COMSATS University Islamabad,

Attock Campus, Attock Pakistan

# Productivity App

***By***

**Muhammad Muaaz Shoaib CIIT/FA20-BCS-074/ATK**

**Maryam Amjad CIIT/-FA20-BCS-009/ATK**

***Supervisor***

***Dr.Farhan Adil***

***Bachelor of Science in Computer Science***

***(2020-2024)***

## COMSATS University Islamabad,
## Attock Campus, Attock Pakistan

# Productivity App

**A project presented to**
**COMSATS University, Islamabad**

**In partial fulfillment**
**of the requirement for the degree of**

***Bachelors of Science in Computer Science***
***(2020-2024)***

**By**

**MUHAMMAD MUAAZ SHOAIB CIIT/FA20-BCS-074/ATK**
**MARYAM AMJAD CIIT/FA20-BCS-009/ATK**

# DECLARATION

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied the report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be a reproduction of some other, we will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Muhammad Muaaz Shoaib                                        Maryam Amjad

---------------------------                                        --------------------------

# CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS (CS) “Productivity App” was developed by **MUHAMMAD MUAAZ SHOAIB (CIIT/FA20-BCS-074/ATK)** and **MARYAM AMJAD (CIIT/FA20-BCS-009/ATK)** under the supervision of “Dr. Farhan Adil” and that in his opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Sciences

-------------------------------------

**Supervisor**

-------------------------------------

**External Examiner**

-------------------------------------

**Head of Department**

**(Department of Computer Science)**

# Executive Summary

Productivity App aims to develop a productivity enhancement application designed to revolutionize task management, focus maintenance, and overall productivity. With a suite of modules, our app empowers users to overcome daily challenges efficiently. Our app's purpose is to equip users with tools to streamline workflow, reduce cognitive load, and achieve goals effectively.

Central to our app is the Task Manager module, offering users a hub to organize tasks, set reminders, and monitor progress. By tracking tasks, users can manage workload effectively, promoting productivity and goal attainment. This module fosters effective time management, crucial for maximizing productivity.

Complementing task management is the Focus Mode module, addressing distractions prevalent in modern work environments. By creating a serene atmosphere and silencing notifications, our app enables deep work, enhancing concentration and productivity. Customizable settings cater to individual preferences, fostering sustained focus and efficient task completion.

Supplementary features include the Pomodoro Timer module for structured work sessions and the Habit Tracker module for cultivating positive habits. Seamless integration with other productivity tools ensures a cohesive workflow. Our app offers a holistic solution for enhancing productivity, empowering users to excel in their professional and personal endeavors.

# Acknowledgement

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Dr. Farhan Adil” a. Without his personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are grateful to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us with the values of honesty & hard work.

Muhammad Muaaz Shoaib Maryam Amjad

-------------------------- --------------------------

# Abbreviations

| | |
|---|---|
| **API** | Application Programming Interface |
| **AI** | Artificial Intelligence |
| **BO** | Business Objective |
| **CS** | Computer  Science |
| **CIIT** | Comsats Institute of Information and Technology |
| **CI** | Communication Interface |
| **FE** | Front end |
| **HI** | Hardware Interface |
| **FR** | Functional Requirement |
| **IDE** | Integrated development environment |
| **ID** | Identification |
| **SDD** | Software Design |
| **SRS** | Software Requirement Specifications |
| **SDK** | Software development kit |
| **UC** | Use Case |

# Table of Contents

# List of Figures

# List of Tables

# 1 Introduction.

## 1.1 Vision Statement

For individuals seeking to enhance their productivity and streamline daily tasks, the Productivity Plus app is a versatile digital tool designed to optimize time management and task organization. Unlike traditional productivity methods, our app offers a comprehensive solution that integrates task management, focus enhancement, and productivity analytic into a single platform. With intuitive features and seamless user experience, Productivity Plus empowers users to achieve their goals efficiently, whether at work, school, or home.

