enhancement.



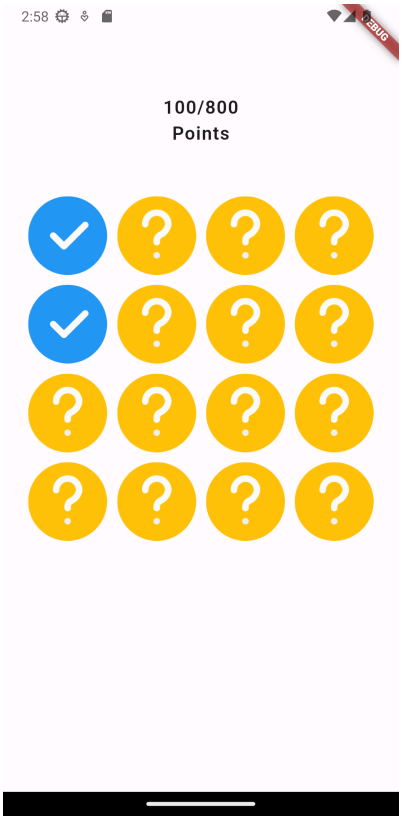
(a) Picture Matching 1



(b) Picture Matching 2



(c) Picture Matching 3



(d) Picture Matching 4

Figure 5.11: Exercise Selected Screen

Setting Screen

The "Settings" screen is a common and important feature found in many apps. It allows users to customize and control how the app operates. Users can adjust preferences and defaults to their liking, such as language, theme, and notification settings.

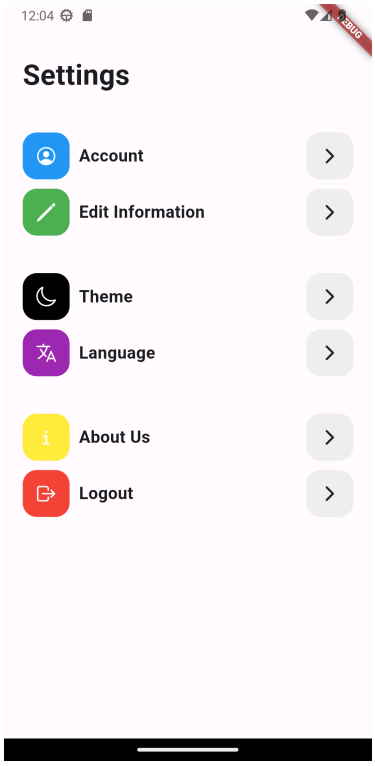


Figure 5.12: Setting Screen

5.3.7 Course Screen

As mentioned on the homepage, if you want to explore more courses, simply click on the "See All" option. This will take you to the Course Screen, where you can find a list of all available courses.

