



THE EGYPTIAN E-LEARNING UNIVERSITY

Faculty Of Computer And Information Technology

Insomnia Eraser

Supervised by:

Prof: Amira Idrees

Eng: Nourhan Khashaba

Team members:

1-Anas Mahmoud Mohamed	(20-00668)
2-Mostafa Mohamed shawky gad	(20-01693)
3-Marwan hesham abdl monem	(20-01319)
4-Adham Alaa Abdel Azzeem	(20-01203)
5-Abdallah salah abdelkareem	(20-00539)
6-Marco Milad Mankaruios	(20-00023)

Table of Contents

CH 1 Introduction

1.1 Introduction.....
1.2 Problem Background.....
1.3 Problem statement.....
1.4 Significance of the project.....
1.5 Project Objective.....
1.6 Project Scope and Limitations.....
1.7 Project Software Requirements.....
1.8 Project Hardware Requirements.....
1.9 Project Expected Output.....
1.10 Report Outline.....

CH 2 RELATED EXISTING SYSTEMS

2.1 Introduction.....
2.2 Existing Systems.....
2.3 Overall Problems of Existing Systems.....
2.4 Overall Solution Approach.....
2.5 Summary.....

CH 3: SYSTEM REQUIREMENTS ENGINEERING AND PLANNING

3.1 Introduction
3.2 Feasibility Study
3.3 Requirements Elicitation Techniques
3.4 Targeted Users
3.5 Functional Requirements Definition.....
3.6 Functional Requirements Specification
3.7 Non Functional Requirements
3.8 Summary

CH 4: SYSTEM DESIGN

4.1 Introduction.....
4.2 Context Diagram.....
4.3 Data Flow Diagram (DFD).....
4.4 Entity Relationship Diagram (ERD).....
4.5 UML Use Case Diagram.....
4.6 UML Activity Diagram.....
4.7 UML Sequence Diagram.....
4.8 UML Class Diagram.....
4.9 Summary.....

CH 5: Mobile Application Implementation

5.1 Kotlin.....
5.2 Architectural Pattern (MVVM).....
5.3 Firebase.....
5.4 Splash Screen & Storyboard.....
5.5 Authentication Module Screens.....
5.6 Home Scree.....
5.7 Books Library Screen.....
5.8 Ai Medical Consultant.....
5.9 Healthy Food.....
5.10 Meditation & Relaxation.....
5.11 Chat with Doctor
5.12 Authentication
5.13 Chats

CH 6: The used Algorithm

6.1 Supervised Learning
6.2 Types of algorithms
6.3 Advantages Of Algorithms
6.4 Disadvantages Of Algorithms

6.5 Algorithms Work
6.6 Overfitting in Algorithms
6.7 Underfitting in Algorithms
6.8 Data Classification and Accuracy
6.9 The algorithm we chose to use Random forest

CH 7: Dataset

7.1 Dataset

CH 8: Reference

8.1 Reference

CH 9: Conclusion

9.1 Conclusion

CH 10: Future Work

10.1 Future Work

Chapter one
Introduction

1.1 Introduction

Sleep is a critical aspect of human health, allowing the body and mind to rest, repair, and rejuvenate. However, millions of people worldwide struggle with sleep insomnia, a condition characterized by difficulty falling asleep, staying asleep, or both.

Insomnia can have a significant impact on overall well-being, leading to daytime fatigue, irritability, impaired cognitive function, and an increased risk of chronic health conditions.

The causes of sleep insomnia are complex and multifaceted, ranging from stress and anxiety to medical conditions and environmental factors. Addressing insomnia often requires a multi-faceted approach, including lifestyle modifications, cognitive-behavioural therapy, and, in some cases, medication. With proper diagnosis, treatment, and self-care strategies, individuals with insomnia can improve their sleep quality and overall health.

At the heart of this project lies the goal we try as much as possible to connect the patient with several specialist doctors to promote a seamless connection between patients and a variety of specialist doctors.

By providing a flexible and smooth communication tool in addition to trying to treat the patient from salvation from providing some scientific health systems and proposals that help change the lifestyle to an appropriate healthy on

1.2 Problem Background

Background of the Problem of Sleep Insomnia

Sleep is a fundamental biological process that is essential for maintaining physical and mental health. It is during sleep that the body and mind repair and rejuvenate, allowing for optimal functioning during the day. However, for millions of individuals worldwide, sleep is not an easy commodity. Insomnia, a sleep disorder characterized by difficulty falling asleep, staying asleep, or both, affects approximately 10-20% of the adult population in the United States (American Academy of Sleep Medicine, 2023).

Prevalence and Impact of Insomnia

Insomnia is more common in women than in men, and its prevalence increases with age. Older adults are particularly susceptible to insomnia, with up to 50% of individuals over the age of 65 experiencing sleep disturbances (National Sleep Foundation, 2023). Insomnia can have a significant impact on overall well-being, leading to daytime fatigue, irritability, impaired cognitive function, and an increased risk of chronic health conditions, such as heart disease, stroke, and diabetes (American Academy of Sleep Medicine, 2023; National Sleep Foundation, 2023).

Causes of Insomnia

The causes of insomnia are complex and multifaceted, and often involve a combination of factors. Common causes of insomnia include:

Stress and anxiety: Stressful life events, such as job loss, relationship problems, or financial difficulties, can significantly disrupt sleep patterns (Morin et al., 2011; Perlis et al., 2006).

Medical conditions: Certain medical conditions, such as chronic pain, heartburn, and sleep pane, can interfere with sleep (National Sleep Foundation, 2023).

Medications: Some medications, such as antidepressants, beta-blockers, and corticosteroids, can have adverse effects on sleep (American Academy of Sleep Medicine, 2023).

Mental health disorders: Mental health disorders, such as depression and anxiety, are frequently associated with insomnia (American Academy of Sleep Medicine, 2023).

Lifestyle factors: Poor sleep habits, such as irregular sleep schedules, caffeine and alcohol consumption, and excessive daytime napping, can contribute to insomnia (National Sleep Foundation, 2023).

Diagnosis and Treatment of Insomnia

The diagnosis of insomnia typically involves a thorough medical evaluation and sleep assessment. In some cases, additional tests, such as a polysomnography, may be necessary to rule out underlying medical conditions (American Academy of Sleep Medicine, 2023). Treatment for insomnia depends on the underlying cause and may include:

Lifestyle modifications: Implementing good sleep habits, such as establishing a regular sleep schedule, creating a relaxing bedtime routine, and avoiding caffeine and alcohol before bed, can significantly improve sleep quality (National Sleep

Foundation, 2023).

Cognitive-behavioural therapy: Cognitive-behavioural therapy for insomnia (CBT-I) is a highly effective treatment that focuses on identifying and changing negative thoughts and behaviours that interfere with sleep (Morin et al., 2011; Perlis et al., 2006).

Medication: In some cases, medication may be prescribed to help individuals fall asleep or stay asleep. However, medication should be used as a last resort and only under the supervision of a healthcare provider (American Academy of Sleep Medicine, 2023).

Conclusion

Insomnia is a prevalent and debilitating sleep disorder that can have a significant impact on overall health and well-being. Understanding the causes, diagnosis, and treatment of insomnia is essential for effectively managing this condition and improving the quality of life for individuals affected by it. With proper diagnosis, treatment, and self-care strategies, individuals with insomnia can successfully manage their sleep and improve their overall health.

References

American Academy of Sleep Medicine. (2023). Insomnia. Retrieved from <https://aasm.org/missing-the-mark-melatonin-finding-best-treatment-insomnia/>

1.3 Problem Statement

In this project, we will discuss in detail a solution to a very large contemporary problem of insomnia. Insomnia is a common sleep disorder that can make it hard to fall asleep, hard to stay asleep, or cause you to wake up too early and not be able to get back to sleep. You may still feel tired when you wake up. Insomnia can sap not only your energy level and mood but also your health, work performance and quality of life. We target the people of all age who suffer from insomnia, and the causes of this disease varied from one person to another, and according to the age group, some of them suffer due to exposure to a psychological condition, trauma or stresses in life, travel and Poor sleep habits. we try to treat them in the best and simplest way through several different. We will provide the patient with some important information about the disease, help him discover it, and help treat it by communicating with the doctor at any time in the case of chronic disease. - If he has some other sub-problems, such as excessive thinking, anxiety, or drinking stimulants, we will provide him with his own library that contains everything he needs to get rid of these problems, as it contains a special health regimen for him (exercise, food), or listening to the Qur'an to get rid of thinking, or music calm and helps him sleep, with some medical and scientific advice for all that

1.4 Significance of the project

Treating sleep disorders is crucial to society as it contributes to enhancing individuals' overall health and psychological well-being. Improving sleep quality can increase productivity at work, mitigate negative effects on daily performance, and address anxiety prevalent in society. Furthermore, focusing on treating sleep problems helps

reduce the overall burden on public health and health care costs, ultimately benefiting society as a whole.

Treating sleep problems is crucial for individuals as good sleep plays a vital role in both physical and mental health. Lack of sleep can lead to negative effects such as increased stress, diminished concentration, and impaired memory. Additionally, sleep disorders can contribute to an elevated risk of chronic conditions like heart disease and diabetes. By restoring a proper sleep pattern, overall health and well-being can be enhanced, improving the individual's quality of life.

1.5 Project Aim and Objectives

The project aims to

- 1- Help patients identify and overcome insomnia through many means.
- 2- Make it easier for patients to find doctors.
- 3- Facilitate booking and managing appointments, as the patient can book at any time convenient for him, which saves him time.
- 4- Improve communication that patients and doctors can Communicate better through text or voice messages or chat through the application, which helps in exchanging information faster and easier.
- 5- Providing various medical programs that help change the patient's lifestyle.

1.6 Project Scope

This project will focus on helping People who suffer from insomnia can be saved from identifying most of the disease in the first place and alleviating

the disease by providing various medical programs that help change the patient's lifestyle in addition to providing a space for them. Interact with the writing via chat and voice messages initially and treat their problem Through many therapeutic methods, the application will be called Free to download from Play Store. The app also helps people who suffer from sadness, stress, overthinking and functioning in life It does not require direct treatment.

1.7 Project Software and Hardware Requirements

- Hardware requirements:**

- Small dataset hardware recommendations

- Software requirements:**

- Operating system

- SharePoint Server 2013

- SQL Server

- Requirements for Project Server 2013 features

-

1.8 Project Expected Output

- 1- helps People who suffer from insomnia can learn more about the disease.

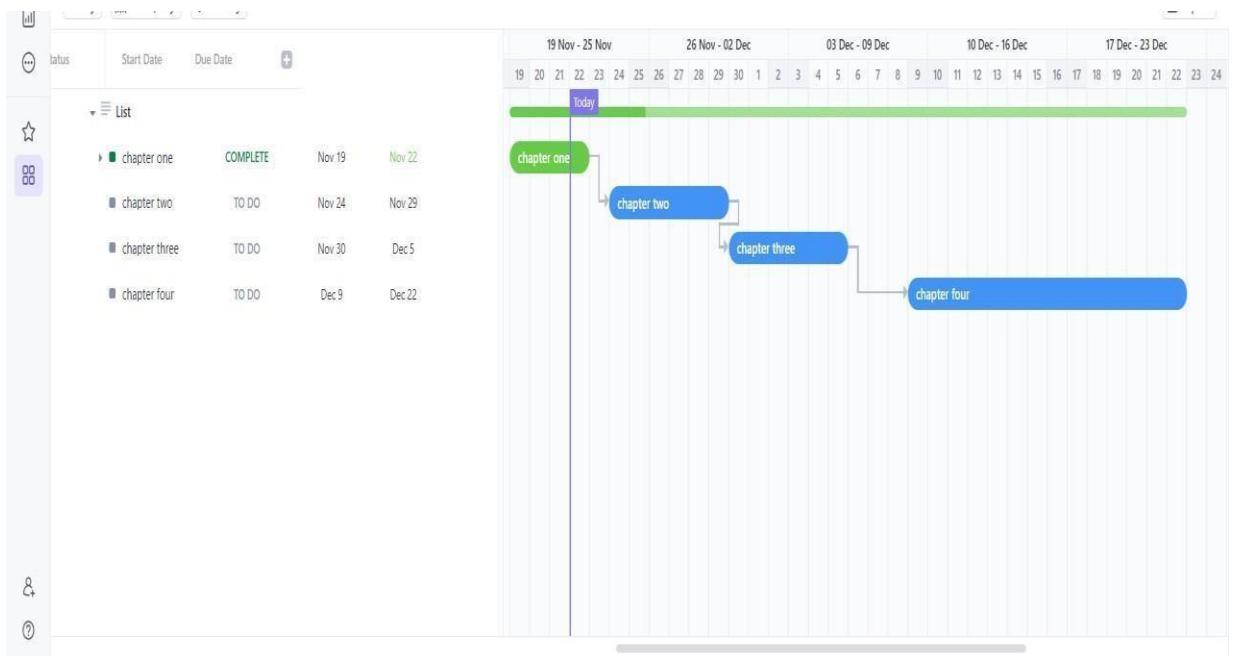
- 2- Alleviating the suffering of patients by offering various medical programs that help change the patient's lifestyle for the better.

- 3- Make space for them to communicate with doctors via chat and voice messages initially and solve their problem.

- 4- The application will be available Free to download from Play Store

5-The app also helps people who suffer from sadness, stress, overthinking and functioning in life It does not require direct treatment.

1.9 Project Schedule



1.10 Report Outline

This chapter gives a scope description and overview of everything, the remainder of this document includes four chapters as following:

Chapter 2: provides an overview existing systems and Overall Problems of Existing Systems, Overall Solution Approach system.

Chapter 3: provides an overview of the system requirements engineering and planning functionality in more detail. Also, explains functional and non-functional requirements for the overall system and users and feasibility study and targeted users.

Chapter two

RELATED EXISTING SYSTEMS

2.1 introduction

Many existing programs for insomnia rely on medication or relaxation techniques. While these can be effective, they don't always address the root causes of insomnia, nor do they provide a comprehensive solution. Our program takes a holistic approach to address your unique circumstances and help you achieve a better quality of life.

2.2 Existing Systems

Here are some of the Existing systems in insomnia apps:

Example 1 User Interface (UI):

This includes the design, layout, and interactive elements users engage with while using the app. It involves how users navigate through different features, access settings, and interact with the content.

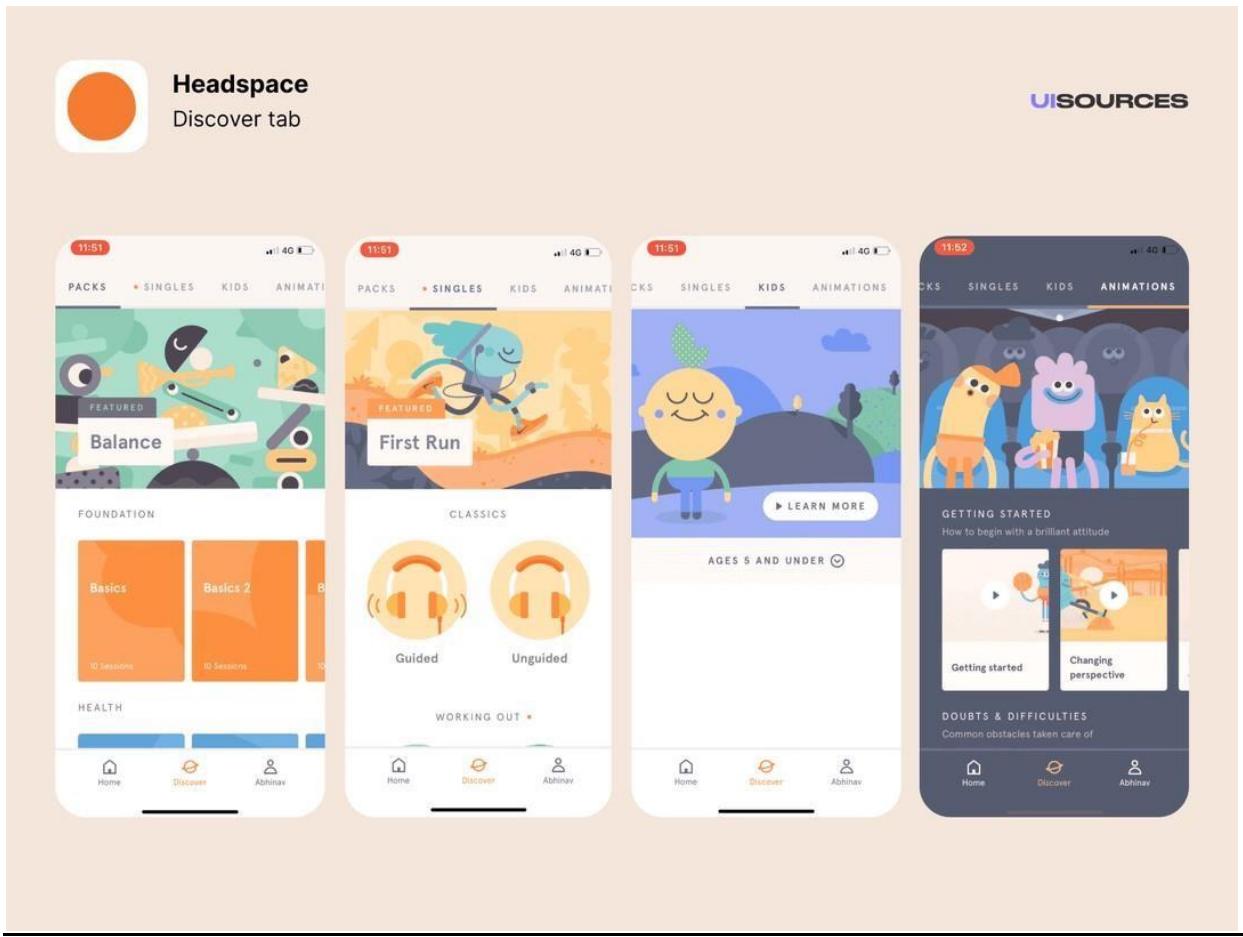
Benefits:

Enhanced User Experience: A thoughtfully designed UI makes it easier for users to navigate through different features and functionalities. It streamlines the user's interaction, reducing confusion and frustration.

Retention and Loyalty: A user-friendly UI can lead to increased retention rates and user loyalty. Users tend to stick with apps that provide a positive and hassle-free experience.

Reduced Errors and Frustrations: A UI designed with user experience in mind minimizes errors and prevents users from getting stuck or experiencing frustration due to unclear complicated navigation.

Graphical User Interface (GUI) Design for headspace :



Advantage	Disadvantage	Solution
Guided Meditation	Subscription Costs	More Free Content
Diverse Content	Content Limitations	Enhance Personalization
User-Friendly Interface	Dependency on Technology	Promote Offline Practice

Example 2 Backend Infrastructure:

This comprises the servers, databases, and other backend components that manage data storage, user authentication, and the overall functioning of the app.

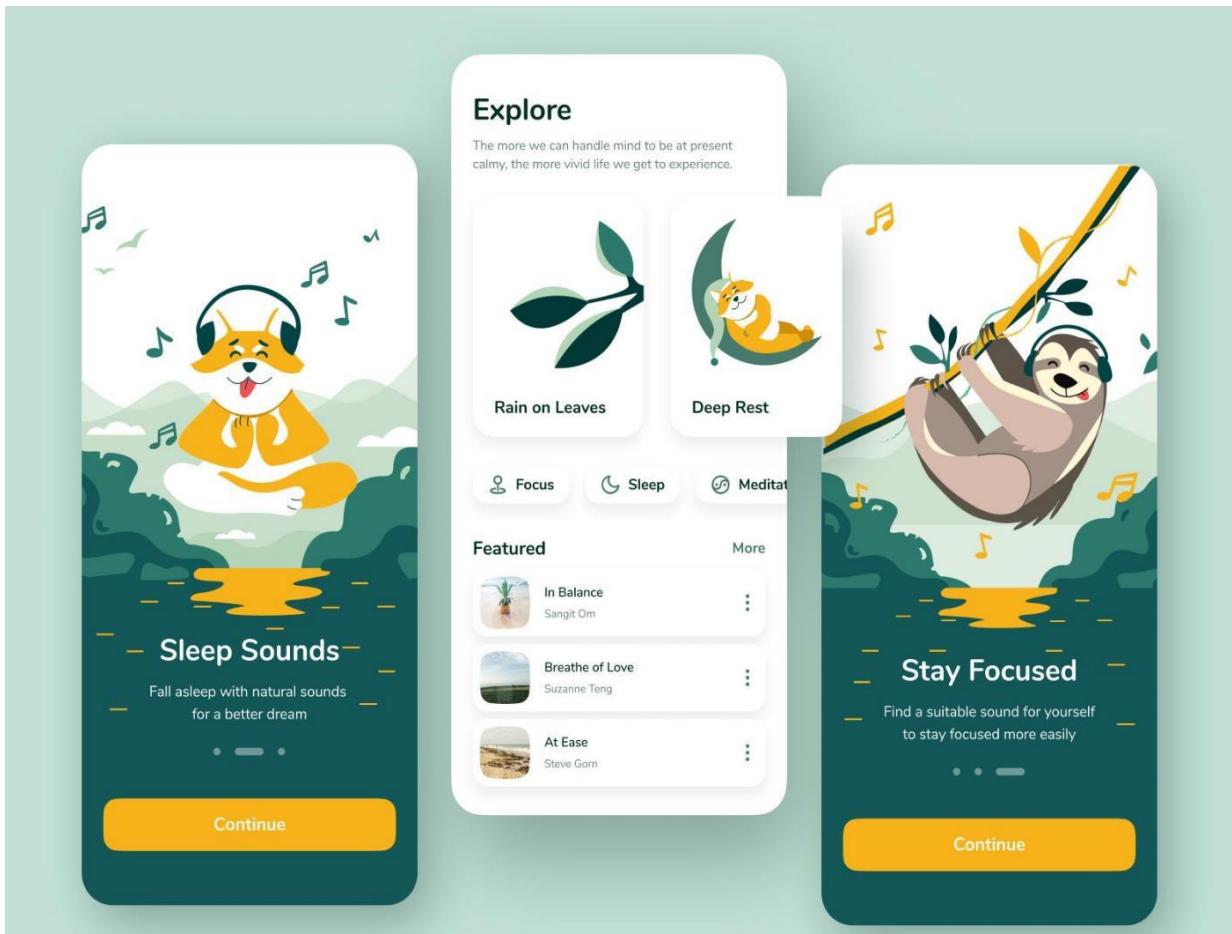
Benefits:

Reliability and Performance: A robust backend infrastructure ensures the app operates smoothly, providing reliable access to data and content, and maintaining optimal performance even during high traffic periods

Seamless Functionality: Backend systems facilitate the seamless operation of app features, data management, and third-party integrations, ensuring a cohesive user experience.