Our app caters to the diverse needs of users across various industries and lifestyles, providing customizable solutions to maximize productivity potential. Productivity App offers a unique and innovative approach to productivity enhancement. Unlike existing productivity tools that focus on isolated aspects of productivity, our app delivers a holistic solution that adapts to individual preferences and workflows, resulting in tangible improvements in efficiency and effectiveness.

Productivity Plus sets itself apart from competitors by offering a comprehensive suite of modules that address the entire productivity spectrum. From task management and focus optimization to habit formation,Notes keeping and productivity analytics, our app provides unparalleled value and utility. With continuous updates and enhancements based on user feedback and market trends, Productivity App is poised to become the go-to productivity companion for individuals seeking to excel in today's fast-paced world.

## 1.2 Related System Analysis/Literature Review

**Table 1 Related System Analysis with proposed project solution**

| Application Name | Weakness | Proposed Project Solution |
|---|---|---|
| Notepad in Phone | Limited functionality, lacking advanced features like images support.<br>No Virtual Assistant | Comprehensive features including multimedia support for images .<br>Integrated with AI powered chatbot. |
| Microsoft Todo | No Progress Tracking<br>No Virtual Assistant<br>No Focus Mode.<br>No Habit Tracker | Users can Track Progress of each Task.<br>AI powered Virtual Assistant.<br>Focus Mode to minimize distraction.<br>Habit building functionality to build healthy habit. |

**Table 2 : Comparison Table of Productivity App With Microsoft Todo**

| Features | Productivity App | Microsoft Todo |
|---|---|---|
| Tracking Progress | ✓ | ✕ |
| Virtual Assistant | ✓ | ✕ |
| Focus Mode | ✓ | ✕ |
| Habit Tracker | ✓ | ✕ |

| | | |
|---|---|---|
| Get Notification | ✓ | ✓ |
| Edit, Delete Task | ✓ | ✓ |

**Table 3 :Comparison table of Productivity App with Notes App**

| Features | Productivity App | Notes App |
|---|---|---|
| Text Notes | ✓ | ✓ |
| Image Notes | ✓ | ✗ |
| Edit, Delete Notes | ✓ | ✓ |
| AI powered chatbot | ✓ | ✗ |

**Table 4 :Comparison table of Productivity App with Pomodoro Productivity Timer App**

| Features | Productivity App | Pomodoro Productivity timer |
|---|---|---|
| Set desired Sessions | ✓ | ✗ |
| Set desired work duration | ✓ | ✓ |
| Set desired break duration | ✓ | ✓ |
| Analytics | ✓ | ✗ |

## 1.3 Project Deliverables

- **Proposal**

Proposal is a document which will be delivered to describe the need and purpose of the proposed system.

- **Complete project**

The Complete project includes mobile application. Front-end & backend source code will be included in the mobile application.

- **Reports**

A report will be delivered which includes software requirements specification (SRS), software design specification and other project requirement documents. Software requirement specification includes functional and non-functional requirements. Software design specification includes the DFD diagrams and other system diagrams.

## 1.4 Include Test Cases System Limitations/Constraints

*LI-1: Productivity app might occasionally face technical problems such as crashes or bugs, impacting its functionality and user experience.*
*LI-2: Users relying on the app cloud-based features may encounter limitations in offline environments or areas with poor internet connectivity.*
*LI-3: Our app is tailored to work seamlessly with the most recent version of Android. However, users on older devices or outdated operating systems may encounter compatibility issues.*
*LI4: Productivity app is currently only available in English, which may limit its accessibility to non-English speaking users.*
*LI5: Our app does not have the capability to address human errors, such as neglecting to enter a task or prematurely marking a task as completed before it is truly finished.*