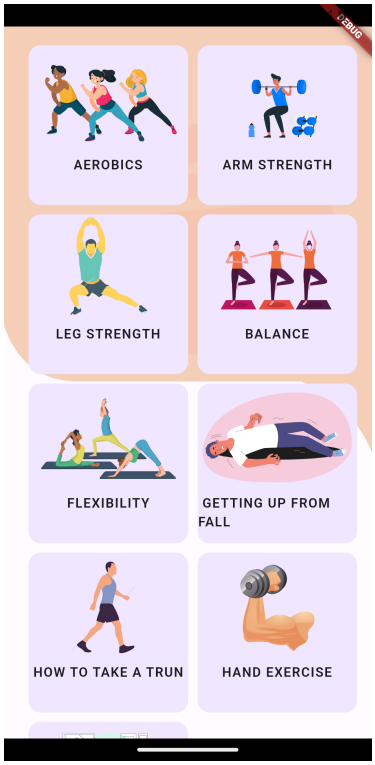


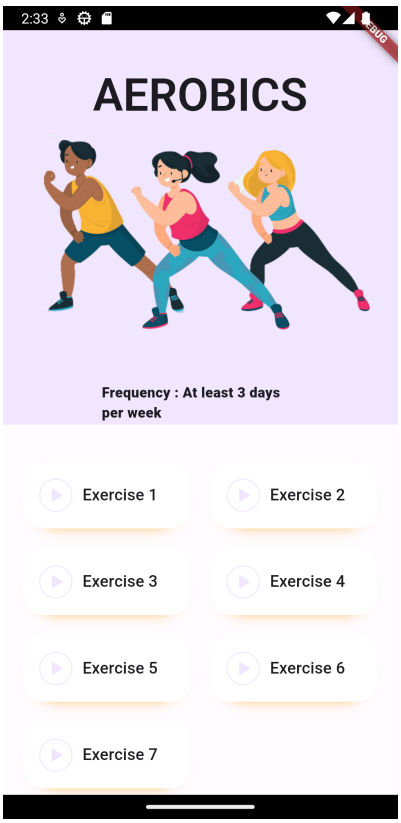
Figure 5.13: Course Screen

5.3.8 Exercise Selected Screen

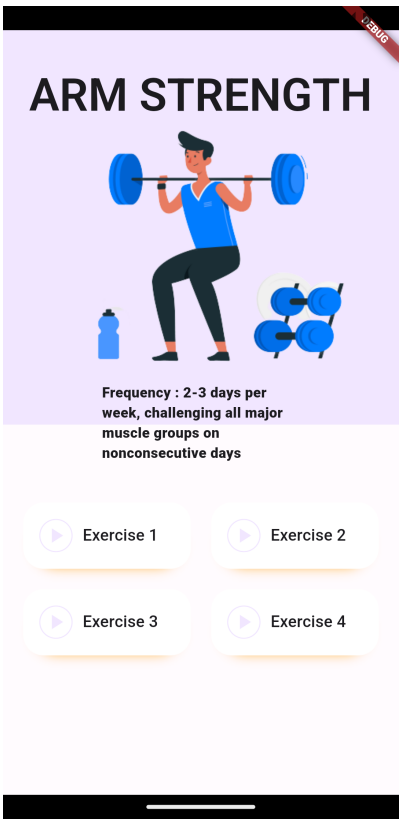
Once you have selected a specific course, you will be taken to the Exercise Selected Screen. Here, you can find all the exercises related to that course, along with the total number of exercises available. Please refer to the image titled "Exercise Selected Screen" (Figure 5.14) to view the additional screens that have been included.

5.3.9 Demo Screen

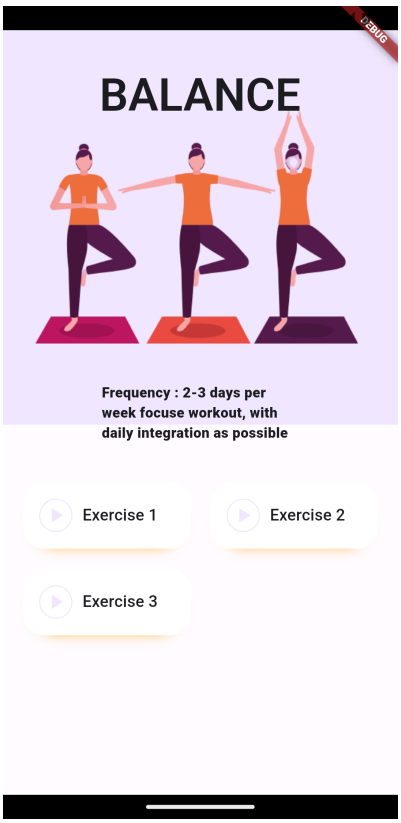
If you click on any of the exercise numbers, you will be directed to a new screen, where you can view a demonstration of the exercise and receive helpful tips. Please refer to the image titled "Exercise Selected Screen" (Figure 5.15) to view the additional screens that have been included.