Data Security and Privacy: Backend infrastructure implements security measures to protect user data, ensuring it's encrypted, secure, and compliant with privacy regulations, fostering user trust.

Graphical User Interface (GUI) Design for calm



Advantage	Disadvantage	Solution
Varied Effectiveness	Subscription Costs	Enhance Personalization
Offline Access	Content Limitations	Promote Offline Usage
Relaxation Tools	Varied Effectiveness	Expand Free Content

Example 3 Functionality Modules:

These are the core components that deliver the app's functionality. For insomnia apps, it might involve modules for guided meditation, sleep tracking, relaxation techniques, sound generation, etc.

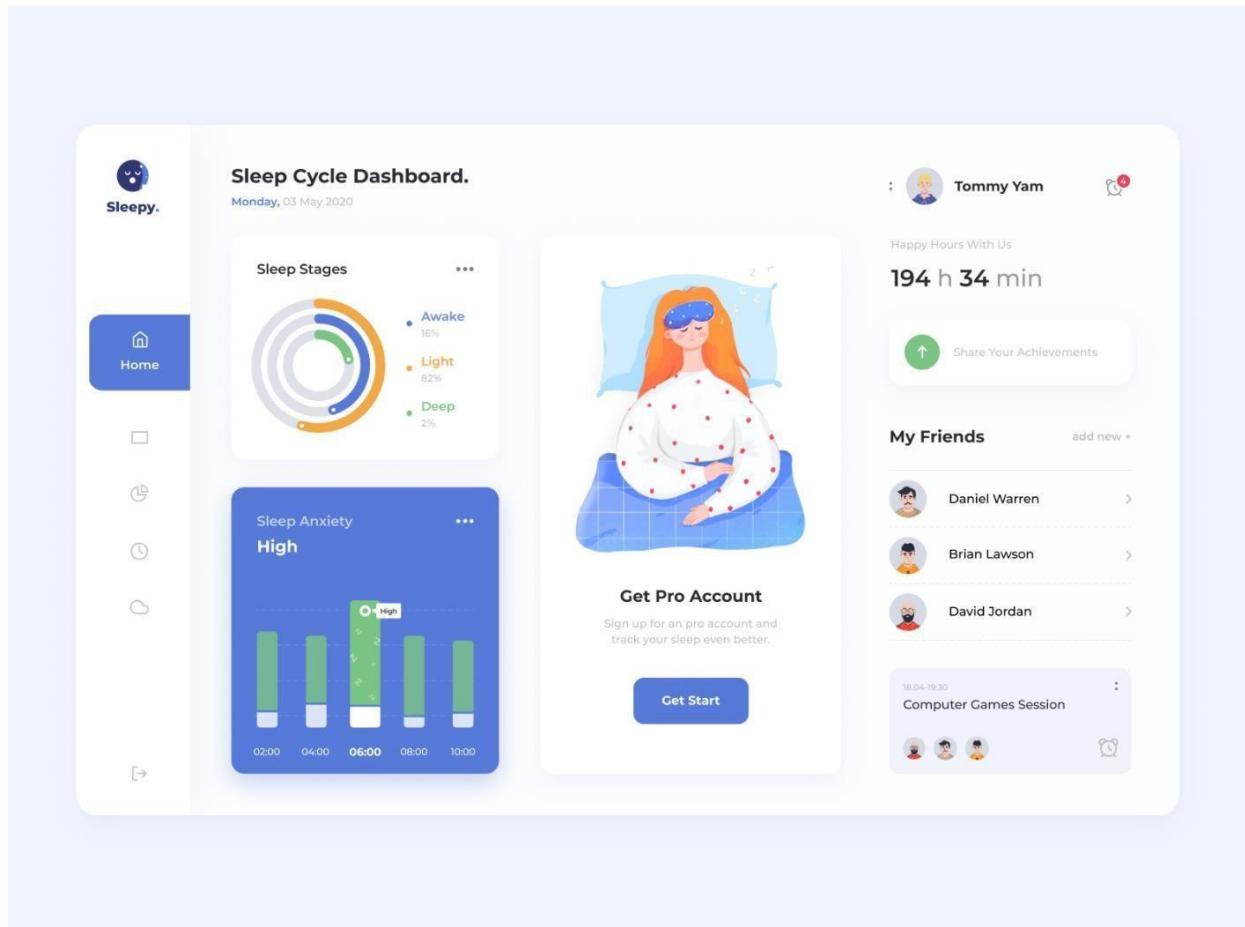
Benefits:

Scalability and Flexibility: Modular design facilitates scalability. Developers can add or modify modules as needed, allowing the app to evolve with user demands or technological advancements without major overhauls.

Easy Maintenance and Updates: Modules can be updated or maintained independently, making it easier for developers to fix bugs, add new features, or improve existing functionalities without affecting the entire app.

Faster Development Time: Modular design often speeds up the development process. Developers can work on separate modules simultaneously, reducing dependencies and allowing for quicker deployment of new features.

Graphical User Interface (GUI) Design for Sleep Cycle:



Advantage	Disadvantage	Solution
Sleep Tracking	Battery Consumption	Battery Optimization
Alarm Features	Device Placement Requirements	Enhanced Accuracy and Data Interpretation
Sleep Analysis and Trends	Accuracy Concerns	Diverse Tracking Methods

Example 4 Data Analytics and Tracking:

Many apps collect data on user behavior and preferences. Existing systems may involve analytics tools to understand how users interact with the app, which features they use most, and other relevant metrics.

Benefits:

Insightful Sleep Patterns: By analyzing sleep data, users gain insights into their sleep patterns, including sleep duration, sleep quality, and the various stages of sleep (light, deep, REM). This information helps users understand their sleep habits and potential areas for improvement.

Personalized Recommendations: With data analytics, apps can offer personalized suggestions or recommendations tailored to individual sleep patterns. These could include tips for better sleep, optimal wake-up times, or adjustments to improve sleep quality based on tracked data.

Identification of Trends and Anomalies: Data analytics can identify trends or irregularities in sleep patterns over time. It can detect patterns of better sleep on certain days or disruptions that coincide with particular events or behaviors, aiding in understanding external factors affecting sleep.

Graphical User Interface (GUI) Design for pzizz :



Advantage	Disadvantage	Solution
Dynamic Audio Technology	Subscription Costs	Free Content Access
Customizable Experience	Dependency on Technology	Offline Usage Promotion
Variety of Content	Effectiveness Variation	Enhanced Personalization and Customization

Example 5 Third-Party Integrations:

Some apps integrate with external services or APIs (Application Programming Interfaces) for functionalities such as payment gateways, data syncing with wearables, or content sharing.

Security Measures: This includes measures to protect user data, ensure secure transactions (if applicable), and prevent unauthorized access to sensitive information.

Benefits:

Enhanced User Engagement: Integrations with popular platforms or services users are familiar with can increase engagement. For example, integrating with fitness trackers or wearables can sync sleep data, allowing users to see correlations between activity and sleep quality.

Seamless User Experience: Third-party integrations can streamline the user experience by allowing users to access multiple services or functionalities within one app. This convenience can encourage longer app usage and higher user satisfaction.

Scalability and Rapid Development: Integrating third-party services often speeds up the development process. Instead of building features from scratch, integrating with established services can save time and resources, allowing for faster app updates and improvements.

Graphical User Interface (GUI) Design for Relax Melodies



Advantage	Disadvantage	Solution
Diverse Sound Library	Subscription Model	Offer Free Content
Customizable Mixes	Dependency on Technology	Offline Usage Promotion
Sleep Improvement	Effectiveness Variation	Enhanced Personalization

Example 6 Security Measures:

This includes measures to protect user data, ensure secure transactions (if applicable), and prevent unauthorized access to sensitive information.

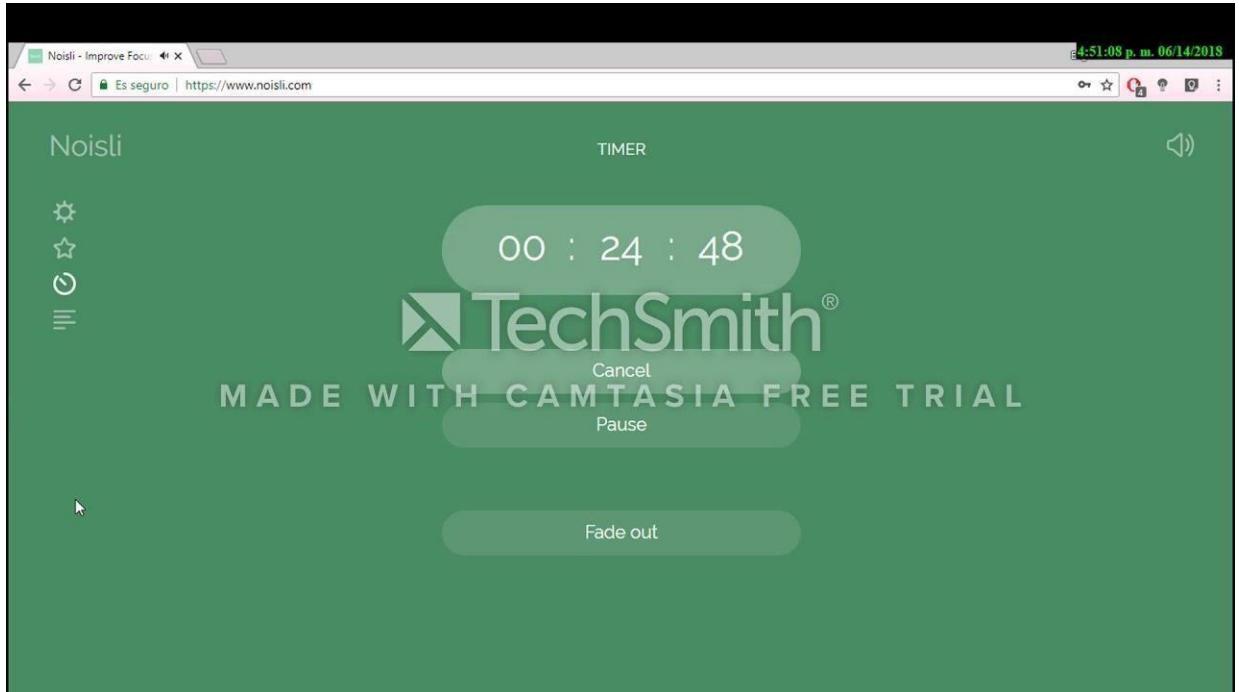
Benefits:

User Data Protection: Security measures safeguard sensitive user data, including personal information and usage patterns, protecting it from unauthorized access or breaches.

Maintaining Trust: By prioritizing security, apps build and maintain user trust. Users feel more confident using an app that prioritizes their data privacy and security, fostering long-term relationships.

Preventing Unauthorized Access: Robust security measures such as encryption, authentication protocols, and secure connections prevent unauthorized access to user accounts or sensitive data.

Graphical User Interface (GUI) Design for Offer Free Content:



Advantage	Disadvantage	Solution
Diverse Sound Library	Subscription Model	Offer Free Content
Customizable Mixes	Dependency on Technology	Offline Usage Promotion
Focus and Productivity	Effectiveness Variation	Enhanced Personalization

Example 7 Updates and Maintenance:

App developers regularly update their apps to fix bugs, introduce new features, or improve performance. The existing system includes mechanisms for delivering these updates to users.

Benefits:

Bug Fixes and Stability: Updates and maintenance address software bugs, glitches, or crashes that users might encounter. This ensures stability and smooth functioning of the app, reducing user frustration.

Security Enhancements: Updates often include security patches to address vulnerabilities or potential threats. This protects user data and the app from security breaches or cyberattacks.

Feature Enhancements: Updates may introduce new features, functionalities, or improvements based on user feedback or technological advancements, enhancing the overall user experience.

Graphical User Interface (GUI) Design for Sleepa



Advantage	Disadvantage	Solution
Improved Sleep Quality	Subscription Costs	Offer Free Content
Customization	Dependency on Technology	Promote Offline Usage
Convenience	Effectiveness Variation	Enhance Personalization

2.3-overall problem of existing system

- 1- **Effectiveness Variability:** Not all users respond the same way to the content or techniques provided in these apps. Some individuals may find certain methods or features ineffective for managing their insomnia.

- 2- **Limited Personalization:** Despite attempts at customization, some apps might still lack the depth of personalization needed to address the unique sleep issues or preferences of individual users.
- 3- **Dependency on Technology:** While these apps aim to aid sleep, they might inadvertently contribute to increased screen time or dependency on devices, potentially counteracting the goal of improving sleep quality.
- 4- **Subscription Models:** Many insomnia apps operate on subscription-based models, limiting access to premium content or features for users who prefer free resources.
- 5- **Data Privacy Concerns:** As these apps often collect sleep-related data, ensuring robust data privacy and security measures is crucial to maintain user trust.
- 6- **Effect on Sleep Cycles:** Introducing artificial sounds or lights to aid sleep might interfere with natural sleep cycles for some users, impacting sleep quality in the long run.
- 7- **Efficacy in Chronic Cases:** For individuals with chronic or severe insomnia, these apps might offer limited effectiveness compared to professional medical intervention or therapy.
- 8- **Accuracy in Sleep Tracking:** Some apps offering sleep tracking functionalities might lack accuracy compared to dedicated sleep monitoring devices, potentially providing misleading information about sleep patterns.

2.4 – Overall Solution Approach

1- Enhanced Personalization:

Implement AI-driven recommendations or algorithms to personalize content based on user feedback and sleep patterns.
Offer diverse content tailored to specific sleep issues or preferences, allowing users to create custom sleep plans.

2- Reducing Dependency on Screens:

Introduce offline functionalities that allow users to access content without continuous reliance on devices before sleep.
Promote relaxation techniques that don't require screen interaction, such as audio-only sessions or guided meditations.

3- Data Privacy and Security Measures:

Strengthen data encryption, secure storage, and transparent privacy policies to protect user data and build trust.

Ensure compliance with data protection regulations and regularly update security protocols to mitigate potential risks.

4- Accuracy in Sleep Tracking:

Continuously improve algorithms for sleep tracking to enhance accuracy and provide meaningful insights into sleep patterns.

Integrate with or allow compatibility with dedicated sleep monitoring devices for more precise sleep data.

5- Balanced Subscription Models:

Offer a blend of free and premium content to allow users to experience the app's benefits before committing to subscriptions.

Provide flexible subscription plans that cater to various user needs and preferences.

6- Professional Guidance Integration:

Collaborate with sleep experts or therapists to integrate professional guidance or educational content within the app.

Offer resources for severe cases, providing pathways for users to seek professional medical help if needed.

7- Continuous Improvement and User Feedback:

Regularly update the app, introducing new features, improving existing functionalities, and addressing user-reported issues promptly.

Encourage and incorporate user feedback to drive app enhancements and ensure it evolves based on user needs.

8- Educational Content and Sleep Hygiene Promotion:

Provide resources on sleep hygiene, educate users about healthy sleep habits, and promote behavioral changes to improve sleep quality.

2.5 - Summary

existing system is Depending on the :

- User Interface (UI)

- Backend Infrastructure
- Functionality Modules
- Data Analytics and Tracking
- Third-Party Integrations - Security Measures
- Updates and Maintenance

While current real systems offer many benefits, they also face several general problems:

- Effectiveness Variability
- Limited Personalization
- Dependency on Technology
- Subscription Models
- Data Privacy Concerns
- Effect on Sleep Cycles
- Efficacy in Chronic Cases
- Accuracy in Sleep Tracking

While current real systems have many issues, there are many solutions:

Enhanced Personalization

Reducing Dependency on Screens

Data Privacy and Security Measures Accuracy in Sleep Tracking

Balanced Subscription Models

Professional Guidance Integration

Chapter three

SYSTEM REQUIREMENTS ENGINEERING AND PLANNING

3.1 Introduction

The body of psycho-behavioural and neurocognitive empirical evidence describing the precise mechanisms that underlie the link between insomnia and negative mood is thin. However, subjective sleep insufficiencies and dysregulated mood observations exhibited more robust relationship as compared to objective findings from polysomnography or even fullsetup sleep electroencephalography data. This points to the fact that psychological factors that hinder sleep efficiency might play significant roles in justifying the sleep.

3.2 Feasibility

Our system is not related to a specific organization helps anyone who suffers from Insomnia or needs to relieve stress by communicating with a psychiatrist and getting a solution through a treatment session or group therapy and can ask a health care to read books or organize his healthy life in general.

Technical Feasibility

The system is feasible technically, although there is some risk.

- Users and doctors have enough experience about how to deal with the system such as how to log in and deal with the system.
- Team members understand all the processes that will be carried out on.
- There are many systems on the Internet that provide almost similar services.
- The system is easy to deal with different age groups.
- Medical information and transactions under the supervision of specialists.

–Risks related to the development team:

- Developers are familiar with the technology used in "Android".
- The tools used may be a little expensive.

–Risks related to users:

- Some users do not have experience in dealing with such a system.

- The user may be concerned about the information in their profile and information leakage.
- Some users may deal inappropriately, harass the team of doctors or other users, send spam, and so on

The project size MEDIUM:

- 1- The project team consists of six people.
- 2- Although the system contains a medium variety of features. (depend)
- 3- The project time frame is too big, but we will conquer this problem by working together in parallel.
- 4- Compatibility with the system medium existing technical infrastructure should be good due to An Internet infrastructure is already available for many people. Also, there is moderate compatibility in our system due to there being less interface with other system.

Organizational Feasibility

- The Champion:** The development team and supervisors provide time and effort for the system.
- The organizational management:** The development team and some clinics are interested in the project.

-Project sponsor:-

- Doctors who want to find work through our system.
- Development team.
- Mental health organizations.

-System users:

- Doctor: Treats patients, responds to consultations, update patient's doctor.
- User: Join Session or group therapy, make a test, read books, evaluate (rate) doctor.
- Project manager: The development team should have enough details about all users in the system and store the information about them in the database.

Economical Feasibility

–Tangible benefits:

- We will gain a little income from users who make an appointment with a doctor.
- Use the premium version of the system which includes the advanced features in the system.
- From ads, which user who cannot book an appointment at its price with a doctor or uses the advanced features in the system should view the ads to get these features for a free or lower cost.

– **Intangible benefits:**

Our team is a startup for the development team. We think that this project will help us to make other projects and learn more about tools that will be used in the system. It will decrease the problem of the embarrassment of going to a psychiatrist and helping anyone with insomnia problem.

3.3 Requirements Elicitation Technical

Requirements Elicitation Technical Requirements elicitation techniques are methods used to gather and capture the needs, expectations, and specifications of stakeholders for a software application like an insomnia application. Here are some commonly used techniques for eliciting requirements:

1-Interviews: Conduct one-on-one or group interviews with potential users, sleep experts, psychologists, healthcare professionals, and other stakeholders. Ask open-ended questions to gather insights into their sleep patterns, challenges, preferences, and expectations from the application.

2-Surveys and Questionnaires: Design and distribute surveys or questionnaires to a wide audience to collect quantitative and qualitative data on sleep habits, preferences, and desired features. Analyse the responses to identify common patterns and requirements.

3-Focus Groups: Organize interactive group discussions with a diverse set of stakeholders to encourage brainstorming, idea sharing, and gathering different perspectives on the desired features and functionalities of the insomnia application.

4-Observations: Observe and analyse the sleep behaviours, routines, and patterns of potential users. This can be done through in-person observations, video recordings, or using sleep tracking devices. This technique provides valuable insights into real-life sleep challenges and can help identify specific requirements.