## 1.5 Tools and Technologies

**Table 5 Tools and Technologies for Proposed Project**

| Tools And Technologies | **Tools** | **Version** | **Rationale** |
|---|---|---|---|
| | Visual Studio Code | 2024 | IDE |
| | Android Studio | 2023 | IDE |
| | Microsoft word | 2016 | Documentation |
| | Draw.io | | Diagrams |
| | **Technology** | **Version** | **Rationale** |
| | Flutter | 3.19.6 | Front-end Development |
| | Firebase | 9 | Back-end Development |
| | Gemini | | API integration |
| | Dart | 3.3.4 | Language for flutter |

## 1.6 Relevance to Course Modules

1. **Report Writing:** Equipped me with skills for documenting project requirements and details effectively.
2. **Software Engineering Concepts:** Provided a foundation in system architecture and design patterns, aiding in creating comprehensive diagrams and structuring the project systematically.
3. **Mobile Application Development**: Offered practical experience in building mobile apps, aiding in developing a user-friendly interface and implementing essential features.
4. **Database Course:** Taught principles of database management crucial for structuring app data efficiently and ensuring reliable storage and retrieval mechanisms.
5. **Human-Computer Interaction Course:** Provided insights into designing intuitive interfaces, resulting in a visually appealing and user-friendly app interface that enhances user experience.

# 2 Problem Definition

## 2.1 Problem Statement

In today's frenetic world, people often struggle to stay on top of their multiple responsibilities, leading to decreased productivity and organization. The constant juggling act between work, personal life, and other commitments can be overwhelming, causing stress, burnout, and missed deadlines. Furthermore, the numerous distractions that surround us only add to the problem, making it difficult to concentrate and complete tasks efficiently. To combat these challenges, our mobile app is designed to provide a personalized solution for boosting productivity and time management. By offering a suite of features and tools, our app helps users stay focused, prioritize tasks, and maximize their daily output. Whether you're a student aiming for academic success, a professional managing a heavy workload, or an individual seeking to simplify your workflow, our app is a versatile productivity tool that helps you achieve your goals and manage your responsibilities effectively.

## 2.2 Problem Solution

In today's fast-paced world, maintaining productivity amidst a lot of responsibilities is challenging. Balancing work, school, and personal commitments often lead to stress, burnout, and missed deadlines. Distractions exacerbate these issues, hindering focus and wasting time. Our mobile app aims to counter these challenges by boosting productivity. It offers a comprehensive to-do list to track tasks and prioritize work, ensuring deadlines are met. The calendar feature facilitates efficient task scheduling and reminders. Additionally, our app includes a focus mode, blocking distractions and fostering concentration. A versatile note-taking feature allows users to capture and organize ideas and information. By leveraging our app, users can enhance productivity, manage time effectively, and achieve goals while minimizing stress. It serves as the ultimate productivity tool, optimizing workflow and maximizing efficiency.

## 2.3 Objectives of the Proposed System

*BO-1: Enable users to create tasks efficiently.*
*BO-2: Provide tools for scheduling and managing time effectively.*
*BO-3: Offer features to minimize distractions and maintain focus during work sessions.*
*BO-4: Allow users to track progress and achieve milestones effectively.*
*BO-5: Implement reminders and notifications to keep users informed of upcoming tasks and deadlines.*
*BO-6: Provide a platform for users to capture ideas, notes, and information in text,and image formats.*
*BO-7: Offer insights into user productivity trends, task completion rates, and time allocation.*
*BO-8: Integrate with popular productivity tools and services to streamline workflows and enhance productivity.*
*BO-9: Provide offline capabilities to enable users to access and manage their notes even without an internet connection.*
*BO-10: Assist users in building and maintaining positive habits by providing tracking features.*
*BO-11: Offer strategies and tools to help users minimize distractions and maintain focus during work sessions.*

## 2.4 Scope

The productivity app aims to provide users with a comprehensive set of functionalities to enhance their efficiency and organization across various aspects of life. At its core, the app will offer robust task management capabilities, enabling users to create, and track tasks effectively. This includes features such as task lists and reminders to help users stay on top of their commitments. Additionally, the app will encompass comprehensive time management tools, including calendars and scheduling functionalities, enabling users to allocate their time wisely and manage their schedules efficiently. To foster concentration and minimize distractions, the app will incorporate a focus mode feature, which blocks distractions and creates an environment conducive to deep work. Moreover, users will have access to a versatile note-taking feature, facilitating the capture and organization of ideas, thoughts, and important information. Overall, the productivity app is designed to provide users with a solution to manage tasks, time, and focus effectively, empowering them to achieve their goals and optimize their productivity in their personal and professional lives.