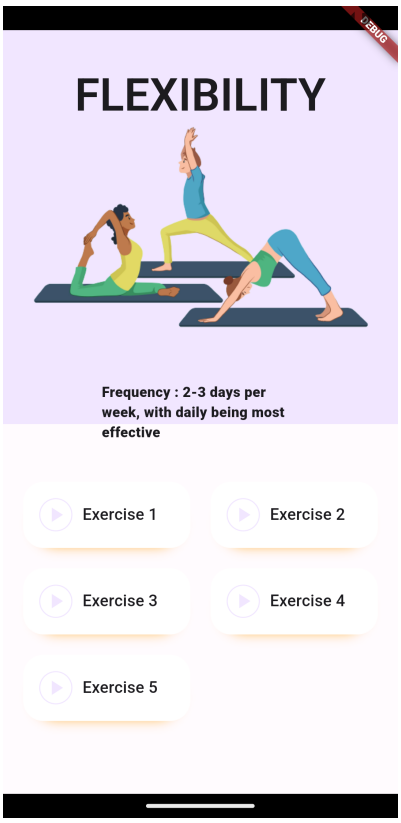
(a) aerobics



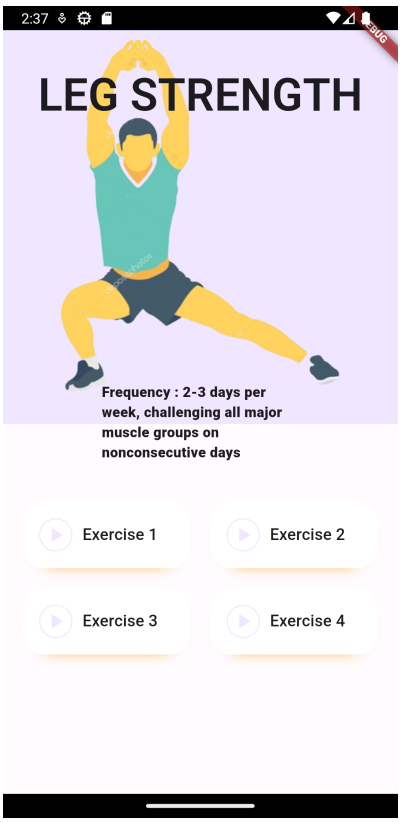
(b) arm



(c) balance

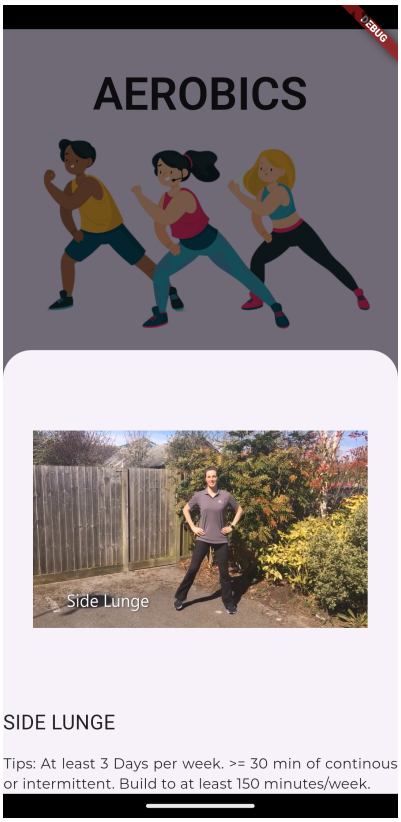


(d) flexibility

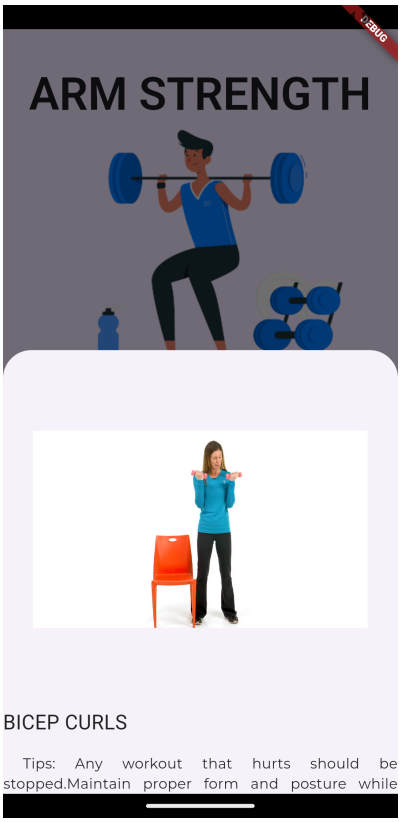


(e) leg

Figure 5.14: Exercise Selected Screen



(a) aerobics



(b) arm

Figure 5.15: Demo Screen

CHAPTER 6

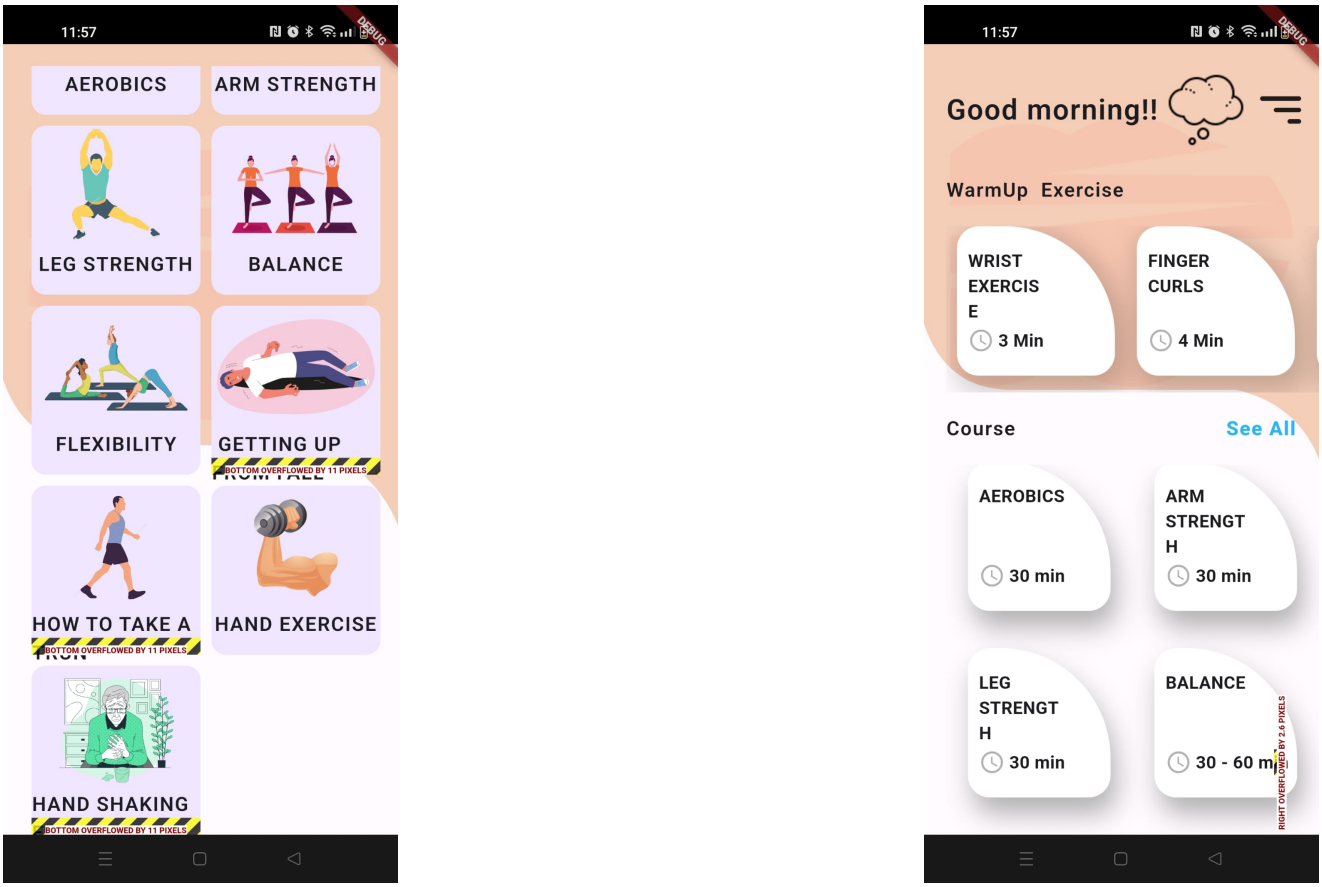
Testing and Evaluation

6.1 Application Testing with Android Phones

We conducted thorough testing of the Parkinson's exercise app across a diverse range of Android devices to ensure its functionality, performance, stability, and user interface are validated across various operating systems and hardware specifications. During the testing process, we focused on core app features such as exercise demonstrations, customization options, and account management. We rigorously evaluated each screen and user flow to identify any bugs and issues before its release. We paid particular attention to app responsiveness, navigation, layout adaptation, and network request handling. We closely monitored the app's impact on battery life, storage, RAM, and stability during prolonged testing periods, documenting and fixing any crashes or issues that arose. Finally, we performed extensive regression testing after updates to confirm the absence of new problems. This comprehensive testing methodology helped validate the app's capabilities and user experience for the target Android audience.

6.2 Error Found While testing the application

During comprehensive testing on various Android devices, we identified several issues that required bug fixing and troubleshooting. Some devices exhibited performance lag during video demonstrations, likely due to incompatible codecs or graphics capability. A few instances of app crashing occurred on older OS versions lacking support for newer APIs. Sporadic UI layout problems arose on devices with very large or very small screen sizes. Loading user profile data from the server sometimes timed out, indicating optimization was needed for network calls. Occasional stability issues like frozen screens or unresponsive buttons were traced back to flaws in underlying code logic. We used trace logging and debugging tools to pinpoint the source of errors such as null pointer exceptions or infinite loops. Identifying and addressing these errors and exceptions through the testing process was crucial for delivering a seamless user experience.