5-Prototyping: Create prototypes or mockups of the insomnia application to gather feedback and validate requirements. Stakeholders can interact with the prototype to provide feedback on the user interface, features, and overall user experience.

6-Workshops and Collaborative Sessions: Organize workshops or collaborative sessions with stakeholders to facilitate discussions, idea generation, and requirements prioritization. Use techniques like brainstorming, affinity diagramming, or storyboarding to gather requirements effectively.

7-Document Analysis: Review existing documents such as research papers, medical guidelines, sleep studies, or industry standards related to insomnia and sleep disorders. Extract relevant information to identify requirements and ensure the application aligns with established best practices.

8-Use Cases and User Stories: Develop use cases or user stories that describe specific scenarios, interactions, and behaviours.

3.4 Targeted User

The targeted users would generally be individuals who are experiencing difficulties falling asleep or staying asleep. These users may be suffering from chronic insomnia or occasional sleep disturbances. The application can be designed to cater to a wide range of users, including those who prefer natural remedies, those who are interested in cognitive-behavioural therapy for insomnia (CBT-I), or those who simply want to improve their sleep hygiene.

The application can be beneficial for various groups, including:

1. Individuals with chronic insomnia: People who consistently struggle with sleep and have difficulty falling asleep or staying asleep over an extended period.

2. Individuals with occasional sleep disturbances: People who experience intermittent sleep problems due to factors such as stress, travel, or changes in routine.

3. Individuals seeking non-medication alternatives: Users who prefer natural remedies or nonpharmacological interventions to address their sleep issues.

4. Individuals interested in faster communication with doctors at any time

5. Individuals looking to improve sleep hygiene: Users who want to adopt healthy sleep habits and improve their overall sleep hygiene to optimize their sleep quality.

It is important to note that insomnia applications should not replace professional medical advice. In severe cases of chronic insomnia or if the sleep problems persist, it is recommended to consult a

healthcare professional for a proper diagnosis and treatment plan, we just trying to give user good sleep experience.

3.5 Function Requirements Specification

Patient functional requirement

- The system allows patient login by using basic information about him, like as name, age, email, governorate and Marital status.
- The system allows to all patient to make a record which contains all the previous history of him. at any time, patient can download this record.
- The system allow patient to decide with the doctor whether he works with him online or offline.
- The system allow patient to decide with the doctor whether he will communicate with him by chat or voice record
- The system allows to patient to pay by many method (credit, cash, Fawry).
- The patient is the one who will choose the doctor who will communicate with him from among the group of doctors of the system he has nominated.

- Each patient in the system has a degree of progress in the treatment and this appear on his account.
- After finishing the consultations, the system allows the patient to make feedback to provide the impression about the doctor and his / her ways of treatments
- The system allow patient to make rate for the doctor and this rate is update.

doctor functional requirement:

- The system allows the doctor to make a personal profile and upload His cv and certification.
- Doctor use the record of patient to make first impression about patient's problem.
- doctor give the patient instructions about his medicine.
- The system allow doctor to decide what the patient need in his treatment, lonely session or with group.
- If the doctor is not available, the doctor asks the system to make him offline.

Library functional requirement:

- The system allows to the patient to access to the library in any time he wants like electronic libraries.
- First The patient registers with his email
- The system allow patient can access easily to the library and choose any books that he needs to solve his problem.
- it makes it is easy for the patient so the doctor will recommend some books for him.
- After the patient reads the books, he can give a background on the idea of the library and rate this book and our services.

3.6 Non-functional Requirements

Security:

- The chat is between the patient and the doctor only, and it is not possible to take a screenshot of the chat.
- As soon as the patient gets out of the application, the system directly scans the patient's chat.
- The system must allow users (patient or doctor) to log into their account by entering their email and password or with their Google accounts.
- Encryption the information of the pay method for patient
- The system must allow users (patient or doctor) to reset their password by clicking on "I forgot my password" and receiving a link to their verified email address.

Scalability:

- The system will have doctors from different governorates and cover huge amount of patient.

Usability:

- The system provides clear features that make it easy for the patient.
- The system uses simple GUI that allow patient perform all available feature (like as registration of doctor, patient, receptionist).

Maintainability:

- The system searches for the error if it is present in the data base or the code. and when the system is updated, maintenance is done.

Availability:

- The system determines the doctors available at this time and allows the patient to enter it through chat or zoom and then the doctor determines the appropriate date for the patient to continue his treatment.

Interoperability:

- There must be interaction between the patient and the bank system, Fawry. Recovery:
 - the system can be recovered to previous state in case of hardware damage to easily allow patient and other user to continue their work.

3.7 user requirement

1. User Registration and Profile Management:

- Users should be able to create an account easily and quickly.
- The registration process should require minimal personal information.

2. Intuitive User Interface:

- The application should have a user-friendly interface that is easy to navigate.

3. Compatibility and Accessibility:

- The application should be compatible with various devices, including smartphones, tablets, and web browsers.

4. Data Privacy and Security:

- Users' personal data and sleep-related information should be protected and kept confidential.
- The application should adhere to data protection regulations and provide transparent information about data collection and usage.

3.8 Business Requirements

1. Market Differentiation:

- The application should offer unique features and capabilities that distinguish it from competitors.
- It should provide innovative solutions and approaches to addressing sleep-related issues.
- The application should aim to capture a significant market share by providing a compelling value proposition.

2. User Base Expansion:

-The application should target a wide audience, including individuals with varying sleep concerns and demographics.

-It should appeal to both individuals already diagnosed with insomnia and those seeking to improve their sleep quality.

3. Data Analytics and Research:

-The application should collect and analyse sleep data to gain insights into user behaviour and preferences.

-The application should comply with relevant data protection and privacy regulations when handling user data.

4. Brand Reputation and Trust:

-The application should prioritize user data security and privacy to build trust with users.

-It should maintain a reliable and stable platform, minimizing downtime and technical issues.

5. Scalability and Performance:

-The application should be designed to handle a growing user base without compromising performance.

-Scalability considerations should be taken into account to accommodate increased traffic and data storage needs.

6. Continuous Improvement:

-The application should have a plan for continuous improvement, including regular updates and feature enhancements.

-User feedback and analytics should be used to identify areas for improvement and address any shortcomings.

7. Business Sustainability:

-The application should be designed for long-term sustainability and growth.

-It should have a viable business model that ensures profitability and allows for ongoing support and development.

3.9 Technical Requirements

1. Platform Compatibility:

-The application should be compatible with multiple platforms, including iOS, Android

-It should be optimized for different screen sizes and resolutions to provide a consistent user experience across devices.

2. Performance:

-The application should be responsive and provide quick loading times to ensure a seamless user experience.

-It should be able to handle a large number of concurrent users without significant performance degradation.

3. Data Storage and Management:

-The application should have a robust and scalable data storage solution to handle the increasing volume of sleep data.

-It should ensure data integrity, security, and privacy in accordance with relevant regulations and compliance requirements.

4. User Authentication and Security:

- The application should implement secure user authentication mechanisms to protect user accounts and personal information.

5. Scalability and High Availability:

- The application should be designed to scale horizontally to handle increased user demand and growing data volumes.

6. Data Backup and Disaster Recovery:

-The application should have regular data backup procedures to prevent data loss.

-It should implement disaster recovery strategies to minimize downtime and ensure data availability in case of system failures or disasters

3.7 Summary

The insomnia application is a user-friendly tool designed to assist individuals in managing and improving their sleep patterns. It offers features such as personalized recommendations, and guidance on optimizing the sleep environment. The application focuses on non-functional requirements such as performance, usability, security, and reliability to ensure a positive user experience.

Overall, the insomnia application aims to provide insights and resources to help users effectively address insomnia and achieve better sleep quality.

Chapter four
SYSTEM DESIGN

4.1 Introduction

Sleep is a critical aspect of human health, allowing the body and mind to rest, repair, and rejuvenate. However, millions of people worldwide struggle with sleep insomnia, a condition characterized by difficulty falling asleep, staying asleep, or both. Insomnia can have a significant impact on overall well-being, leading to daytime fatigue, irritability, impaired cognitive function, and an increased risk of chronic health conditions.

The causes of sleep insomnia are complex and multifaceted, ranging from stress and anxiety to medical conditions and environmental factors. Addressing insomnia often requires a multi-faceted approach, including lifestyle modifications, cognitive-behavioural therapy, and, in some cases, medication. With proper diagnosis, treatment, and self-care strategies, individuals with insomnia can improve their sleep quality and overall health.

The design system includes effective solutions to improve sleep quality, as it contains a simple and clear user interface that allows users to easily access the various features in the application

4.2 Context diagram

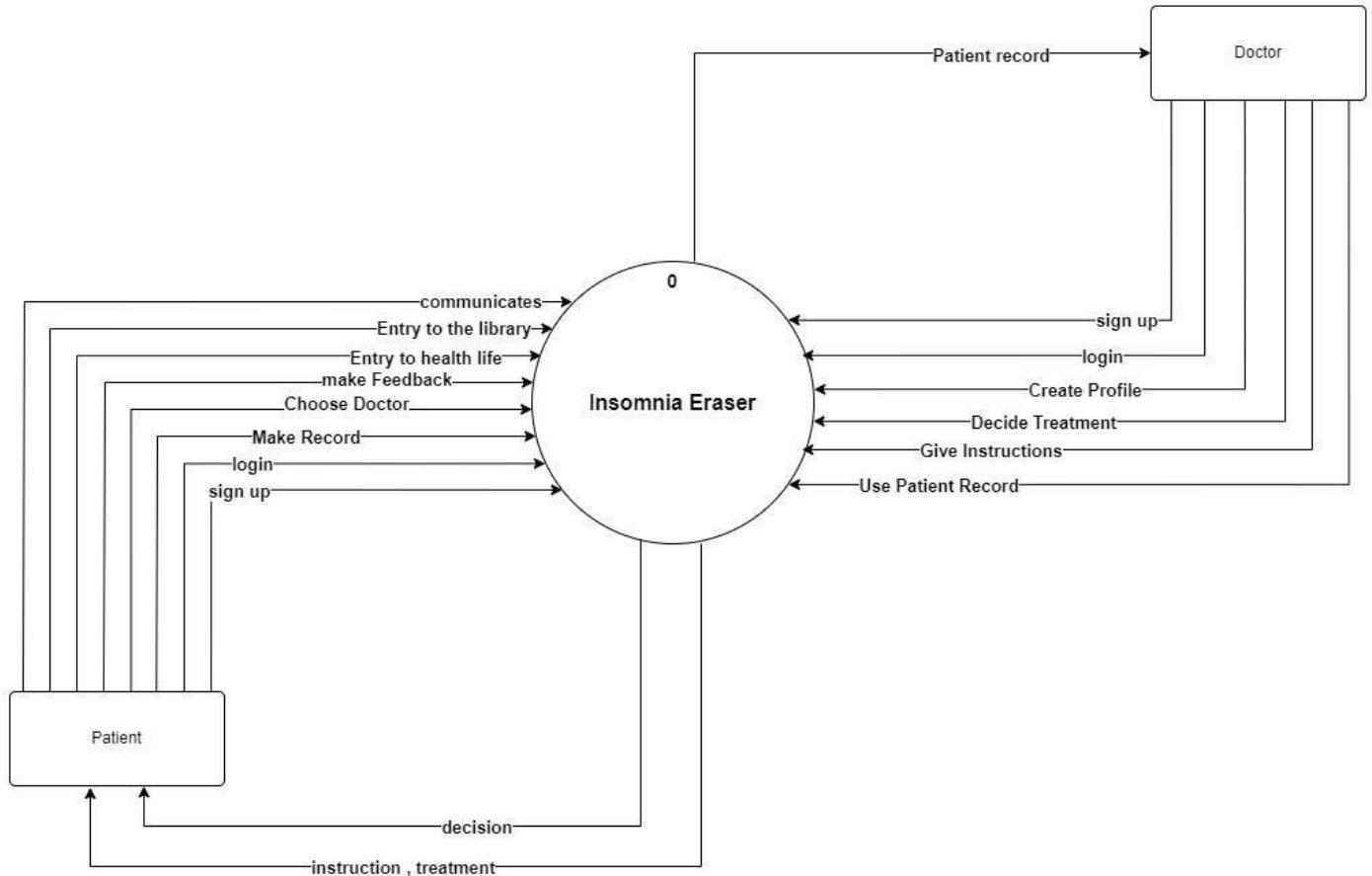


Figure 4.1 Context diagram

4.3 Data flow diagram

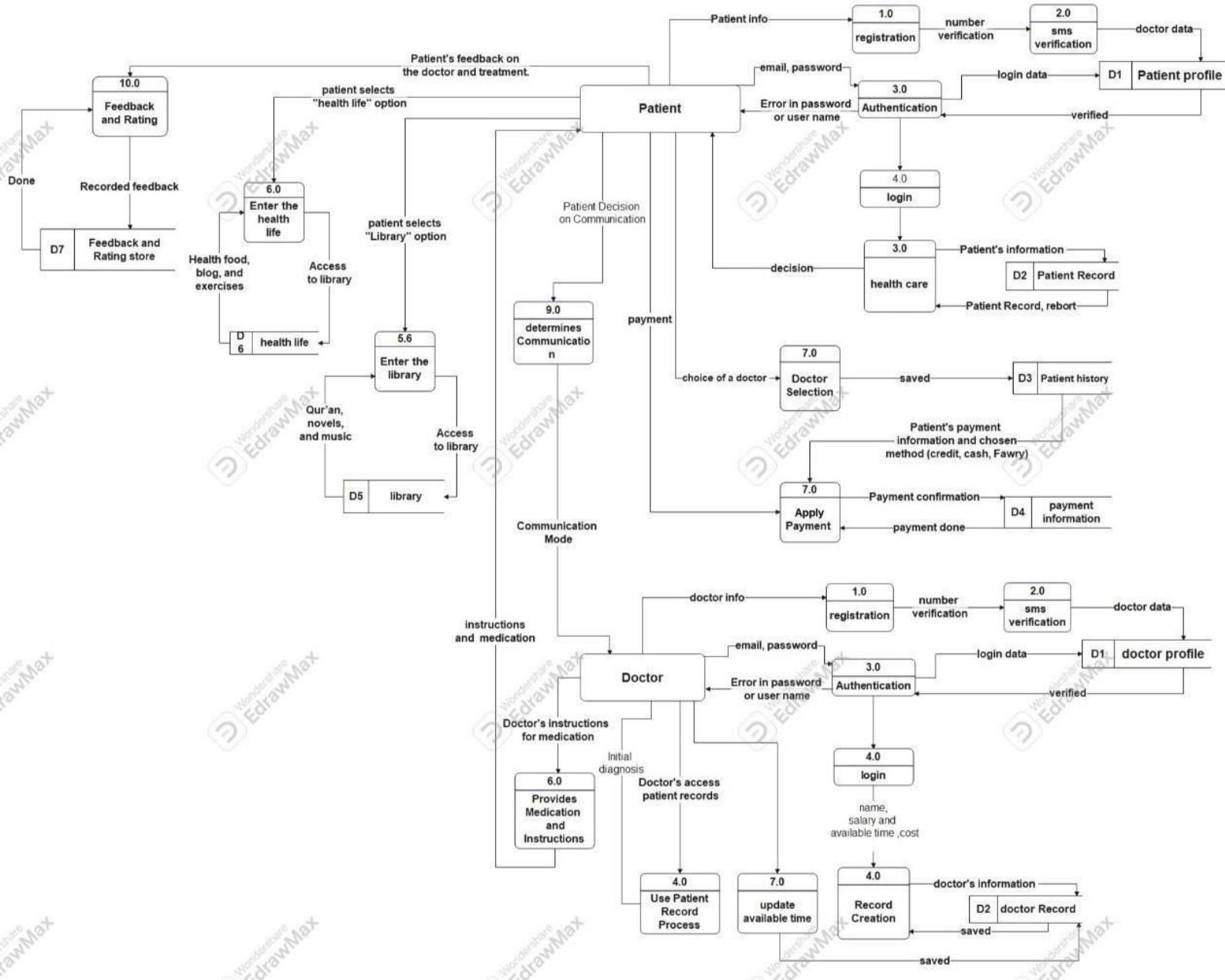


Figure 4.2 data flow diagram

4.4 Entity relationship diagram

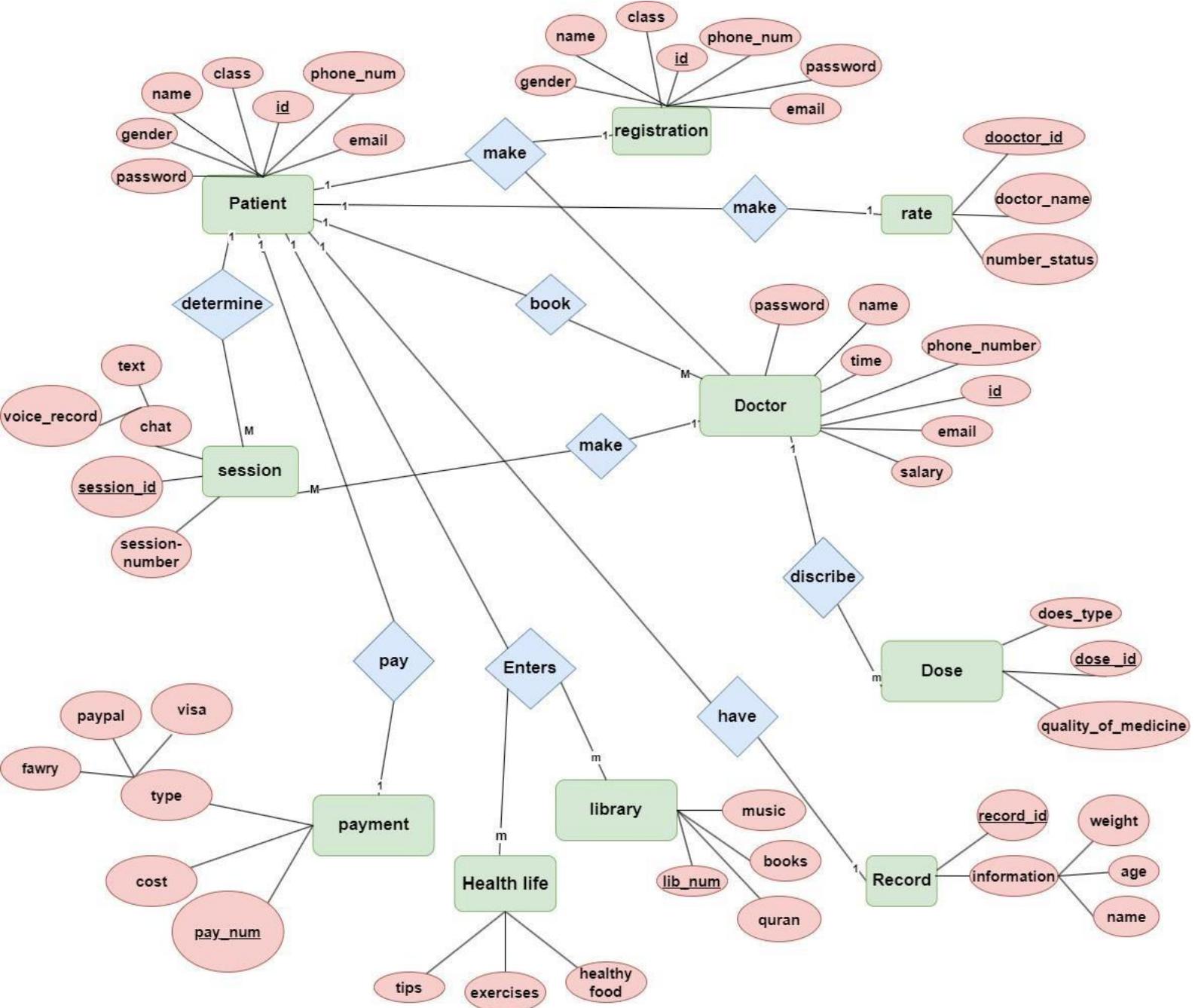


Figure 4.3 ERD diagram

4.5 UML use case diagram

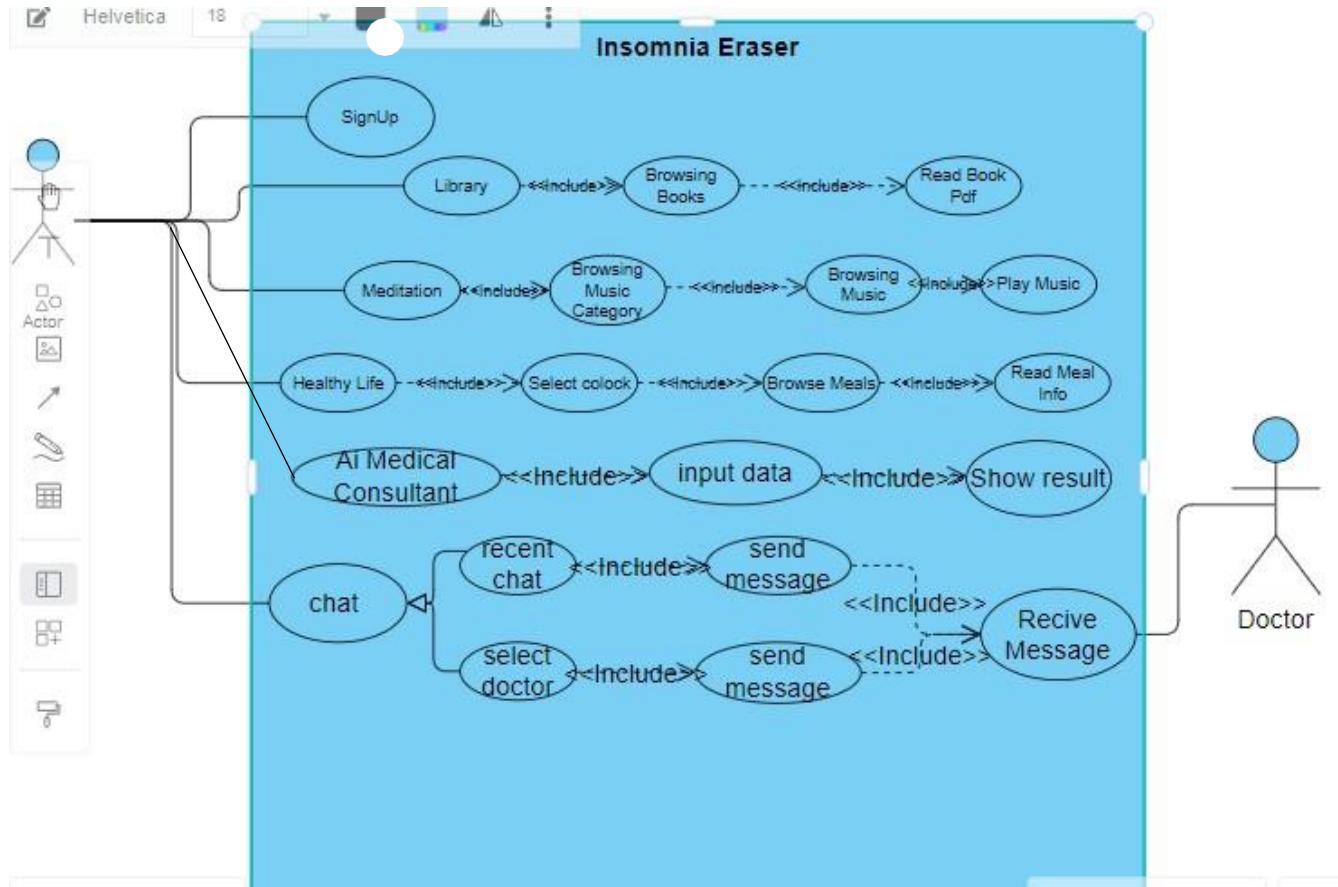
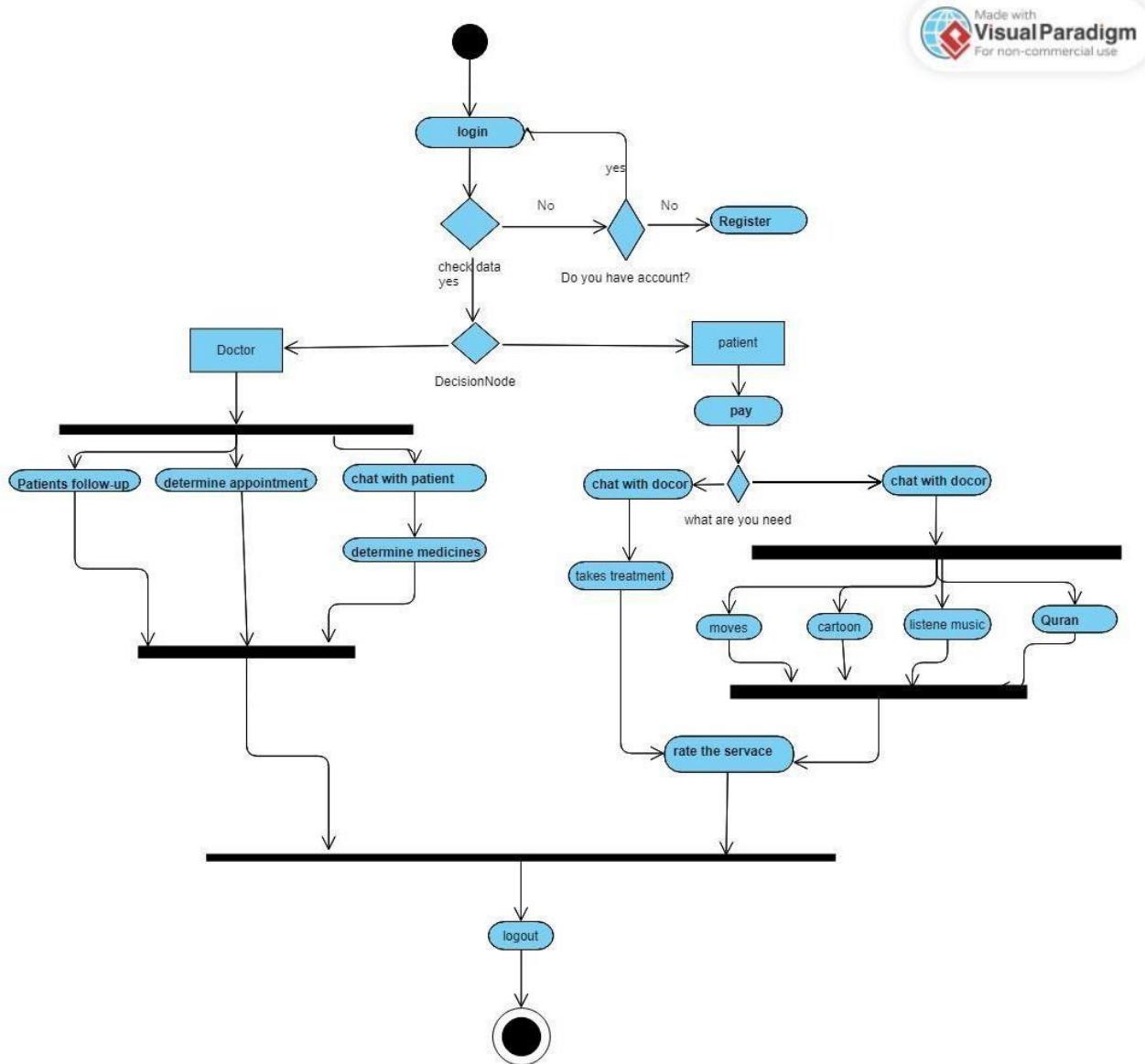
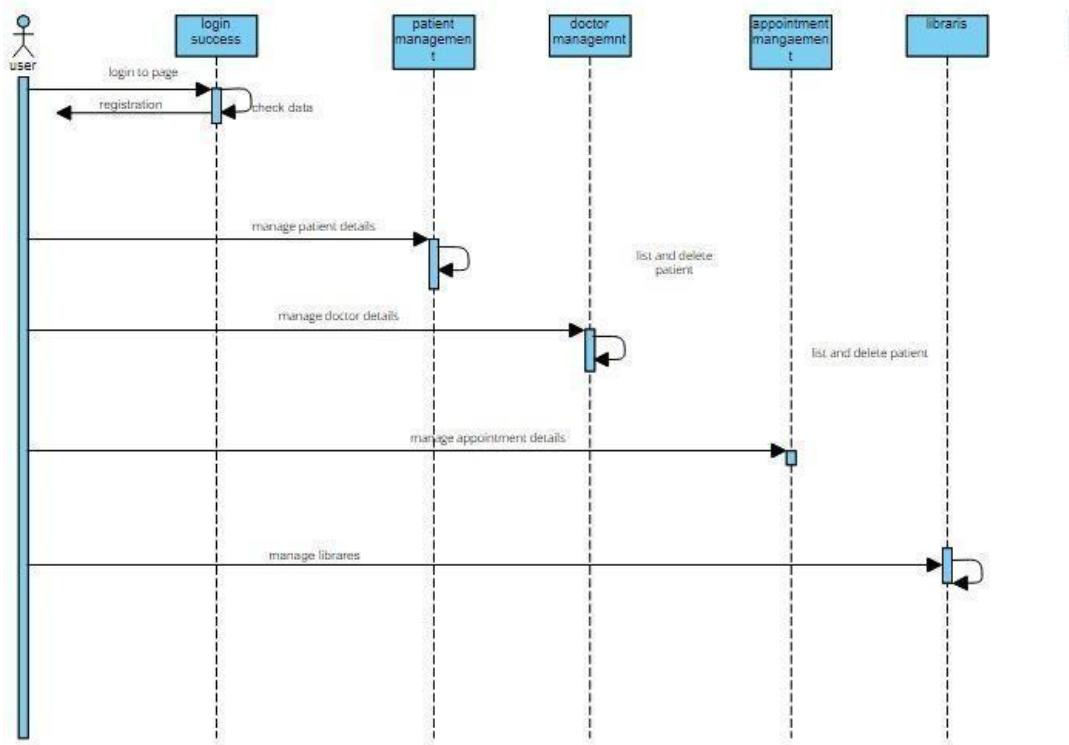


Figure 4.4 use case diagram

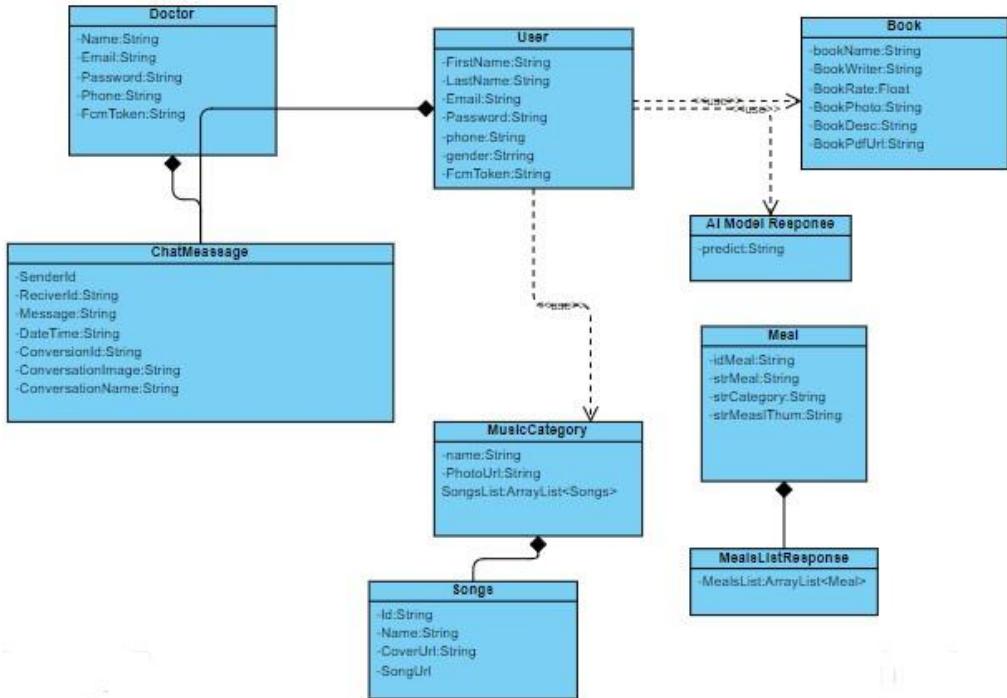
4.6 UML Activity diagram



4.7 UML Sequence diagram



4.8 class diagram



4.7 Summary

Absolutely, a well-crafted design system plays a crucial role in ensuring a cohesive and userfriendly experience. It provides consistency, streamlines development, and enhances usability across the entire application. The design system is one of the most important parts of any system because of its importance in the user interface and clarifying the steps of the application, so it must be as accurate as possible.

Chapter Five

Mobile Application

Implementation

4.1. Kotlin

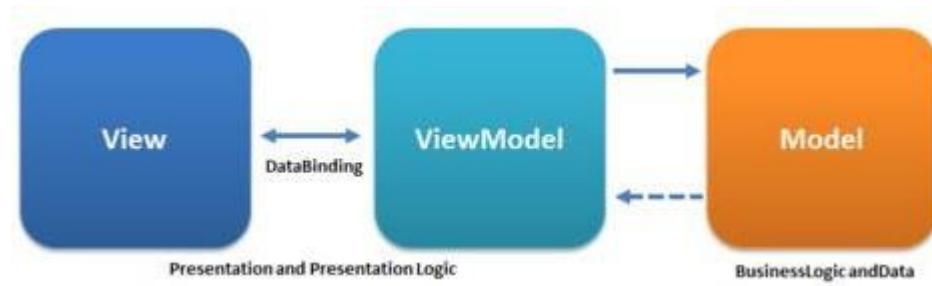


Kotlin is a modern, versatile programming language developed by JetBrains. It became an official language for Android app development by Google in 2017 and offers several features that make it a popular choice among developers. Here's a quick overview:

- Ease of Use: Kotlin is designed to be simple and easy to read, making it ideal for both new and experienced developers.
- Full Java Interoperability: Kotlin can be used with an existing Java project without any issues, allowing both languages to work together seamlessly in the same project.
- Type Safety: Kotlin ensures code stability by reducing errors caused by null values and incorrect type definitions.
- Concise and Clear Functions: Kotlin allows for writing concise and clear functions, reducing the amount of code needed and increasing productivity.
- Strong Tooling Support: Kotlin has robust support for development tools like IntelliJ IDEA and Android Studio, making programming and debugging easier.

Kotlin is used for developing Android applications, server-side applications, desktop applications, and even web applications. It is a flexible and modern language, making it a strong choice for contemporary programming projects.

4.2. Architectural Pattern (MVVM)



The MVVM (Model-View-ViewModel) pattern is an architectural design pattern used in Android development to improve separation of concerns, making code more maintainable and testable. It is a natural evolution of the traditional MVC pattern and offers several key improvements. Here's a quick overview of MVVM:

- **Model:** This layer contains the business logic and data source. The model defines how data is managed, whether it comes from a local database or a remote server.
- **View:** This is the layer responsible for the user interface (UI). In Android, this includes Activities and Fragments. The view should be lightweight and only responsible for displaying data provided by the ViewModel.
- **ViewModel:** This acts as a bridge between the View and the Model. It handles business logic and interactions between the View and the Model. The ViewModel holds LiveData, which can be observed by the View, allowing the UI to update automatically when data changes.

Benefits of Using MVVM:

- **Clear Separation of Concerns:** Dividing the code into distinct units (Model, View, ViewModel) makes it easier to understand and maintain.
- **Code Reusability:** Components can be reused more easily due to the clear separation of responsibilities.
- **Testability:** The MVVM pattern makes unit testing easier since the ViewModel can be tested independently of the UI.
- **Automatic UI Updates:** Using LiveData or Observable, the View can update itself automatically when the data in the ViewModel changes, reducing the need for manual UI updates.

4.3. Firebase



Firebase is a comprehensive app development platform developed by Google that provides a suite of cloud-based tools and services to help developers build, improve, and grow their applications. Here's a quick overview of its core features and benefits:

Core Features:

- **Realtime Database:** A NoSQL cloud database that allows data to be stored and synchronized in real-time across all clients.
- **Cloud Firestore:** A flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform.
- **Authentication:** Easy-to-use authentication services that support email and password login, social media login, and more.
- **Cloud Storage:** Secure file uploads and downloads for user-generated content like photos and videos.
- **Cloud Functions:** Serverless functions that let you run backend code in response to events triggered by Firebase features and HTTPS requests.
- **Analytics:** Detailed analytics to track user behavior and measure the effectiveness of your app's features.
- **Cloud Messaging:** A robust solution for sending push notifications and messages to engage your users.
- **Remote Config:** Allows you to dynamically change the behavior and appearance of your app without publishing an app update.
- **Crashlytics:** Helps you track, prioritize, and fix stability issues in real-time to improve app quality.

Benefits:

- **Ease of Integration:** Firebase integrates seamlessly with other Google services and can be easily added to your existing applications.
- **Scalability:** Designed to scale with your application, from small startups to large enterprises.
- **Cross-Platform Support:** Provides tools and SDKs for iOS, Android, web, and more, making it versatile for different platforms.
- **Time-Saving:** Simplifies backend development, allowing developers to focus more on building front-end features and user experience.
- **Community and Support:** Strong community support and extensive documentation help developers resolve issues and implement features quickly.

Firebase is widely used in the industry for its reliability, ease of use, and extensive feature set, making it a preferred choice for many developers building both small and large-scale applications.

Firebase Auth

Authentication

Users Sign-in method Templates Usage Settings Extensions

Search by email address, phone number, or user UID					Add user	⋮
Identifier	Providers	Created ↓	Signed In	User UID		
emanmohamed@yahoo...	✉️	Jun 9, 2024		9ftAliRI5EfQge9GWpZIvqyKdG...		
mahmoudhegazy@gmail...	✉️	Jun 9, 2024		X6TVvOXF2qU39PjFtvalUqm...		
					Rows per page:	50 ▾ 1 – 2 of 2 ⏪ ⏩

Firestore Database

Doctors > WR94Hzn0n9hk.			More in Google Cloud
(default)	Doctors	WR94Hzn0n9hkJX6H0mM9sMHshmr1	⋮
+ Start collection	+ Add document	+ Start collection	
Doctors	WR94Hzn0n9hkJX6H0mM9sMHshmr1	+ Add field	
Library		email: "ahmedmohamed@gmail.com"	(string) 🖊️
category		fcmToken: "fv9PZ46tQiujJfgd_zR40K:APA91bHkNfHhiAN1s7lp6lfcUTC-Gergueo-Wnsc3z_rPDZYoyAtkaNBTSyUSSzXt10MuW_1xjEIL7_w2sL0KrH0e7PlJqq6RHsShj1AP6XNEIWlh7p-lk-u4N7fNSaWB_ntz0816jN"	
chat		id: "WR94Hzn0n9hkJX6H0mM9sMHshmr1"	
conversations		image: "https://firebasestorage.googleapis.com/v0/b/ensomina-earse.appspot.com/o/doctor.1717421607026.jpg?alt=media&token=b796b555-7d1b-4f6c-9929-d259b7427a87"	
patients		name: "Ahmed Mohamed"	
songs		password: "123456789"	
		phoneNumber: "01018941581"	
			Activate Windows Go to Settings to activate Windows.

chat > ORFxAMY1LiEyJ...			More in G
(default)	chat	ORFxAMY1LiEyJGgH3toT	⋮
+ Start collection	+ Add document	+ Start collection	
Doctors	8bY1jmT7WyKlojBUpbf	+ Add field	
Library	BH70dDD0377DzaNuWzCf	message: "Hello "	
category	DCFeuLDyZjvWiKnKULLd	receiverId: "WR94Hzn0n9hkJX6H0mM9sMHshmr1"	
chat	ORFxAMY1LiEyJGgH3toT	senderId: "YJmf3rV870XRLKBH2fyACLxD3n1"	
conversations	WI0DTv0IcEuXspmcxv8a	timestamp: June 5, 2024 at 12:45:18 PM UTC+3	
patients	XuHnhxjro6VRYsnKU2Cp		
songs	gj1nuIUjkCB7gWZrMkwE		
	hhDfBukSwsGDx8Qg7Cbw		
	jN6Kyn8DaCC4n17JX0xi		
	mGFqBYhevACMd7Ayk7oF		
	ta4mMTqkYtvwRoNOVU4j		
	xaFvDL67ax9dEndM18Gv		
	zxZNWuwflz16kAatJB5R		

How Data is Stored in Firestore?

Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform. It stores data in the form of documents and collections. Here's a breakdown of how data is stored and structured in Firestore:

Structure:

1. **Documents:** The basic unit of storage in Firestore. Each document is a set of key-value pairs, where each key must be a string, and the value can be a variety of data types, such as strings, numbers, booleans, dates, arrays, and nested objects. Documents are identified by a unique ID.
2. **Collections:** Groups of documents. Collections do not store actual data; they act as containers for documents. A collection can have multiple documents, and a document can reference multiple subcollections.
3. **Subcollections:** Collections nested within documents. This allows for a hierarchical data structure.