(a) Error 1

(b) Error 2

Figure 6.1: Error

6.3 Evaluation of testing application

Our thorough testing methodology proved effective in validating the app's functionality, user experience, and performance across the target Android environment. Testing on real devices ensured compatibility across various hardware and OS versions that customers will use. Rigorously evaluating each screen and component uncovered crucial bugs that were subsequently fixed, resulting in a higher-quality product. Incorporating user feedback during acceptance testing proved valuable, leading to UX improvements addressing actual patient needs. Monitoring resource usage enabled optimization of battery and performance. Extensive regression testing gave confidence that fixes and enhancements did not introduce new issues. While no testing process is foolproof, our layered testing strategy maximized coverage of critical app behaviours and use cases. Additional automated testing could potentially improve efficiency for rapid iteration. Overall, the testing enabled the delivery of a stable, user-friendly app that meets the essential requirements for assisting Parkinson's patients with customized exercise routines.

6.4 MongoDB Testing with Application

When developing an application with MongoDB, it is crucial to conduct thorough testing to ensure functionality, performance, and reliability. End-to-end testing is necessary, including the front-end, APIs, and database interactions. Unit tests should validate individual components such as controllers and models, while integration tests should confirm smooth interactions between the app frontend and the database. During testing, database calls should be mocked to avoid hitting the actual database. Test databases that imitate production serve as a sandbox for validation. Additionally, queries and indexes must be tested for performance with large datasets, and replication and sharding capabilities should be tested for fault tolerance. It is also essential to conduct security testing to prevent vulnerabilities. Automated UI tests can simulate actual user workflows, while load and stress testing can reveal capacity limits. Comprehensive testing across units, integrations, UI, security, performance, and infrastructure is key to delivering a robust application using MongoDB.

6.5 Error Found While testing the MongoDB

When testing an application with MongoDB, there are several common errors to be aware of:

- Connection errors: These can happen when the application is unable to connect to the MongoDB server due to issues with the URI, network, or authentication.
- Timeout errors: These occur when queries take too long to return, which can be an indication of indexing issues or non-optimized queries.
- Validation errors: These errors happen when attempting to insert data documents that don't match the schema, resulting in errors.
- Duplicate key errors: These occur when attempting to insert documents with the same value for a unique index, which will fail.
- Null reference errors: These happen when the code tries to access properties of null objects returned from the database.
- Race conditions: Concurrent operations from tests can generate unexpected orders of execution.
- Failing transactions: Transactions comprising multiple operations can fail atomicity.

- Indexing issues: The wrong index for queried fields can cause slow performance.
- Memory limit errors: Queries that return huge results can exceed the RAM and cause a crash.
- Shard key issues: Documents missing shard key or key type mismatch will result in an error.
- Replication lag: Tests may pass on one node but fail on another.
- Unauthorized access: Incorrect permissions can cause authentication errors.
- Data corruption: Hardware issues or concurrency bugs can corrupt data.

CHAPTER 7

Project Management

The Gantt chart in Figure 1 showcases the project plan and schedule, spanning 14 weeks. It consists of four major phases, namely, Data Generation and Design of the App, Interaction Between Apps, and Final Check.

Each phase comprises several tasks with estimated start and end dates, along with progress indicators. For instance,

- Phase 1 focuses on research.
- Phase 2 involves collecting data, designing app screens, and covering backend development.
- Phase 3 writing a report and testing.
- Finally, Phase 4 is dedicated to final demonstrations and report submission.

7.0.1 Phase 1

Phase 1 focuses on performing extensive research into existing solutions and approaches related to exercise recommendations for Parkinson's patients. This will encompass thoroughly reviewing clinical guidelines, academic studies, and industry offerings. Both traditional exercises and innovative techniques leveraging wearable and machine learning will be explored. Gaps and opportunities for improvement will be identified. Competitive analysis will shed light on differentiating features. This research will establish a strong foundation to inform subsequent design decisions and technology choices.

7.0.2 Phase 2

Phase 2 involves data collection, app design, and backend development. Real-world patient data will be gathered through partnerships with healthcare providers, ensuring a robust dataset. Concurrently, app screens and workflows will be designed to deliver an accessible and intuitive user experience. Backend infrastructure will be implemented, including a scalable

database, cloud servers, and core recommendation algorithms. Rigorous testing will validate all components prior to integration.

7.0.3 Phase 3

Phase 3 focuses on finalizing the front-end design, comprehensive testing, and report writing. Clinical partners will provide feedback to refine the UI/UX. Exhaustive user acceptance and functionality testing will uncover bugs and usability issues. Load and security testing will confirm performance and reliability. Technical documentation will detail architecture and code. A final report will summarize project learnings, analyze the effectiveness of the solution, and propose future work.

7.0.4 Phase 4

Phase 4 centres on demonstrations and final report submission. Showcases of the working application will be presented to key stakeholders. The final project report will be reviewed and submitted to conclude the effort.

you can refer to the website to see the project management **JIRA**

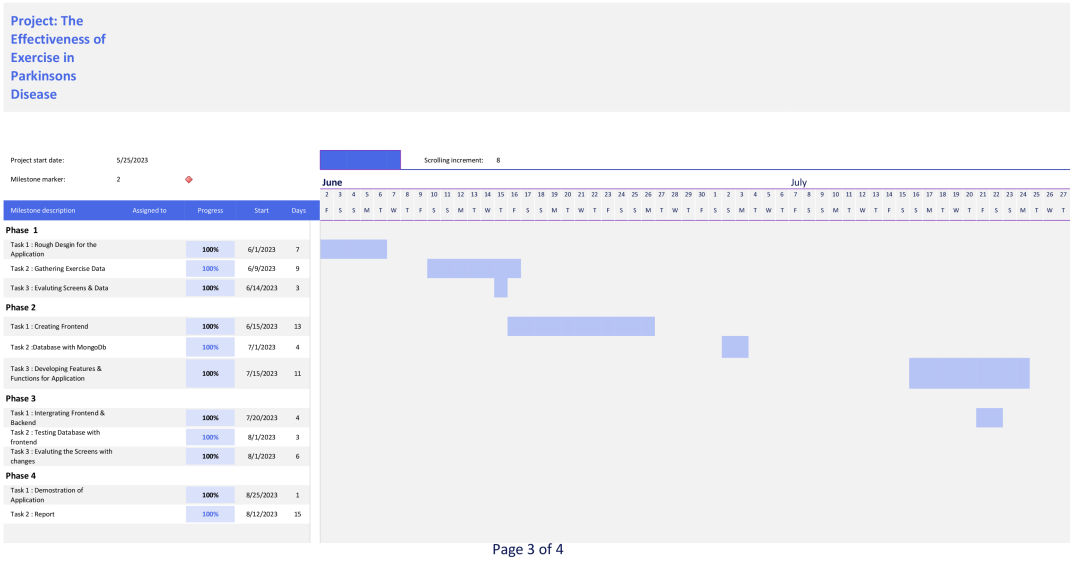


Figure 7.1: Gantt Chart

CHAPTER 8

Conclusion

I conducted a dissertation report to evaluate the effectiveness of exercise in improving symptoms and slowing the progression of Parkinson's disease. After reviewing various studies, I found that regular physical activity, including aerobic exercise, strength training, balance training, and flexibility exercises, can significantly enhance motor function, gait, balance, coordination, flexibility, and overall quality of life for individuals with Parkinson's. Although there is no cure for Parkinson's disease, exercise is becoming a critical component of disease management.

To make exercise more accessible for Parkinson's patients, I developed a mobile application that provides customized exercise routines. The app allows users to select exercises tailored to their specific symptoms and includes video guides showing proper techniques. I conducted thorough testing across diverse Android devices to ensure the app's functionality and user experience. During testing, I addressed some issues relating to performance, UI layout, and stability across different hardware and OS versions. Overall, the development and testing process resulted in a stable, user-friendly application that assists Parkinson's patients with specialized exercise needs.

CHAPTER 9

Future Scope

A. Studies have explored using virtual reality (VR) interventions to improve mobility in Parkinson's disease patients. [**amirthalingam2021virtual**]found Nintendo Wii-based play therapy improved posture. observed VR training significantly enhanced gait and balance. These support using VR for task repetition to promote neuroplasticity in Parkinson's.[**finley2021design**] designed a VR mobility training game incorporating personalized design and gamification for Parkinson's rehabilitation. VR can enhance neuroplasticity, sensory-motor integration, cognitive engagement, mirror neuron activation, and motivation systems.[**canning2020virtual**] Research topics include VR video games for motor function, gaming technologies for rehab, and VR in other neurological disorders.[**campo2021can**] Overall, studies demonstrate VR's potential for customized, engaging interventions to improve symptoms in Parkinson's disease.

The integration of virtual reality (VR), augmented reality (AR), mixed reality (MR) and artificial intelligence (AI) technologies presents exciting opportunities to enhance rehabilitation for individuals with Parkinson's disease. Below are some areas of future work that could further leverage these technologies:

1. Expand the VR mobility training game to incorporate more personalized and adaptive features using AI algorithms that analyze patient performance and adjust difficulty levels.
2. Develop more comprehensive VR environments and simulations to help patients practice real-world activities like walking, climbing stairs, or performing daily living tasks. The environments can be dynamically adjusted by AI to match patient needs.
3. Create an AR system that projects real-time cues, feedback, and instructions onto the physical environment as patients go through exercises and mobility drills. The system could use computer vision and AI to monitor and correct patients.
4. Create an AR system that projects real-time cues, feedback, and instructions onto the physical environment as patients go through exercises and mobility drills. The

system could use computer vision and AI to monitor and correct patients.