Firebase Storage

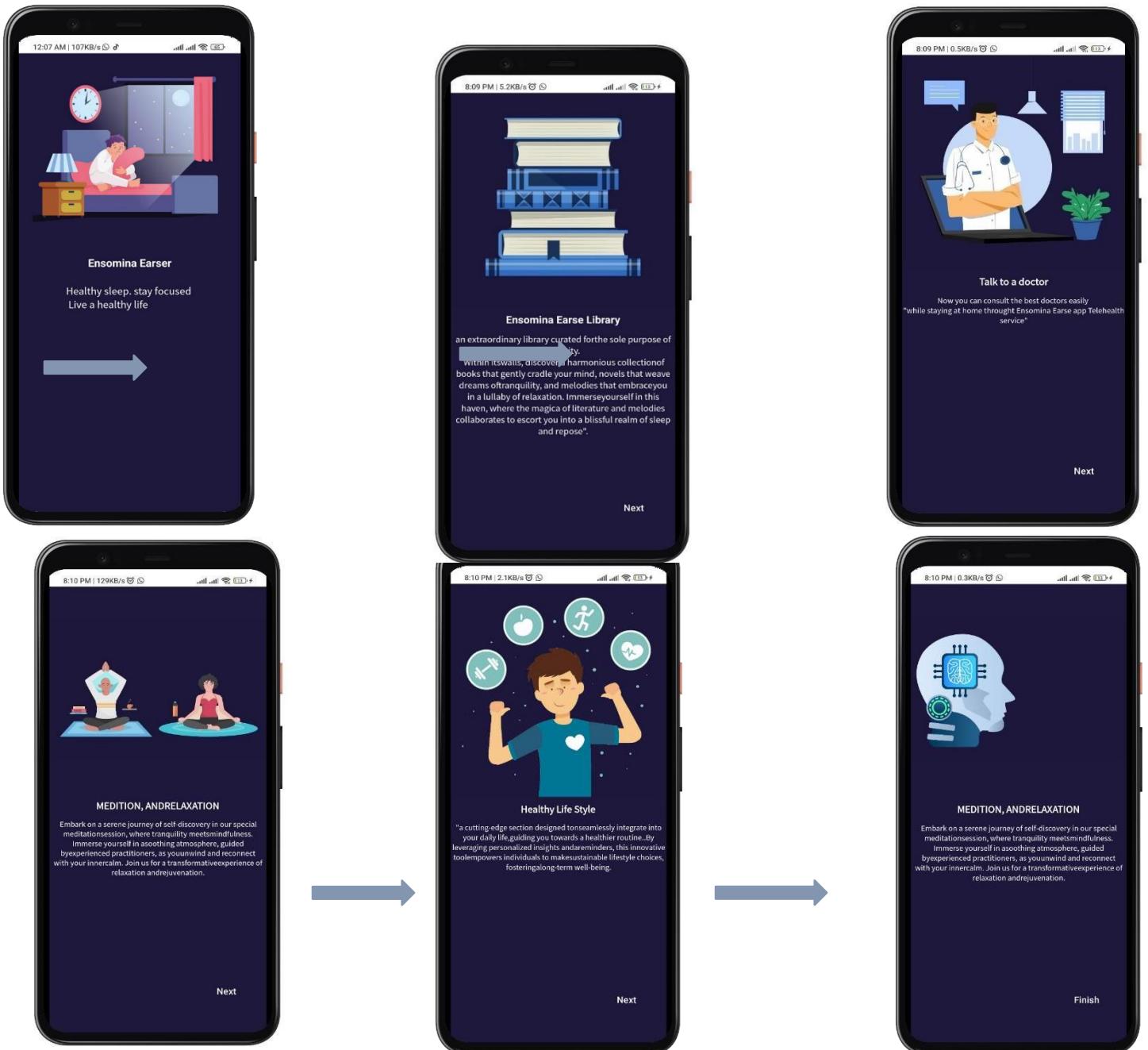
gs://ensomina-earse.appspot.com			Up
	Name	Size	Type
<input type="checkbox"/>	category_images/	—	Folder
<input type="checkbox"/>	songs/	—	Folder
<input type="checkbox"/>	songs_cover/	—	Folder
<input type="checkbox"/>	Collected Essays.jpg	19.02 KB	image/jpeg
<input type="checkbox"/>	Greening IT.pdf	1.41 MB	application/pdf
<input type="checkbox"/>	Greening.ico	22.61 KB	image/icon

Firebase Storage is a cloud-based storage solution provided by Google's Firebase platform. It is designed to store and serve user-generated content such as photos, videos, and other binary files. Firebase Storage is built on Google Cloud Storage, offering robust and scalable storage capabilities with strong security and integration with Firebase service

5.4 Splash Screen & Storyboard

A splash screen is a graphical control element consisting of a window containing an image, a logo, or any other visual element that appears when an application is launching. It typically serves two main purposes: providing an initial impression and masking the loading process of the app

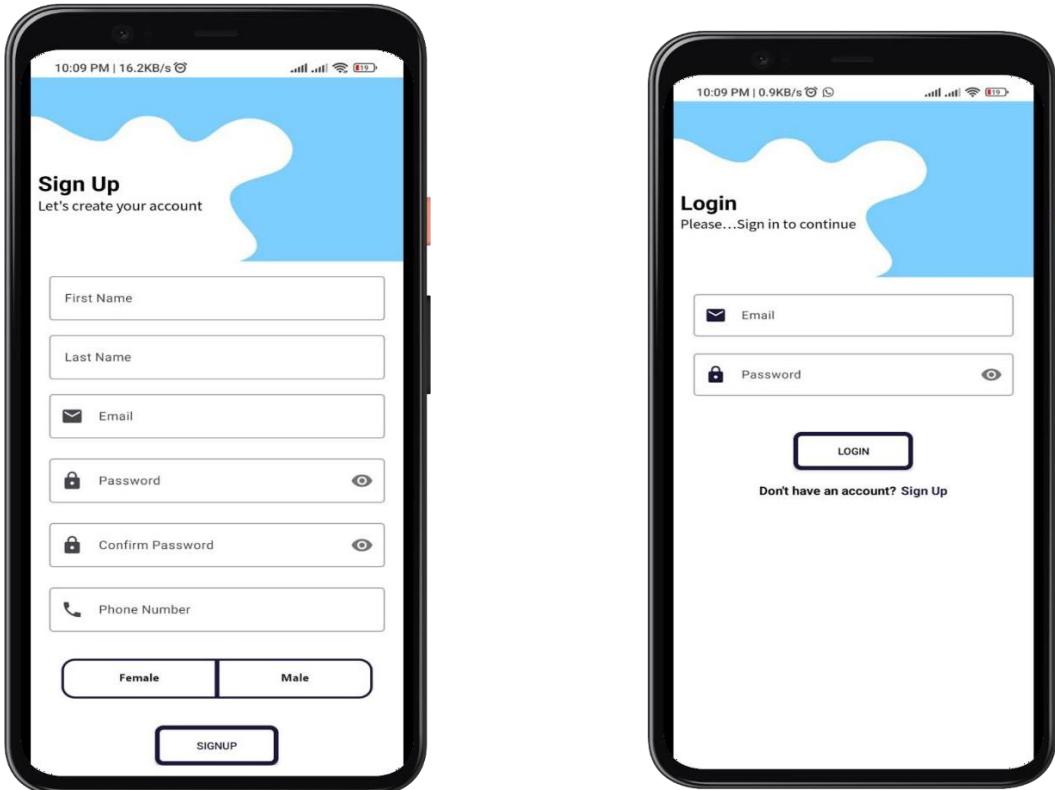
Storyboard: It is several screens that show the features of the application



Storyboard appears for the first time only when the user installs the application to be aware of the features in the application Every time the user opens the application, a Splash Screen screen appears to him We check whether the user has logged in or not, and if he has logged in, we go to the main page. If he has not logged in, he goes to the login page.

```
private fun checkOnBoardingState() {
    CoroutineScope(Dispatchers.Main).launch {
        delay(3000L)
        if
            (preference.getBoolean(Constants.ONBOARDING_STATE) && preference.getBoolean("Login"))
            {
                findNavController().navigate(R.id.action_splashFragment_to_homeFragment)
            }
            else if (preference.getBoolean(Constants.ONBOARDING_STATE) &&
!preference.getBoolean("Login")){
                findNavController().navigate(R.id.action_splashFragment_to_loginFragment)
            }
            else
                findNavController().navigate(R.id.action_splashFragment_to_viewPagerFragment)
        }
    }
}
```

5.5. Authentication Module Screens



5.6 Home Screen

The Home Page of our Android application serves as the central hub for all the features designed to help users combat insomnia and improve their overall well-being. The Home Page is intuitively organized into five main sections, each providing a unique service:

1. Library of Books:
 - Description: This section offers a curated collection of books specifically focused on treating insomnia. Users can browse through a variety of titles that provide insights, techniques, and therapies to improve sleep quality.
 - Features: Includes book summaries, user reviews, and recommendations based on user preferences.
2. Library of Songs for Relaxation:
 - Description: Here, users can find a wide selection of soothing music and soundscapes designed to promote relaxation and aid in falling asleep. The collection includes nature sounds, calming instrumental tracks, and guided sleep meditations.
 - Features: Users can create playlists, mark favorites, and set timers for playback.
3. AI Consultation for Insomnia Diagnosis:
 - Description: This innovative section utilizes artificial intelligence to assess whether the user is suffering from insomnia. By answering a series of questions, users receive a personalized evaluation and recommendations based on their responses.
 - Features: Interactive questionnaire, instant feedback, and personalized advice.
4. Healthy Foods:
 - Description: This section provides information on healthy foods that can help improve sleep quality and overall health. Users can explore recipes, dietary tips, and nutritional information specifically tailored to combat insomnia.
 - Features: Recipe suggestions, meal planning guides, and health tips.
5. Chat with a Doctor:
 - Description: Users have the opportunity to chat directly with a medical professional through this section. This service offers personalized medical advice, answers to health-related questions, and professional guidance for managing insomnia.
 - Features: Real-time chat, appointment scheduling, and access to medical resources.

Each section is designed to offer a comprehensive approach to managing and overcoming insomnia, providing users with the resources and support they need to achieve better sleep and improve their quality of life.



```
private fun setListeners(){
    binding.libraryBtn.setOnClickListener { it: View!
        findNavController().navigate(R.id.action_homeFragment_to_libraryFragment)
    }

    binding.cvMeditation.setOnClickListener { it: View!
        findNavController().navigate(R.id.action_homeFragment_to_meditationFragment)
    }

    binding.cvMedicalConsultant.setOnClickListener { it: View!
        findNavController().navigate(R.id.action_homeFragment_to_medicalConsultantFragment)
    }

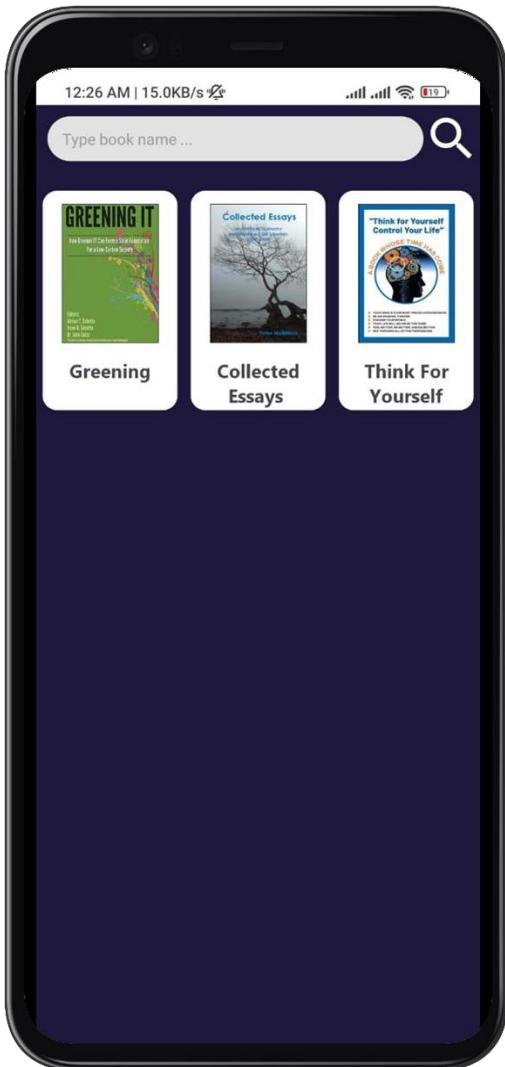
    binding.cvHealthyFood.setOnClickListener { it: View!
        findNavController().navigate(R.id.action_homeFragment_to_healthyFoodFragment)
    }

    binding.cvChat.setOnClickListener { it: View!
        findNavController().navigate(R.id.action_homeFragment_to_mainChatFragment)
    }
}
```

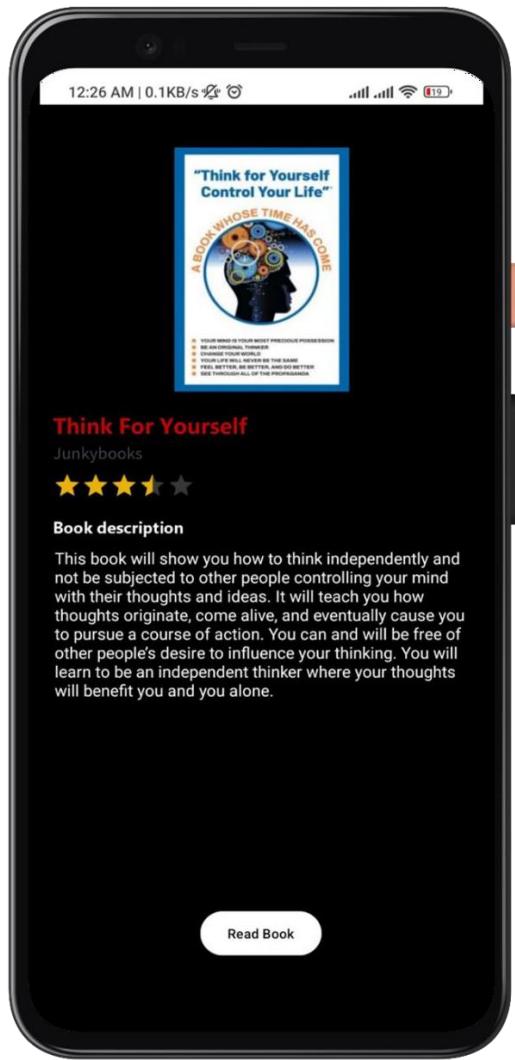
This method, `setListeners`, sets up navigation for five different Screens on the home screen.

5.7. Books Library Screen

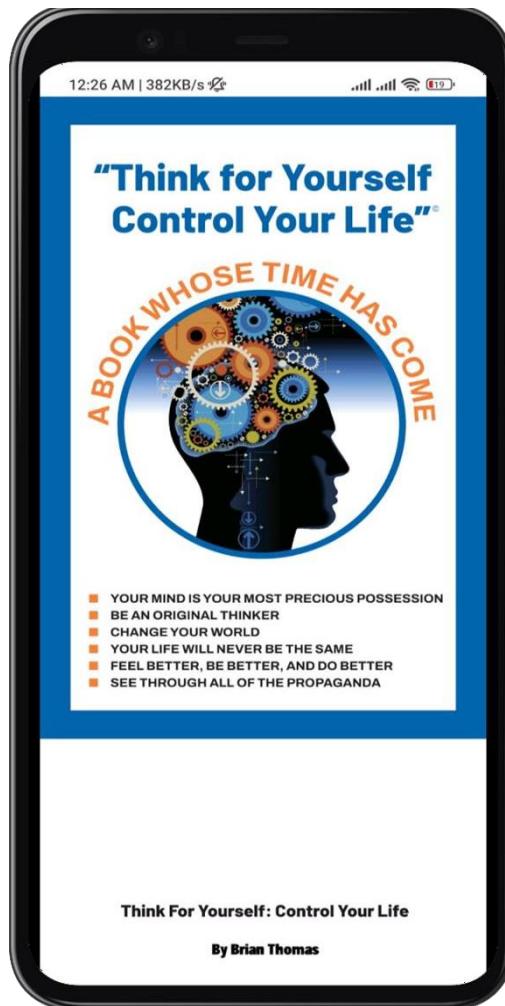
Books Screen



Book Details Screen



Pdf Viewer



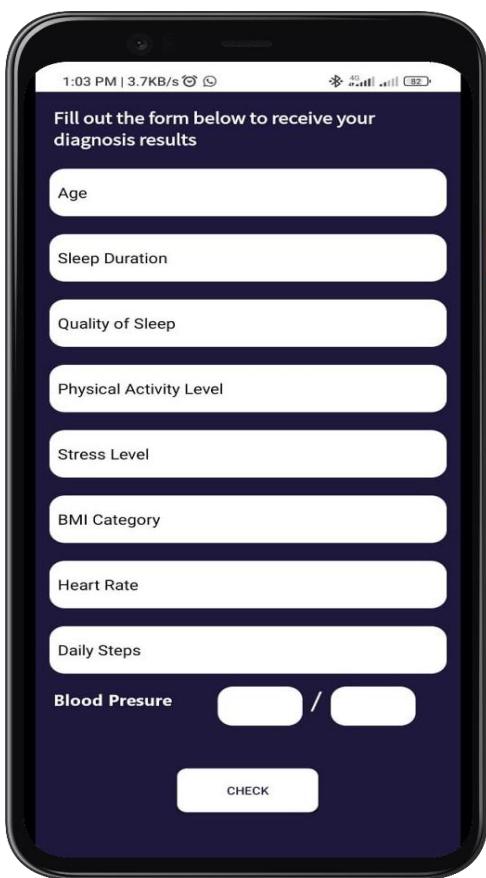
In this section, books are displayed through Firebase, and when we click on the book, we go to the book's details page, then click on Read, and the book is downloaded to the phone, then the book opens in the PDF viewer.

```
private suspend fun getBooksFromFireStore() {  
    binding.bookRv.apply {  
        adapter = booksAdapter  
        layoutManager = GridLayoutManager(requireContext(), 3, GridLayoutManager.VERTICAL, false)  
    }  
  
    viewModel.getLibraryBooks().collect{  
        if (it.isNotEmpty()) {  
            dataState()  
            booksAdapter.diff.submitList(it)  
        }  
    }  
}
```

5.8. Ai Medical Consultant

- Collect User Data

- Result Dialog



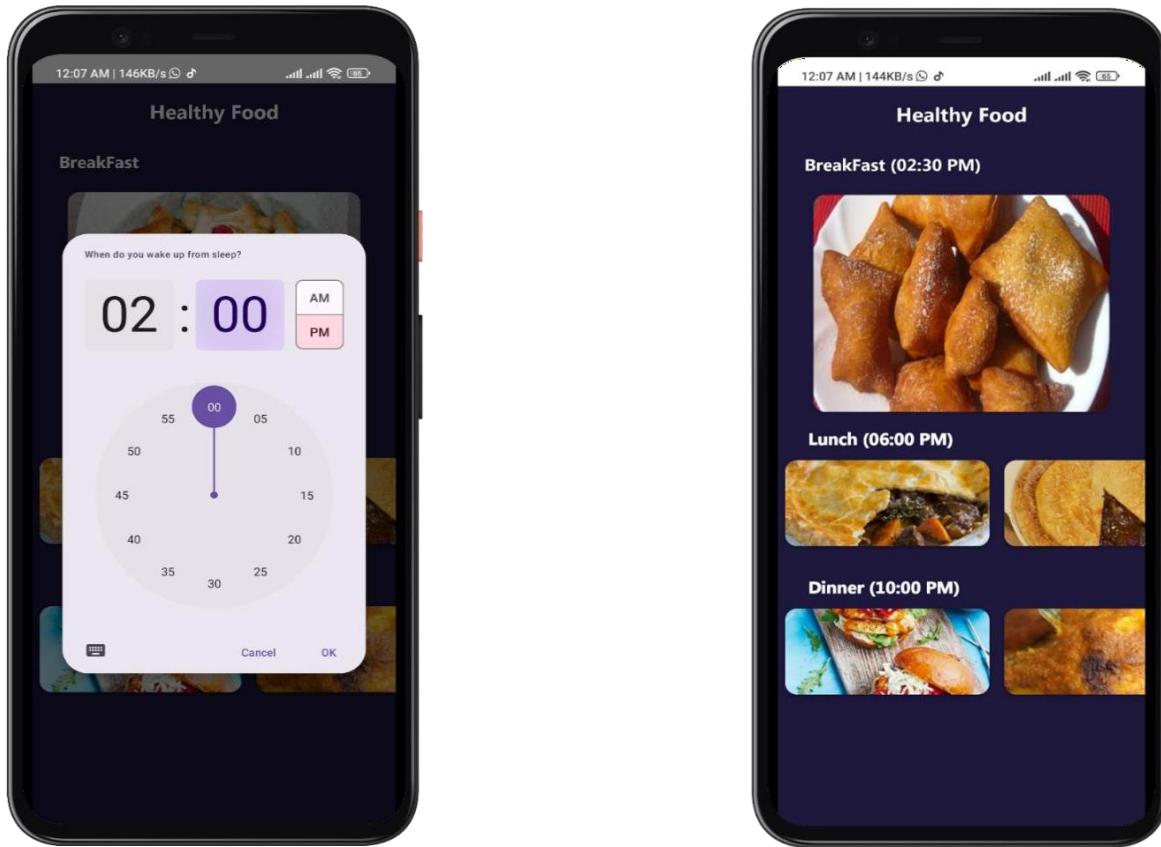
On this screen, the user writes some of his data, then the data is sent to an artificial intelligence model, and he is diagnosed with whether he suffers from insomnia or not

```
val requestBody: RequestBody = MultipartBody.Builder()
    .setType(MultipartBody.FORM)
    .addFormDataPart("Age", binding.ageTv.text.toString())
    .addFormDataPart("Sleep Duration", binding.SleepDurationTv.text.toString())
    .addFormDataPart("Quality of Sleep", binding.qualityOfSleepTv.text.toString())
    .addFormDataPart("Physical Activity Level", binding.activityTv.text.toString())
    .addFormDataPart("Stress Level", binding.stressLevelTv.text.toString())
    .addFormDataPart("BMI Category", binding.bmiTv.text.toString())
    .addFormDataPart("Heart Rate", binding.heartRateTv.text.toString())
    .addFormDataPart("Daily Steps", binding.dailyStepsTv.text.toString())
    .addFormDataPart("systolic_bp", binding.systolicBpEt.text.toString())
    .addFormDataPart("diastolic_bp", binding.diastolicBpEt.text.toString())
    .build()

RetrofitInstance.api.getResult(requestBody).enqueue(object : Callback<AiModelResponse> {
    override fun onResponse(p0: Call<AiModelResponse>, p1: Response<AiModelResponse>) {
        if (p1.isSuccessful) {
            binding.checkBtn.visibility = View.VISIBLE
            binding.loadingGif.visibility = View.INVISIBLE
        }
    }

    override fun onFailure(call: Call<AiModelResponse>, t: Throwable) {
        binding.checkBtn.visibility = View.INVISIBLE
        binding.loadingGif.visibility = View.VISIBLE
    }
})
```

5.9 Healthy Food



This screen in our Android application is designed to help users establish a structured and healthy daily meal routine by suggesting optimal meal times based on their wake-up time. The main features of this screen include:

Wake-Up Time Selection:

- Purpose: Users are prompted to select the time they typically wake up in the morning. This input serves as the basis for generating meal times for breakfast, lunch, and dinner.
- User Interaction: The user can set their wake-up time using a time picker interface.

Suggested Meal Times:

- Breakfast Time: Based on the wake-up time, the application calculates and suggests an appropriate time for breakfast, ensuring the user starts their day with a nutritious meal.
- Lunch Time: The application also suggests a time for lunch, typically about 4-5 hours after breakfast, helping to maintain energy levels throughout the day.
- Dinner Time: Lastly, the application proposes a dinner time, usually about 5-6 hours after lunch, to ensure a balanced distribution of meals.

Meal Information and Contents:

- Meal Details: For each suggested meal time (breakfast, lunch, and dinner), the user can view detailed information about the recommended meals.
- Nutritional Information: This includes a breakdown of the nutritional content such as calories, macronutrients (proteins, fats, carbohydrates), and essential vitamins and minerals.
- Meal Contents: Users can see what the meal consists of, including specific foods and portion sizes, to help them make informed dietary choices.

```

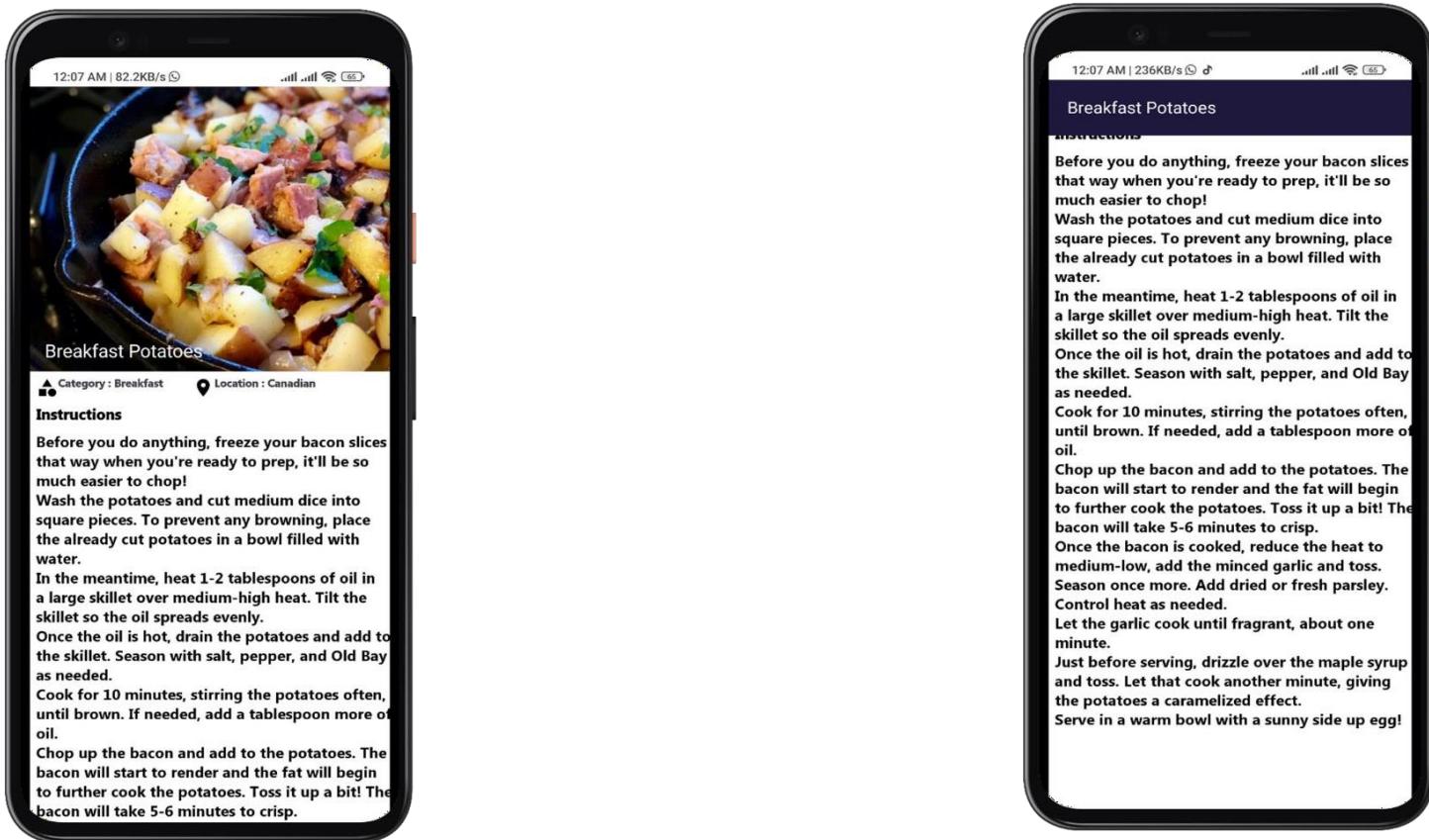
    }

    private fun getLunchMeals(){
        viewModel.getLunchMeals( category: "beef").observe(viewLifecycleOwner) { it: MealsList!
            lunchMealsAdapter.diff.submitList(it.meals)
            binding.lunchMealsRv.apply {
                this: RecyclerView
                adapter = lunchMealsAdapter
                layoutManager = LinearLayoutManager(
                    requireContext(),
                    GridLayoutManager.HORIZONTAL, reverseLayout: false
                )
            }
        }
    }
}

```

- This is a private function named `getLunchMeals` which means it can only be accessed within the class it is defined in.
- `viewModel.getLunchMeals("beef")` calls a method from the `viewModel` to fetch a list of lunch meals that include beef.
- `.observe(viewLifecycleOwner)` sets up an observer on the `LiveData` returned by `getLunchMeals`. This ensures that when the data (list of meals) is available or updated, the provided lambda function is executed.
- `viewLifecycleOwner` ensures that the observer is tied to the lifecycle of the fragment or activity, so it will be automatically cleaned up when the view is destroyed.
- Inside the observer lambda, it refers to the data object containing the list of meals.
- `lunchMealsAdapter.diff.submitList(it.meals)` submits the new list of meals (`it.meals`) to the adapter's `DiffUtil`, which handles the efficient updating of the `RecyclerView`'s items.
- `binding.lunchMealsRv` references the `RecyclerView` from the layout using view binding.
- `.apply {}` is a Kotlin scope function that allows you to configure an object (in this case, the `RecyclerView`) within its block.
- `adapter = lunchMealsAdapter` sets the adapter of the `RecyclerView` to `lunchMealsAdapter`, which manages the display of meal items.
- `layoutManager = LinearLayoutManager(...)` sets the layout manager of the `RecyclerView` to a `LinearLayoutManager`. This determines how the items in the `RecyclerView` are laid out.
- `requireContext()` provides the context required for the layout manager.
- `GridLayoutManager.HORIZONTAL` specifies that the layout manager should arrange items horizontally.

- **false** indicates that the layout should not be reversed.



```

private fun getMealDetails(id:String){
    viewModel.getMealDetails(id).observe(viewLifecycleOwner){ it: Meal?
        if (it!=null){
            binding.progressBar.visibility = View.INVISIBLE
            binding.categoryTv.text ="Category : ${it.strCategory}"
            binding.locationTv.text="Location : ${it.strArea}"
            binding.contentTv.text = it.strInstructions
            mealVideo = it.strYoutube.toString()
        }
    }
}

```

This function, `getMealDetails`, performs the following steps:

- Fetches meal details using the provided meal ID by calling the ViewModel's `getMealDetails` method.
- Observes the LiveData returned by `getMealDetails` to get notified when the meal details data is available.
- When the meal details data is available (not null), it updates the UI elements (TextViews) with the meal's category, location, and instructions.
- Hides the progress bar that indicates loading.
- Stores the YouTube video URL of the meal for potential use elsewhere in the application.

This approach ensures that the meal details are dynamically loaded and displayed, and that the UI updates are performed efficiently and correctly, leveraging LiveData and view binding.

5.10 Meditation & Relaxation

The meditation and relaxation section of our application is designed to help users achieve inner peace and deep relaxation. This section is tailored specifically for individuals who struggle with insomnia or stress, providing a variety of options to enhance their sleep quality and overall well-being.

The meditation and relaxation section consists of three main screens:

1. **Music Categories Screen:** This screen allows users to browse different categories of music specifically curated for meditation and relaxation. These categories include a variety of calming music genres and natural sounds that help create a peaceful and relaxing environment.
2. **Music Screen:** After selecting the appropriate category, users are taken to the music screen, which features a diverse collection of tracks and albums designed to provide a soothing meditation experience. Users can choose their preferred music to accompany their relaxation sessions.
3. **Player Screen:** The player screen enables users to play, pause, and navigate through their chosen tracks. This screen provides all necessary controls for managing the meditation and relaxation audio, ensuring an uninterrupted and immersive experience.

By incorporating this section into our application, we aim to offer a comprehensive tool that supports users in managing their stress and improving their sleep quality through the power of music and guided meditation.

Music Categories Screen

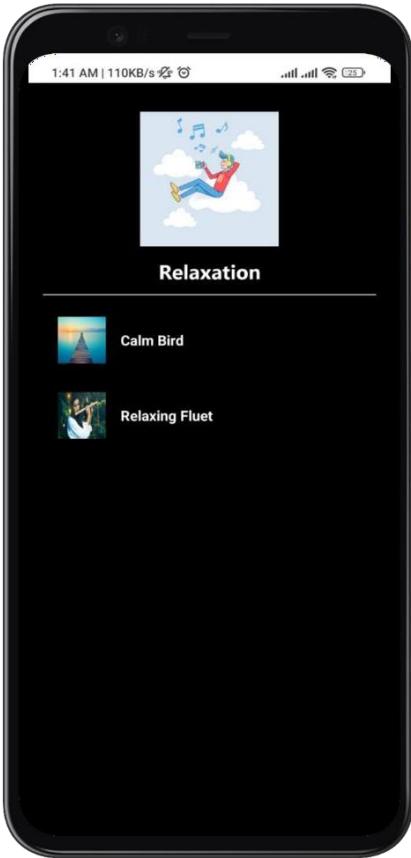


```
CoroutineScope(Dispatchers.Main).launch{ this: CoroutineScope
    delay( timeMillis: 2000L)
    viewModel.getCategories().collect{ it: ArrayList<Category>
        setupCategoryRv(it)
    }
}
```

Collecting Categories:

- `viewModel.getCategories().collect` is a Kotlin coroutine that collects data emitted by the `getCategories` function from the `viewModel`.
- `collect` is used to handle the stream of data emitted by a Flow. Here, it collects a list of categories.

Songs Screen



Player Screen



In our Android application, we have implemented a sophisticated music playback screen using ExoPlayer, a powerful media player library developed by Google. ExoPlayer provides a flexible and efficient way to handle audio and video playback within Android apps, offering more customization and features compared to the standard MediaPlay implementation in Our Application:

- **Initialization and Setup:**
 - ExoPlayer is initialized and configured in our music playback screen to handle audio files. This setup includes defining the player instance, setting the media source, and preparing the player for playback.
- **User Interface Integration:**
 - The player is integrated with a user-friendly interface that includes play, pause, and seek controls. This ensures that users can easily interact with the player and manage their music playback.
- **Handling Playback Events:**
 - ExoPlayer provides listeners for various playback events (e.g., buffering, errors, completion). These listeners are used to update the UI and handle any issues that arise during playback.
- **Adaptive Streaming:**
 - For streaming music, ExoPlayer automatically adjusts the quality based on the network conditions, ensuring continuous playback without interruptions.

```

@OptIn(UnstableApi::class)
private fun setupExoPlayer(){
    MyExoplayer.getCurrentSong()?.apply { this: SongModel
        binding.songTitleTextView.text = this.title
        Glide.with(requireContext()).load(this.coverUrl).circleCrop()
            .into(binding.songCoverImageView)
        exoPlayer = MyExoplayer.getInstance() !!
        exoPlayer.addListener(playerListener)
        binding.playerView.player = exoPlayer
        binding.playerView.showController()
    }
}

```

The setupExoPlayer function initializes and configures ExoPlayer for music playback in an Android application. It performs the following steps:

1. Retrieve the Current Song:
 - o Gets the current song details using MyExoplayer.getCurrentSong().
2. Update UI Elements:
 - o Sets the song title in a TextView.
 - o Loads the song cover image into an ImageView using Glide with a circular crop transformation.
3. Initialize ExoPlayer:
 - o Gets an instance of ExoPlayer from MyExoplayer.
4. Add a Playback Listener:
 - o Adds a listener to ExoPlayer to handle playback events.
5. Bind ExoPlayer to PlayerView:
 - o Sets the PlayerView's player to the ExoPlayer instance.
6. Show Playback Controls:
 - o Displays the playback controls on the PlayerView.

This function ensures that the UI is updated with the current song's details and that ExoPlayer is properly set up for playback, providing a smooth and interactive user experience.

5.11 Chat With Doctor

The doctor chat section in our application is designed to facilitate seamless communication between users and healthcare professionals. This section consists of three main screens, each serving a specific purpose to ensure an efficient and user-friendly experience.

1. Recent Conversations Screen:

- The first screen displays the user's recent conversations with doctors. It provides a quick overview of past interactions, allowing users to easily continue a previous conversation or review their chat history. This screen is designed for quick access and organization, ensuring that users can efficiently manage their communications.

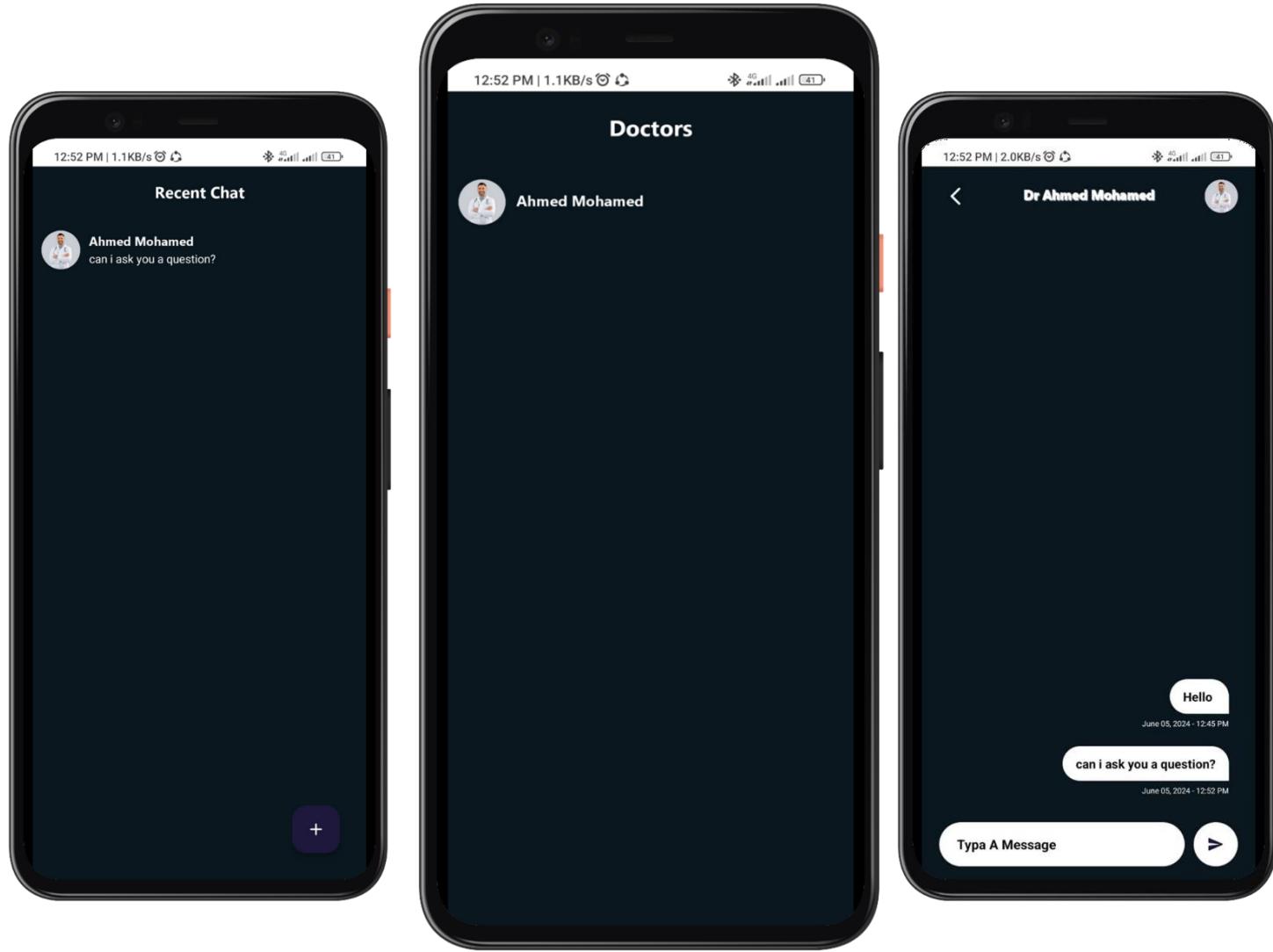
2. Select Doctor Screen:

- The second screen allows users to choose a doctor for their consultation. Users can browse through a list of available doctors, view their profiles, specialties, and availability. This screen helps users find the most suitable healthcare professional for their needs, ensuring they receive tailored advice and support.

3. Chat Screen:

- The third screen is the chat interface where users can engage in real-time conversations with their selected doctor. This screen provides a user-friendly messaging environment, supporting text messages, attachments, and possibly multimedia messages. The chat interface is designed to be intuitive, ensuring that users can easily communicate their concerns and receive timely responses from healthcare professionals.

By incorporating these three screens, the doctor chat section of our application ensures that users have a streamlined and effective way to seek medical advice, manage their communications, and receive the support they need to address their health concerns.



This Kotlin code consists of two functions: `sendMessage` and `sendMessageToFireBase`. They work together to send a chat message from a patient to a doctor and store it in a Firebase Firestore collection.

SendMessage Function

```
private fun sendMessage() {
    val message: HashMap<Any, Any> = HashMap()
    message[Constants.KEY_SENDER_ID] = preferenceManager.getString(Constants.KEY_PATIENT_ID)
    message[Constants.KEY_RECEIVER_ID] = receiverDoctor.id!!
    message[Constants.KEY_MESSAGE] = binding.inputMessage.text.toString()
    message[Constants.KEY_TIMESTAMP] = Date()
    sendMessageToFireBase(Constants.KEY_COLLECTION_CHAT,
        message,
        conversionId,
        isReceiverAvailable
    )
    binding.inputMessage.text = null
}
```

- **Creating a Message HashMap:**

- A `HashMap` named `message` is created to hold the details of the chat message.

- **Adding Message Details:**

- **Sender ID:** The ID of the patient sending the message, retrieved from `preferenceManager`.
- **Receiver ID:** The ID of the doctor receiving the message, obtained from `receiverDoctor`.
- **Message Content:** The text of the message, taken from an input field (`binding.inputMessage.text.toString()`).
- **Timestamp:** The current date and time when the message is sent.

- **Sending the Message:**

- The `sendMessageToFireBase` function is called with the message details and additional parameters (`conversionId` and `isReceiverAvailable`).

SendMessageToFireBase Function

```
fun fun sendMessageToFireBase(  
    collectionName: String,  
    message: java.util.HashMap<Any, Any>,  
    conversionId: String?,  
    isReceiverAvailable: Boolean  
) {  
  
    mFireStore.collection(collectionName).add(message)  
    if (conversionId != null) {  
        goUpdateConversion()  
  
    } else {  
        goAddConversion()  
  
    }  
}
```

1. Adding Message to Firestore

- o The message is added to a specified Firestore collection (`collectionName`).

2. Handling Conversation Updates:

- o If `conversionId` is not null, it indicates an existing conversation, so `goUpdateConversion` is called to update the conversation details.
- o If `conversionId` is null, it indicates a new conversation, so `goAddConversion` is called to create a new conversation record.