5. Build MR experiences that seamlessly blend real and virtual elements, creating engaging gamified exercises and simulations. Integrate biometrics to track patient engagement.
6. Incorporate predictive analytics using AI and machine learning models trained on patient data. This could enable early detection of disease progression and prompt personalized interventions.
7. Explore using AI-powered virtual coaches and assistants that offer customized guidance and support to patients during VR/AR therapy sessions.
8. Develop smart wearables with embedded AI that can track patient movements and symptoms in real-world settings outside of therapy sessions.
9. Create tools for clinicians to closely monitor patient progress through VR, analyze longitudinal data, and adjust interventions accordingly.
10. Conduct studies to clinically validate the use of these technologies and their impact on motor symptoms, quality of life, and long-term outcomes.
11. Explore the potential for in-home VR/AR rehabilitation kits with remote therapist monitoring to increase accessibility.

B. To improve the exercise app for Parkinson's patients, there are several potential upgrades that could be made. Firstly, the video exercise library could be expanded to include a wider range of exercises covering different muscle groups and difficulty levels. This could include beginner, intermediate, and advanced variations to cater to different abilities. Additionally, adaptive exercise progression plans could be implemented, which would increase difficulty over time as users become stronger and more proficient.

C. Another option would be to incorporate tracking features, allowing users to monitor their progress over both short and long-term periods. This could include metrics like workout duration, repetitions completed, weight lifted, and heart rate, which could be tracked per session. Longitudinal analytics would provide insight into trends over weeks and months, while graphs and dashboards would show functional improvements. This data could also inform adaptive exercise plans and keep users motivated.

D. Integrating wearable devices and sensors could provide valuable exercise and health data, including heart rate, calories burned, sleep, and activity outside of therapy sessions. With

user permission, biometric data could be synced to enhance exercise insights. Sensor integration would enable more holistic tracking of exercise efficacy and lifestyle impacts.

E. Social features could also be added to the app, allowing users to connect with one another. This could include user profiles, feeds, messaging, group exercise sessions, and challenges to help motivate continued usage and create a sense of community.

F. Partnering with healthcare experts, such as physical therapists and neurologists, could help optimize the app's exercise guidelines to best assist Parkinson's patients. Incorporating the latest research and clinical practices into video exercises and programs would enhance effectiveness.

G. Conducting more extensive real-world testing with Parkinson's patients would provide crucial feedback to refine the app. This could include observing patients performing app exercises and tracking impacts over time to reveal opportunities for improvement. Testing functionalities like tracking, social engagement, and wearable integration could uncover usability issues.

H. To expand accessibility, it would be beneficial to develop versions of the app for iOS, tablets, and the web. Enabling cross-platform cloud data syncing would allow a seamless transition between devices, while responsive web design would optimize accessibility and convenience.

I. Finally, leveraging emerging technologies like virtual reality could provide more immersive and engaging exercise experiences. This could include VR simulations to gamify rehabilitation routines and provide virtual coaches, AR overlays to demonstrate proper motions, and mixed reality to blend real-world integration. Advanced machine learning techniques could also be used to adapt plans based on changing user needs over time, providing hyper-customized exercise plans for optimal outcomes. With patient consent, integrating with electronic health record systems could inform safer, more tailored exercise recommendations based on medical history, enabling additional personalization of exercise routines to align with clinical guidelines and avoid conflicts.

CHAPTER 10

Links

10.1 Source Code

The source code of the project is available at:

GitLab The project file contains the following two main folders :

1. Assets: Which Contains all images, icons & video.

2. Lib: In these folders, you will get all the pages.

Bibliography

[1] J. Amirthalingam, G. Paidi, K. Alshowaikh, A. I. Jayarathna, D. B. Salibindla, K. Karpinska-Leydier, H. E. Ergin, and A. I. I. Jayarathna, "Virtual reality intervention to help improve motor function in patients undergoing rehabilitation for Cerebral Palsy, Parkinson's Disease, or Stroke: A systematic review of randomized controlled trials," *Cureus*, vol. 13, no. 7, 2021.