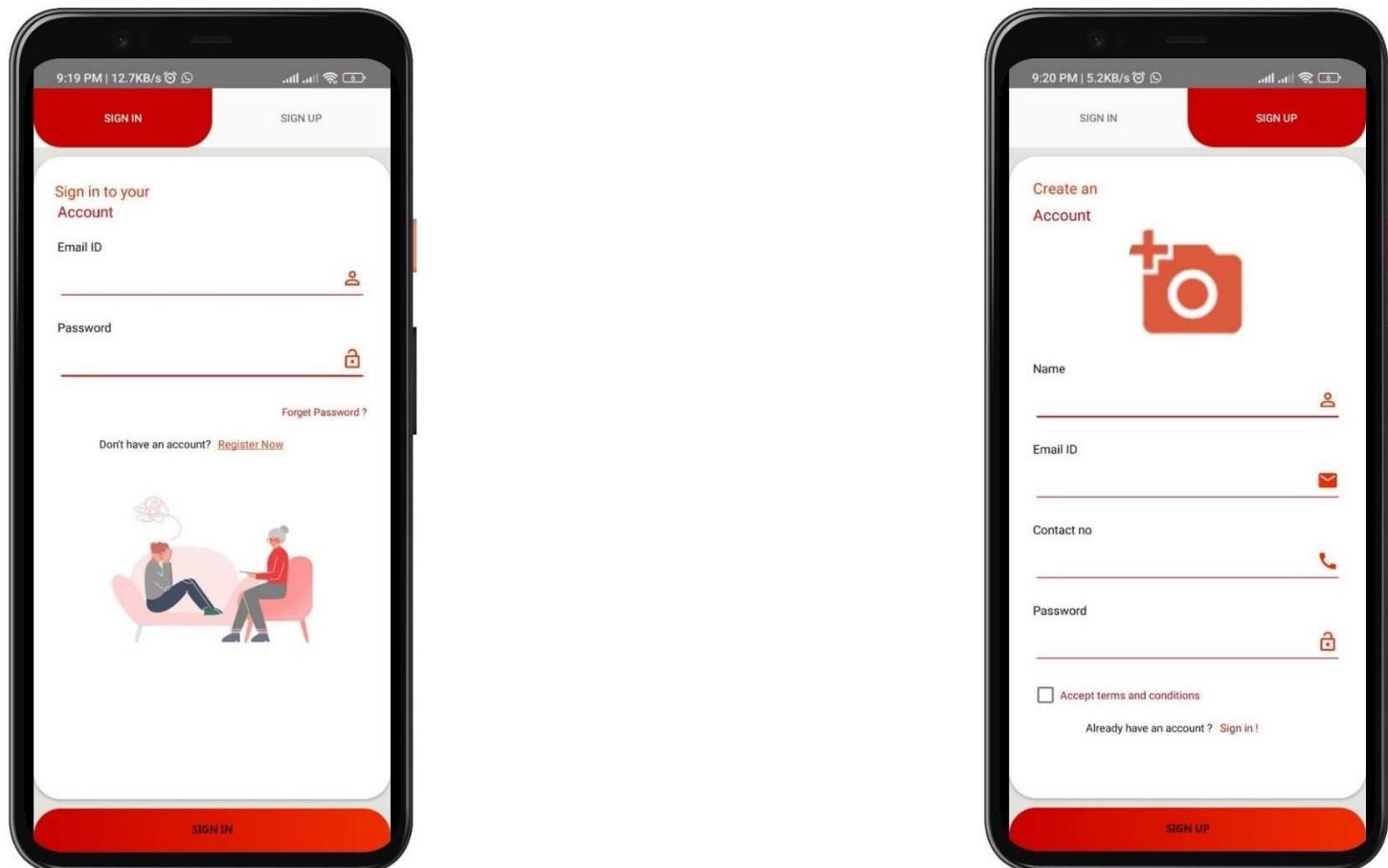
Summary

- **sendMessage:** Prepares the message details and calls `sendMessageToFireBase`.
- **sendMessageToFireBase:** Adds the message to a Firestore collection and updates or creates a conversation record based on whether `conversionId` exists.

This setup allows for efficient sending and handling of chat messages between patients and doctors, leveraging Firebase Firestore for real-time data storage and retrieval.

Insomnia Doctor

5.12. Authentication



This Kotlin function `isValidDetails()` is used to validate user input fields, specifically for email and password, in a sign-in form. Let's go through it step by step:

```

private fun isValidDetails(): Boolean {
    if (binding.editEmailSignIN.text!!.toString().trim().isEmpty()) {
        binding.editEmailSignIN.error = resources.getString(R.string.enter_email)
        binding.editEmailSignIN.requestFocus()
        return false
    }
    else if (binding.editPassSignIn.text!!.toString().trim().isEmpty()) {
        binding.editPassSignIn.error = resources.getString(R.string.enter_email)
        binding.editPassSignIn.requestFocus()
        return false
    }
    return true
}

```

- **private fun isValidDetails(): Boolean { ... }**
 - This is a private function named **isValidDetails** that returns a Boolean (true or false).
 - **Email Field Validation:**

```

if (binding.editEmailSignIN.text!!.toString().trim().isEmpty()) {
    binding.editEmailSignIN.error =
        resources.getString(R.string.enter_email)
    binding.editEmailSignIN.requestFocus() return false
}

```

- Checks if the **editEmailSignIN** field is empty after trimming whitespaces.
- If it is empty:

- Sets an error message using a string resource **R.string.enter_email**.
- Requests focus on **editEmailSignIN** to direct user input there.
- Returns **false**, indicating validation failure.

Password Field Validation

```

else if (binding.editPassSignIn.text!!.toString().trim().isEmpty()) {
    binding.editPassSignIn.error =
        resources.getString(R.string.enter_email)
}

```

If the email field is not empty (the else if block):

- Checks if the editPassSignIn field is empty after trimming whitespaces.
 - If it is empty:
 - Sets an error message using a string resource R.string.enter_email.
 - Requests focus on editPassSignIn to direct user input there.
 - Returns false, indicating validation failure.

Validation Success:

```
return true
```

Summary:

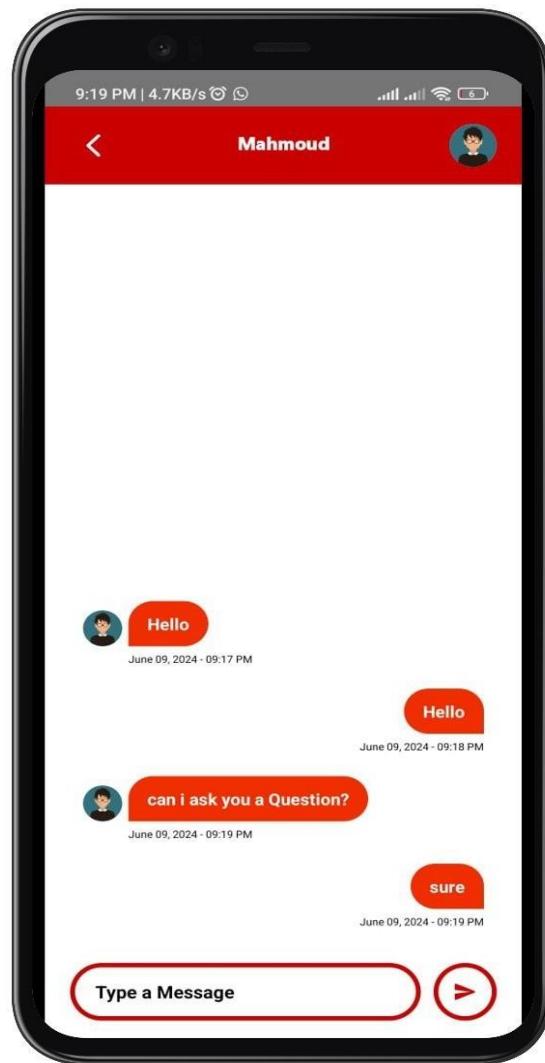
- **Purpose:** To validate email and password fields in a sign-in form.
- **Behavior:**
 - Checks if email field is empty and shows an error if so.
 - Checks if password field is empty and shows an error if so.
 - Returns true if both fields are filled, indicating validation success.
 - Returns false and focuses on the first empty field if any field is empty.

5.13 Chats

Recent Chat Screen



Chat Screen



Chapter Six

The Used Algorithm

- 1-Random Forest
- 2-Gradient Boosting
- 3-Logistic Regression

6.1 Supervised Learning:

1-Random Forest : Random Forest is an ensemble learning method that constructs multiple decision trees during training and merges them to get a more accurate and stable prediction. It can be used for both classification and regression tasks.

Key Concepts of Random Forest:

1. **Ensemble Method:** Combines predictions from multiple decision trees.
2. **Bagging (Bootstrap Aggregating):** Trains each decision tree on a different subset of the training data sampled with replacement.
3. **Random Feature Selection:** Selects a random subset of features for each split in the decision trees.
4. **Out-of-Bag (OOB) Error Estimation:** Uses the samples not included in the bootstrap sample to estimate the model performance.

2-Gradient Boosting: Gradient Boosting is another powerful supervised learning technique, commonly used for both classification and regression tasks. It builds models in a sequential manner, where each subsequent model aims to correct the errors made by the previous ones. This approach results in highly accurate models capable of handling complex datasets.

Key Concepts of Gradient Boosting

1. **Ensemble Method:** Combines the predictions of multiple models (weak learners) to produce a stronger model.
2. **Sequential Learning:** Models are built sequentially, each new model focusing on the errors made by the previous models.
3. **Gradient Descent:** Optimization algorithm used to minimize the loss function by adjusting the model parameters.

3-Logistic Regression: Logistic Regression is a statistical model commonly used for binary classification problems, where the goal is to predict one of two possible outcomes. Despite its name, logistic regression is a classification algorithm rather than a regression algorithm. It estimates the probability that a given input belongs to a certain class.

Key Concepts of Logistic Regression

1. **Sigmoid Function:** The core of logistic regression is the sigmoid function, which maps any real-valued number into a value between 0 and 1. This is useful for predicting probabilities.
2. **Logit Function:** The logit function is the inverse of the sigmoid function. It is used to relate the linear combination of inputs to the probability of the target class.
3. **Cost Function:** The cost function used in logistic regression is the binary cross-entropy (or log-loss), which measures the performance of the classification model whose output is a probability value between 0 and 1.
4. **Gradient Descent:** Optimization algorithm used to minimize the cost function by adjusting the model parameters.

6.2 Types of algorithms

1-Random forest:

1. Classification Random Forest

This is used when the target variable is categorical. The goal is to classify the input data into predefined classes.

Example:

- **Medical Diagnosis:** Classifying patients as having a disease or not based on their medical records.
- **Spam Detection:** Classifying emails as spam or not spam.

2. Regression Random Forest

This is used when the target variable is continuous. The goal is to predict a numeric value.

Example:

- **House Price Prediction:** Predicting the price of a house based on its features.
- **Stock Market Prediction:** Forecasting stock prices based on historical data.

2-Gradient Boosting:

1. Gradient Boosting for Classification (Gradient Boosting Classifier)

This variant is used when the target variable is categorical, and the goal is to classify the input data into predefined classes.

Example:

- **Spam Detection:** Classifying emails as spam or not spam.
- **Customer Churn Prediction:** Predicting if a customer will leave a service.

2. Gradient Boosting for Regression (Gradient Boosting Regressor)

This variant is used when the target variable is continuous, and the goal is to predict numeric values.

Example:

- **House Price Prediction:** Predicting the price of a house based on its features.
- **Sales Forecasting:** Predicting future sales based on historical data.

3-**Logistic Regression:**

1. Binary Logistic Regression

Binary logistic regression is the simplest form, used when the target variable has two classes. The goal is to predict the probability of the binary outcome.

Example:

- **Medical Diagnosis:** Predicting whether a patient has a disease (yes/no).
- **Spam Detection:** Classifying emails as spam or not spam.

2. Multinomial Logistic Regression

Multinomial logistic regression is used when the target variable has more than two classes. It generalizes binary logistic regression by using a softmax function to handle multiple classes.

Example:

- **Iris Classification:** Classifying different species of the Iris flower.
- **Handwritten Digit Recognition:** Classifying digits from 0 to 9.

6.3 Advantages Of Algorithms

1-Random forest:

Random Forest is a versatile and powerful ensemble learning algorithm that offers several advantages, making it widely used across various domains. Here are some of its key advantages:

1. Highly Accurate:

Random Forests typically provide high accuracy in classification and regression tasks. By combining the predictions of multiple decision trees trained on different subsets of the data, Random Forests reduce overfitting and produce robust models.

2. Handles Nonlinearity and Interactions:

Random Forests can capture complex nonlinear relationships and interactions between features in the data without the need for feature scaling or transformation. This makes them suitable for a wide range of tasks, including those with high-dimensional data or intricate relationships.

3. Implicit Feature Selection:

Random Forests naturally perform feature selection by evaluating the importance of each feature based on how much they decrease the impurity in the tree nodes. This helps in identifying the most relevant features for prediction, reducing the dimensionality of the problem and improving model interpretability.

4. Robust to Overfitting:

By aggregating the predictions of multiple trees, Random Forests are less prone to overfitting compared to individual decision trees. Tuning parameters such as the number of trees and maximum depth can further control overfitting, resulting in more generalizable models.

5. Handles Missing Values and Outliers:

Random Forests can handle missing values and outliers in the data without requiring preprocessing. They are robust to noise and can effectively deal with datasets containing irregularities, making them suitable for real-world data with imperfections.

6. Parallelizable and Scalable:

Random Forests can be trained in parallel, enabling efficient processing of large datasets and reducing training time. This parallelizability makes them scalable and suitable for applications requiring fast model training and prediction.

7. Reduces Variance:

By averaging the predictions of multiple trees, Random Forests reduce the variance of the model, leading to stable and reliable predictions. This ensemble approach enhances model robustness and makes Random Forests less sensitive to fluctuations in the data.

2-Gradient Boosting:

1. High Predictive Accuracy:

Gradient Boosting typically produces highly accurate models, often outperforming other machine learning algorithms in terms of predictive accuracy. By sequentially improving the model's predictions based on the errors of previous models, Gradient Boosting reduces bias and variance, resulting in more accurate predictions.

2. Handles Nonlinearity and Complex Interactions:

Gradient Boosting can capture complex nonlinear relationships and interactions between features in the data. By using decision trees as base learners, Gradient Boosting can model intricate relationships without the need for feature scaling or transformation.

3. Feature Importance Estimation:

Gradient Boosting provides a measure of feature importance, indicating which features are most influential in making predictions. This feature importance analysis helps in understanding the underlying patterns in the data and identifying the most relevant features for prediction, aiding in feature selection and model interpretability.

4. Robustness to Overfitting:

Gradient Boosting employs techniques such as shrinkage (learning rate) and tree constraints (maximum depth, minimum samples per leaf) to control overfitting. By iteratively fitting weak learners to the residuals of previous models, Gradient Boosting reduces overfitting and produces more generalizable models.

3-Logistic Regression:

Logistic Regression is a fundamental and widely used statistical technique with several advantages, making it a popular choice for various machine learning tasks. Here are some of the key advantages of Logistic Regression:

1. Simple and Interpretable:

Logistic Regression is a straightforward algorithm that is easy to understand and interpret. The model produces coefficients for each feature, representing the impact of that feature on the predicted probability of

the outcome. This interpretability makes Logistic Regression particularly useful for tasks where understanding the relationships between predictors and the target variable is important.

2. Efficient for Small Datasets:

Logistic Regression performs well even with relatively small datasets, making it suitable for scenarios where data availability is limited. It is computationally efficient and does not require extensive computational resources or time-consuming parameter tuning.

3. Probabilistic Predictions:

Logistic Regression predicts the probability of the outcome rather than just the binary classification. This provides more nuanced insights into the likelihood of different outcomes and allows for better decision-making, especially when uncertainty is present in the predictions.

4. Robust to Noise:

Logistic Regression is less susceptible to overfitting than more complex models, making it robust to noise and irrelevant features in the data. Regularization techniques such as L1 (Lasso) and L2 (Ridge) regularization can further improve model generalization and prevent overfitting.

6.4 Disadvantages Of Algorithms :

1-Random forest:

1. Lack of Interpretability:

Random Forest models, especially when composed of a large number of trees, can be difficult to interpret compared to simpler models like decision trees or logistic regression. Understanding the underlying decision-making process and feature interactions can be challenging, particularly for complex datasets.

2. Computationally Expensive:

Training a Random Forest model can be computationally expensive, especially for large datasets with many features and trees. Each tree in the forest is trained independently, and the ensemble process requires aggregating the predictions of multiple trees, which can increase training time and resource requirements.

3. Memory Consumption:

Random Forest models can consume significant memory, particularly when dealing with a large number of trees or high-dimensional datasets. Storing multiple decision trees and their associated data structures (e.g.,

nodes, leaves) can require substantial memory resources, limiting the scalability of Random Forests for very large datasets.

4. Potential Overfitting:

While Random Forests are less prone to overfitting than individual decision trees, they can still overfit noisy datasets with complex relationships. Increasing the number of trees or allowing deeper trees can exacerbate overfitting, leading to reduced generalization performance on unseen data.

2-Gradient Boosting:

1. Susceptible to Overfitting:

Gradient Boosting models, especially when the number of trees (iterations) is high or when the trees are deep, can be prone to overfitting. Overfitting occurs when the model captures noise or irrelevant patterns in the training data, leading to poor generalization performance on unseen data.

2. Highly Sensitive to Hyperparameters:

Gradient Boosting models have several hyperparameters that require tuning, such as the learning rate, tree depth, and the number of trees. Finding the optimal combination of hyperparameters can be challenging and computationally expensive, and the performance of the model may be sensitive to their values.

3. Computationally Intensive:

Training Gradient Boosting models can be computationally intensive, especially for large datasets or when using complex base learners (e.g., decision trees). Each iteration of the boosting process involves fitting a new weak learner to the residuals of the previous model, which can increase training time and resource requirements.

4. Not Interpretable by Default:

Gradient Boosting models, particularly when composed of many trees, are often considered black-box models. While feature importance scores can provide insights into which features are influential, understanding the precise decision logic of individual trees or the ensemble as a whole can be challenging.

3-Logistic Regression:

1. Limited to Linear Decision Boundaries:

Logistic regression assumes a linear relationship between the independent variables and the log-odds of the dependent variable. It may not perform well when the relationship is highly nonlinear, as it cannot capture complex interactions or higher-order relationships between features.

2. Not Suitable for Non-linear Relationships:

Logistic regression is not effective for modeling complex, non-linear relationships between predictors and the response variable. It may underperform when the decision boundary is not linear or when there are significant interactions between features.

3. Assumption of Independence of Observations:

Logistic regression assumes that observations are independent of each other. Violations of this assumption, such as autocorrelation or clustering of data points, can lead to biased parameter estimates and inaccurate predictions.

4. Sensitive to Outliers:

Logistic regression is sensitive to outliers, especially when the outliers are influential and have a disproportionate impact on the estimated coefficients. Outliers can distort the estimated coefficients and affect the performance of the model.

6.5 Algorithms Work:

1-Random forest:

1. Bootstrapped Sampling (Random Sampling with Replacement):

- Random Forest starts by creating multiple bootstrap samples from the original dataset. Each bootstrap sample is created by randomly selecting observations from the original dataset with replacement. This means that some observations may be sampled multiple times, while others may not be sampled at all.

2. Random Feature Selection:

- At each node of the decision tree, a random subset of features is considered for splitting. This random subset is typically a square root or logarithm of the total number of features. This feature selection helps to decorrelate the trees and introduces diversity in the forest.

3. Growing Decision Trees:

- For each bootstrap sample, a decision tree is grown. The decision trees are typically grown to their maximum depth without pruning. Each node of the tree is split using the best split among the random subset of features selected earlier.

4. Voting (Classification) or Averaging (Regression):

- Once all the trees are grown, predictions are made for each observation in the dataset. In the case of classification, each tree "votes" for a class, and the class with the most votes is assigned to the observation. In regression, the predictions of all trees are averaged to obtain the final prediction.

2-Gradient Boosting:

1. Initialize the Model:

- Gradient Boosting starts by initializing the model with a simple model, often a single decision tree or a constant value. This initial model serves as the starting point for the iterative learning process.

2. Compute Residuals:

- For each iteration, the algorithm computes the residuals (the differences between the actual and predicted values) of the current model. The residuals represent the errors or mistakes made by the current model on the training data.

3. Fit a Weak Learner to the Residuals:

- A weak learner, typically a decision tree with limited depth (a shallow tree), is then trained to predict the residuals of the current model. The goal is to find a model that can capture the patterns or trends in the residuals and improve upon the current model's predictions.

3-Logistic Regression:

1. Data Preparation:

- Before training the model, the dataset is typically preprocessed and prepared. This includes tasks such as handling missing values, encoding categorical variables, and splitting the data into training and testing sets.

2. Model Initialization:

- Logistic Regression starts by initializing the model parameters, including the coefficients (weights) for each feature and the intercept (bias). These parameters are typically initialized randomly or with zeros.