[2] P. Campo-Prieto, G. Rodr'ıguez-Fuentes, and J. M. Cancela-Carral, "Can immersive virtual reality videogames help Parkinson's disease patients? A case study," *Sensors*, vol. 21, no. 14, pp. 4825, 2021.

[3] Ivan E. Sutherland and others. *The ultimate display*. In *Proceedings of the IFIP Congress*, volume 2, pages 506–508, 1965.

[4] James M Finley, Marientina Gotsis, Vangelis Lympouridis, Shreya Jain, Aram Kim, Beth E Fisher. *Design and development of a virtual reality-based mobility training game for people with Parkinson's disease*. Frontiers in Neurology, 11, 2021.

[5] Colleen G Canning, Natalie E Allen, Evelien Nackaerts, Serene S Paul, Alice Nieuwboer, Moran Gilat. *Virtual reality in research and rehabilitation of gait and balance in Parkinson disease*. Nature Reviews Neurology, 16(8), 2020.

[6] Margaret Schenkman, Charity G Moore, Wendy M Kohrt, Deborah A Hall, Anthony Delitto, Cynthia L Comella, Deborah A Josbeno, Cory L Christiansen, Brian D Berman, Benzi M Kluger, others. *Effect of high-intensity treadmill exercise on motor symptoms in patients with de novo Parkinson disease: a phase 2 randomized clinical trial*. JAMA neurology, 75(2), 2018.

[7] Colleen G Canning, Catherine Sherrington, Stephen R Lord, Jacqueline CT Close, Stephane Heritier, Gillian Z Heller, Kirsten Howard, Natalie E Allen, Mark D Latt, Susan M Murray, others. *Exercise for falls prevention in Parkinson disease: a randomized controlled trial*. Neurology, 84(3), 2015.

[8] Jan Mehrholz, Joachim Kugler, Alexander Storch, Marcus Pohl, Bernhard Elsner, Kathleen Hirsch. *Treadmill training for patients with Parkinson's disease*. Cochrane database of systematic reviews, 8, 2015.

[9] Lisa M Shulman, Leslie I Katzel, Frederick M Ivey, John D Sorkin, Knachelle Favors, Karen E Anderson, Barbara A Smith, Stephen G Reich, William J Weiner, Richard F Macko. *Randomized clinical trial of 3 types of physical exercise for patients with Parkinson disease*. JAMA neurology, 70(2), 2013.

[10] Nikolaos Kostikis, Dimitris Hristu-Varsakelis, Marianthi Arnaoutoglou, C Kotsavasiloglou. *A smartphone-based tool for assessing parkinsonian hand tremor*. IEEE journal of biomedical and health informatics, 19(6), 2015.

[11] Samyra HJ Keus, Bastiaan R Bloem, Erik JM Hendriks, Alexandra B Bredero-Cohen, Marten Munneke, Practice Recommendations Development Group. *Evidence-based analysis of physical therapy in Parkinson's disease with recommendations for practice and research*. Movement disorders, 22(4), 2007.

[12] Victoria A Goodwin, Suzanne H Richards, Rod S Taylor, Adrian H Taylor, John L Campbell. *The effectiveness of exercise interventions for people with Parkinson's disease: A systematic review and meta-analysis*. Movement disorders, 23(5), 2008.

[13] Xiaojiao Xu, Zhenfa Fu, Weidong Le. *Exercise and Parkinson's disease*. International review of neurobiology, 147, 2019.

[14] American College of Sports Medicine and others. *ACSM's guidelines for exercise testing and prescription*. Lippincott Williams & Wilkins, 2013.

[15] *Download Python*. https://www.python.org/downloads/ Accessed on: February 4, 2023.

[16] *Flutter Documentation*. https://docs.flutter.dev/ Accessed on: February 4, 2023.

[17] Google. *Android Developer (2014) Workflow Introduction*. https://developer.android.com/studio Accessed on: 2014.

[18] Flutter. *Flutter Developer Workflow Introduction*. https://flutter.dev/multi-platform/mobile Accessed on: [Access Date]

[19] Google. *Android Developer Official (2014) website Setting Up Virtual Devices: Managing Virtual Devices*. https://developer.android.com/studio/run/managing-avds Accessed on: 2014.

[20] Microsoft. *Visual Studio Code*. https://code.visualstudio.com/Download Accessed on: [Access Date].

[21] MongoDB Inc. *MongoDB*. https://www.mongodb.com/ Accessed on: 2009.

[22] MongoDB Inc. *Flutter Developer Workflow Introduction*. https://docs.flutter.dev/ui Accessed on: [Access Date].