3. Calculate Log-Odds:

- Logistic Regression models the relationship between the independent variables (features) and the log-odds of the binary outcome using a linear function. The log-odds (logit) of the probability of the positive class is calculated as a linear combination of the feature values and model coefficients, plus the intercept:

6.6 Overfitting in Algorithms :

1-Random forest:

Complexity of Individual Trees:

- Random Forests can consist of a large number of decision trees, and if each tree is allowed to grow too deep (i.e., with many levels), they may capture noise in the training data and produce overly complex models.

High Variance:

- Random Forests have the potential for high variance due to the use of multiple decision trees. Each tree in the forest may capture different aspects of the data, leading to variability in predictions, especially if the number of trees in the forest is large.

Overfitting to Noisy Data:

- Random Forests may overfit to noisy or irrelevant features in the training data, especially if these features are given undue importance during the tree-building process.

Overfitting to Minority Classes:

- In imbalanced datasets where one class is significantly more prevalent than others, Random Forests may overfit to the majority class and fail to adequately capture the minority classes, leading to biased predictions.

2-Gradient Boosting:

Model Complexity:

- Gradient Boosting can create complex models, especially if the base learners (e.g., decision trees) are allowed to grow too deep or if too many trees are included in the ensemble. Overly complex models may capture noise or random fluctuations in the training data, leading to overfitting.

Overfitting to Residuals:

- Gradient Boosting iteratively fits weak learners to the residuals (errors) of the previous model. If the algorithm is allowed to continue training for too many iterations or if the learning rate is too high, it may overfit to the residuals and fail to generalize well to unseen data.

Overfitting to Noisy Data:

- Gradient Boosting may overfit to noisy or irrelevant features in the training data, especially if these features are given undue importance during the boosting process. This can lead to overly complex models that do not generalize well.

Overfitting to Outliers:

- Outliers in the training data can disproportionately influence the model's predictions, especially if the algorithm is not robust to outliers. Gradient Boosting may overfit to outliers and fail to generalize well to new data.

3-Logistic Regression:

Large Number of Features:

- Logistic regression models with a large number of features relative to the number of observations are more prone to overfitting. In such cases, the model may learn noise or irrelevant patterns in the data, rather than capturing the underlying relationships between features and the target variable.

High Model Complexity:

- If the logistic regression model is allowed to include interactions or higher-order terms between features, it may become overly complex and overfit the training data. Including too many features or polynomial terms can lead to a model that fits the training data too closely.

Multicollinearity:

- Multicollinearity occurs when two or more features are highly correlated with each other. In logistic regression, multicollinearity can lead to unstable coefficient estimates and inflated standard errors, making the model prone to overfitting.

Outliers:

- Outliers in the training data can disproportionately influence the estimated coefficients in logistic regression. If the model is not robust to outliers, it may overfit to these observations and fail to generalize well to new data.

6.7 Underfitting in Algorithms :

1-Random forest:

Insufficient Trees:

- Random Forests with too few trees may not have enough complexity to capture the patterns in the training data effectively. If the number of trees in the forest is too small, the model may underfit and fail to learn the underlying relationships between features and the target variable.

• Shallow Trees:

- If the individual decision trees in the Random Forest are too shallow (i.e., have too few levels), they may not be able to capture the complexity of the data. Shallow trees may lead to underfitting, as they fail to capture the interactions and nonlinear relationships between features.

• Limited Features at Each Split:

- Random Forests use feature sampling (random subsets of features) at each split to introduce diversity into the ensemble. If the number of features considered at each split is too small (e.g., due to a low value of the `max_features` parameter), the model may underfit and fail to capture the important features in the data.

• Imbalanced Data:

- In imbalanced datasets where one class is significantly more prevalent than others, Random Forests may underfit the minority classes and fail to capture their distinguishing characteristics. This can lead to biased predictions and poor performance on the minority classes.

2-Gradient Boosting:

Insufficient Trees:

- Gradient Boosting models with too few weak learners (e.g., decision trees) may not have enough complexity to capture the patterns in the training data effectively. If the number of trees in the ensemble is too small, the model may underfit and fail to learn the underlying relationships between features and the target variable.

Weak Learner Complexity:

- If the individual weak learners (e.g., decision trees) in the Gradient Boosting ensemble are too simple (e.g., too shallow), they may not be able to capture the complexity of the data. Simple weak learners

may lead to underfitting, as they fail to capture the interactions and nonlinear relationships between features.

Weak Learner Type:

- Using weak learners that are too simple or not suitable for the dataset can lead to underfitting. For example, using decision stumps (single-node decision trees) as weak learners may not provide enough complexity to capture the underlying patterns in the data.

Low Learning Rate:

- A low learning rate (also known as shrinkage) in Gradient Boosting slows down the learning process and can lead to underfitting if set too low. A smaller learning rate allows for more gradual updates to the model parameters, but if it is too small, the model may fail to capture the underlying patterns in the data effectively.

3-Logistic Regression:

Linear Assumption:

- Logistic regression assumes a linear relationship between the independent variables (features) and the log-odds of the outcome. If the true relationship is nonlinear or complex, logistic regression may underfit and fail to capture the underlying patterns in the data.

Insufficient Features:

- Logistic regression models with too few features may not have enough complexity to capture the patterns in the training data effectively. If important features are omitted from the model, it may underfit and fail to learn the relationships between features and the target variable.

High Bias:

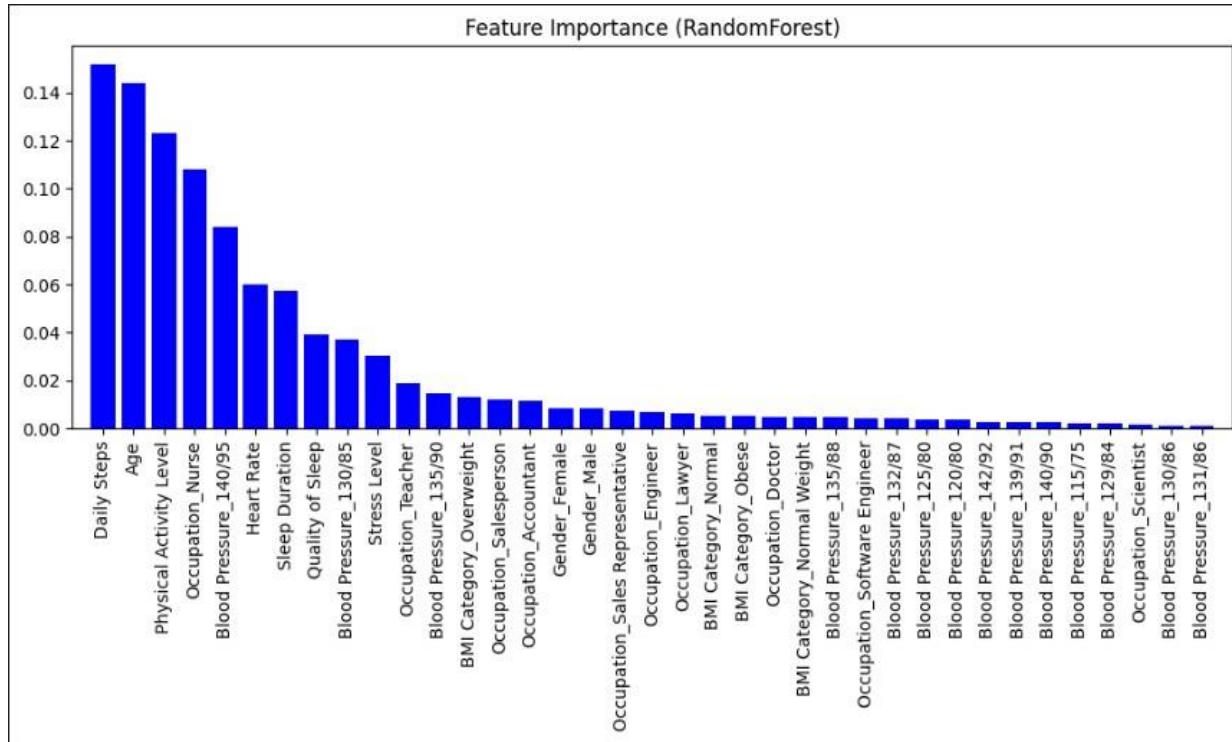
- Logistic regression models with high bias may underfit the training data by oversimplifying the underlying relationships. High bias can occur if the model is regularized too strongly or if the feature space is restricted too much, preventing the model from capturing the true patterns in the data.

Imbalanced Data:

- In imbalanced datasets where one class is significantly more prevalent than others, logistic regression may underfit the minority classes and fail to capture their distinguishing characteristics. This can lead to biased predictions and poor performance on the minority classes.

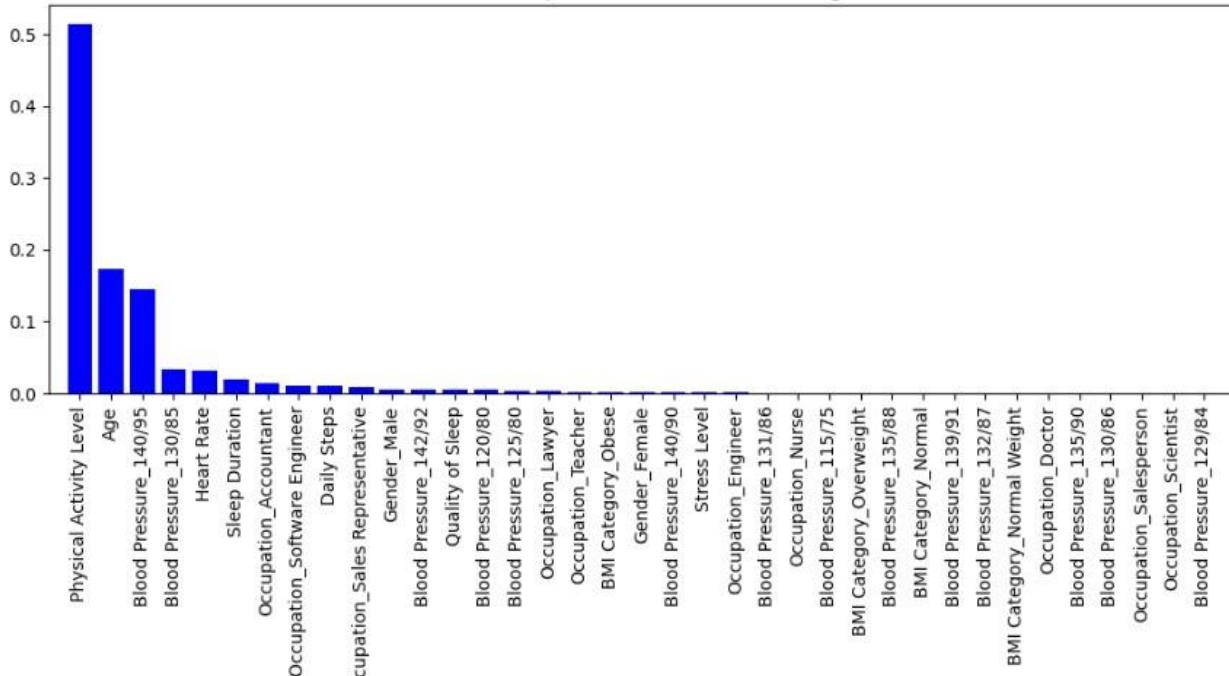
6.8 Data Classification and Accuracy:

1- Random forest: 88%



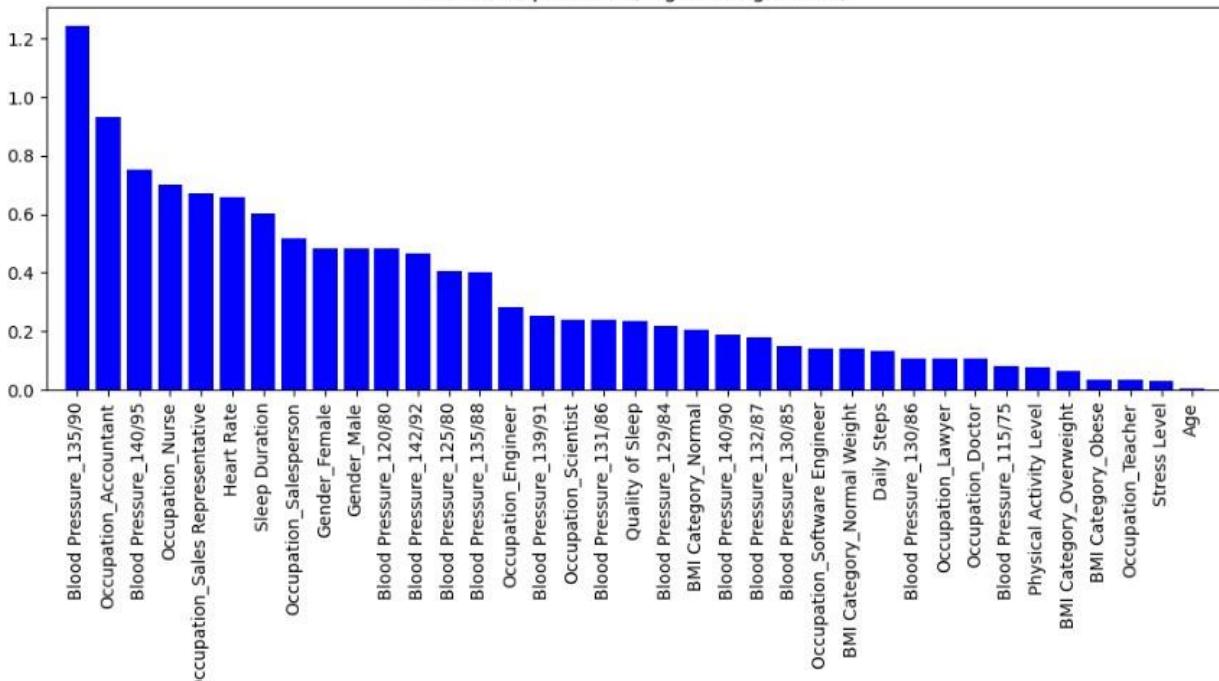
2-Gradient Boosting:88%

Feature Importance (Gradient Boosting)



3-Logistic Regression:85%

Feature Importance (Logistic Regression)



6.9 The algorithm we chose to use Random forest:

After we worked on the three algorithms and we knew what are their advantages and what are their disadvantages and what kind of data they work on we chose random forest because it got a good accuracy and after we made a data classification we found that the most passionate one works on our data.

Chapter Seven

DataSet

7.1 Dataset

Dataset Overview:

The Sleep Health and Lifestyle Dataset comprises 400 rows and 13 columns, covering a wide range of variables related to sleep and daily habits. It includes details such as gender, age, occupation, sleep duration, quality of sleep, physical activity level, stress levels, BMI category, blood pressure, heart rate, daily steps, and the presence or absence of sleep disorders.

Key Features of the Dataset:

Comprehensive Sleep Metrics: Explore sleep duration, quality, and factors influencing sleep patterns.

Lifestyle Factors: Analyze physical activity levels, stress levels, and BMI categories.

Cardiovascular Health: Examine blood pressure and heart rate measurements.

Sleep Disorder Analysis: Identify the occurrence of sleep disorders such as Insomnia and Sleep Apnea.

Dataset Columns:

Person ID: An identifier for each individual.

Gender: The gender of the person (Male/Female).

Age: The age of the person in years.

Occupation: The occupation or profession of the person.

Sleep Duration (hours): The number of hours the person sleeps per day.

Quality of Sleep (scale: 1-10): A subjective rating of the quality of sleep, ranging from 1 to 10.

Physical Activity Level (minutes/day): The number of minutes the person engages in physical activity daily.

Stress Level (scale: 1-10): A subjective rating of the stress level experienced by the person, ranging from 1 to 10.

BMI Category: The BMI category of the person (e.g., Underweight, Normal, Overweight).

Blood Pressure (systolic/diastolic): The blood pressure measurement of the person, indicated as systolic pressure over diastolic pressure.

Heart Rate (bpm): The resting heart rate of the person in beats per minute.

Daily Steps: The number of steps the person takes per day.

Sleep Disorder: The presence or absence of a sleep disorder in the person (None, Insomnia, Sleep Apnea).

Details about Sleep Disorder Column:

None: The individual does not exhibit any specific sleep disorder.

Insomnia: The individual experiences difficulty falling asleep or staying asleep, leading to inadequate or poor-quality sleep.

Sleep Apnea: The individual suffers from pauses in breathing during sleep, resulting in disrupted sleep patterns and potential health risks.

Acknowledgement:

I would like to clarify that the data I am presenting is synthetic and created by me for illustrative purposes.

Chapter Eight

Reference

8.1 Reference

1. *Books:*

- *"**Principles and Practice of Sleep Medicine**" by **Meir H. Kryger, Thomas Roth, and William C. Dement** : This is a comprehensive textbook that covers various aspects of sleep disorders, including insomnia. It provides in-depth information on the physiology of sleep, diagnosis, and treatment of sleep disorders.

2. *Review Articles and Journals:*

- *"**The diagnosis and management of insomnia in clinical practice: a practical evidence-based approach**" by **Michael Sateia (2014)**: This review article provides a detailed overview of the diagnosis and management strategies for insomnia.
- *"**Insomnia: Definitions, Prevalence, Etiology, and Consequences**" by **Charles M. Morin and Colin A. Espie (2016)**: Published in Sleep Medicine Reviews, this paper offers a thorough examination of the definitions, prevalence, etiology, and consequences of insomnia.
- *"**Cognitive behavioral therapy for insomnia (CBT-I)**" by **Perlis, M.L., et al. (2015)**: This article in the journal Sleep Medicine Clinics provides a detailed overview of CBT-I, one of the most effective treatments for insomnia.

3. *Guidelines and Reports:*

- *American Academy of Sleep Medicine (AASM) Clinical Practice Guidelines for the Pharmacologic Treatment of Chronic Insomnia in Adults*: These guidelines provide evidence-based recommendations for the treatment of chronic insomnia.
- *National Institutes of Health (NIH) State-of-the-Science Conference Statement on Manifestations and Management of Chronic Insomnia in Adults*: This report outlines the findings and recommendations from a conference of experts in the field.

4. *Web Resources:*

- ***National Sleep Foundation (NSF)***: Their website offers a wealth of information on sleep health, including articles and resources specifically about insomnia.
 - ***Sleep Education by the American Academy of Sleep Medicine***: This website provides patient education resources, including detailed information about insomnia and its treatment.

Chapter Nine

Conclusion

9.1 Conclusion

This project culminates in the development of an application designed to assist individuals suffering from insomnia by offering a variety of services. Our primary objectives were successfully achieved:

- 1. Multifaceted Support:** We have created an application that allows users to initiate discussions with multiple doctors, facilitating access to medical consultations through a novel artificial intelligence experience.
- 2. Enhanced Doctor Accessibility:** The app streamlines the process of finding doctors and scheduling appointments, empowering users to book appointments conveniently via text, voice messages, or chat, saving them valuable time.
- 3. Improved Communication:** By improving communication channels, patients and doctors can exchange information more efficiently, enhancing overall healthcare interactions.
- 4. Diverse Medical Programs:** We introduced various medical programs aimed at fostering lifestyle changes among patients, thereby promoting better health outcomes.
- 5. Future Development:** Looking forward, we aim to further enhance and expand the services offered within the application to better meet the needs of our users in the near future.

Throughout this project, we overcame several challenges, including sourcing free content such as books and music for the services provided, and implementing innovative uses of artificial intelligence to improve the user experience and effectively address insomnia-related issues.

In essence, this project represents a successful initiative in providing innovative solutions to insomnia, and we hope it contributes to improving the quality of life for those affected by this condition in the future.

Chapter Ten

Future Work

10.1 Future Work

1. Expanding the study sample:

Future studies could be conducted with a larger, more diverse sample of participants to examine the impact of insomnia across different age groups, cultural backgrounds, and socioeconomic levels.

2. Developing therapeutic interventions:

The effectiveness of new types of cognitive behavioral therapy for insomnia (CBT-I) that incorporate new technologies such as virtual reality (VR) therapy or apps for mobile devices can be explored.

3. Study of biological factors:

Future research can be conducted to study the relationship between insomnia and some biological indicators, such as genes, hormones, and neural interactions, to gain a deeper understanding of the mechanisms of insomnia and how to treat it.

4. Technology impact analysis:

Due to the increased use of electronic devices, the impact of exposure to blue light from screens on sleep quality and insomnia can be studied, and strategies can be developed to reduce its negative impact.

5. Research into psychological and social factors:

Studies could be conducted to examine the impact of psychological factors such as anxiety and depression on insomnia, and how to develop comprehensive intervention programs that take these factors into account.

6. Economic evaluation of treatments:

The economic costs of insomnia and its impact on productivity and the health system can be explored, and the economic effectiveness of different interventions can be assessed.

7. Long-term effect of treatments:

Conduct studies that follow participants over long periods of time to understand the long-term impact of different treatments for insomnia, and ensure that positive results are sustained.

8. Environment and sleep:

Explore the effect of the surrounding environment (e.g., noise, temperature, and light) on sleep quality and how to improve environmental conditions to promote good sleep.