## 2.5 Modules

### 2.5.1 Module 1: Task Manager

*FE-1: Easily create new tasks.*
*FE-2: User can mark tasks as completed.*
*FE-3: Set reminders for upcoming tasks.*
*FE-4: User can track progress.*

### 2.5.2 Module 2: Focus Mode

*FE-1: Allow user to turn focus more on and off.*
*FE-2: Block notifications and other distractions to maintain focus.*

### 2.5.3 Module 3: Pomodoro Timer

*FE-1: User can set work and break timers for effective time management.*
*FE-2: Automatically switch between work and break sessions.*
*FE-3: Monitor the number of Pomodoro cycles completed to track productivity.*
*FE-4: User can adjust timer durations and intervals according to their preferences.*

### 2.5.4 Module 4: Habit Tracker

*FE-1: Define specific goals or habits that users want to establish or track.*
*FE-2: Track daily, weekly, or monthly progress towards achieving set goals.*
*FE-3: Provide visual feedback to encourage users and reinforce positive habits.*

### 2.5.5 Module 5: Note Taking

*FE-1: Store various types of notes, including text, images*
*FE-2: Efficiently organize notes.*
*FE-3: Enable users to take assistant from Ai powered chatbot*

### *2.5.6* Module 6: Analytic

*FE-1: Monitor the tasks completed and time spent on tasks.*
*FE-2: Monitor no of work sessions performed.*
*FE-3: Monitor average session time.*
*FE-4: Longest productivity Session Completed*
*FE-4: Track progress towards set goals and provide insights into goal achievement.*

### 2.5.7 Module 7: Calendar

*FE-1: User can schedule event for upcoming days, weeks, or month.*
*FE-2: User can set reminder for the events.*
*FE-3: User can view upcoming events.*

### 2.5.8 Module 8: Integration with Other Productivity Tool

*FE-1: Integrate with popular productivity tools such as Google Gemini chat bot to boost productivity.*

# 3 Requirement Analysis

## 3.1 User classes and characteristics

**Table 6 : User class and characteristics**

| User class | Description |
|---|---|
| **Student** | Students find the "Productivity App" indispensable for managing academic responsibilities. They create to-do lists for assignments and study tasks, set study reminders, and employ focus mode for productive study sessions. The app helps students schedule classes, study sessions, and track their academic progress. They can capture lecture notes in different formats and use analytics to measure productivity and time allocation to subjects, ensuring academic success. |
| **Professionals** | Professionals use the app to optimize their work efficiency and organization. They create task lists for work projects, set deadlines, and rely on the focus mode for uninterrupted work periods. The app assists professionals in scheduling meetings and appointments, integrating with project management tools, and taking notes during work-related activities. The analytics feature helps them gauge their productivity and project advancements. |
| **Business Owners** | Entrepreneurs and small business owners employ the app to manage their businesses efficiently. They create task lists for business operations, set project deadlines, and focus on important business tasks. The app helps schedule meetings with clients and partners and integrates with business software and tools. Analytics enable them to track business productivity and growth. |
| **Educators and Trainers:** | Educators and trainers use the app to manage courses, lessons, and student interactions. They create task lists for course preparation, set deadlines for assignments, and employ the focus mode for focused teaching. The app assists them in scheduling classes and syncing with educational calendars. Analytics help them monitor teaching productivity and student performance. |

## 3.2 Requirement Identifying Technique

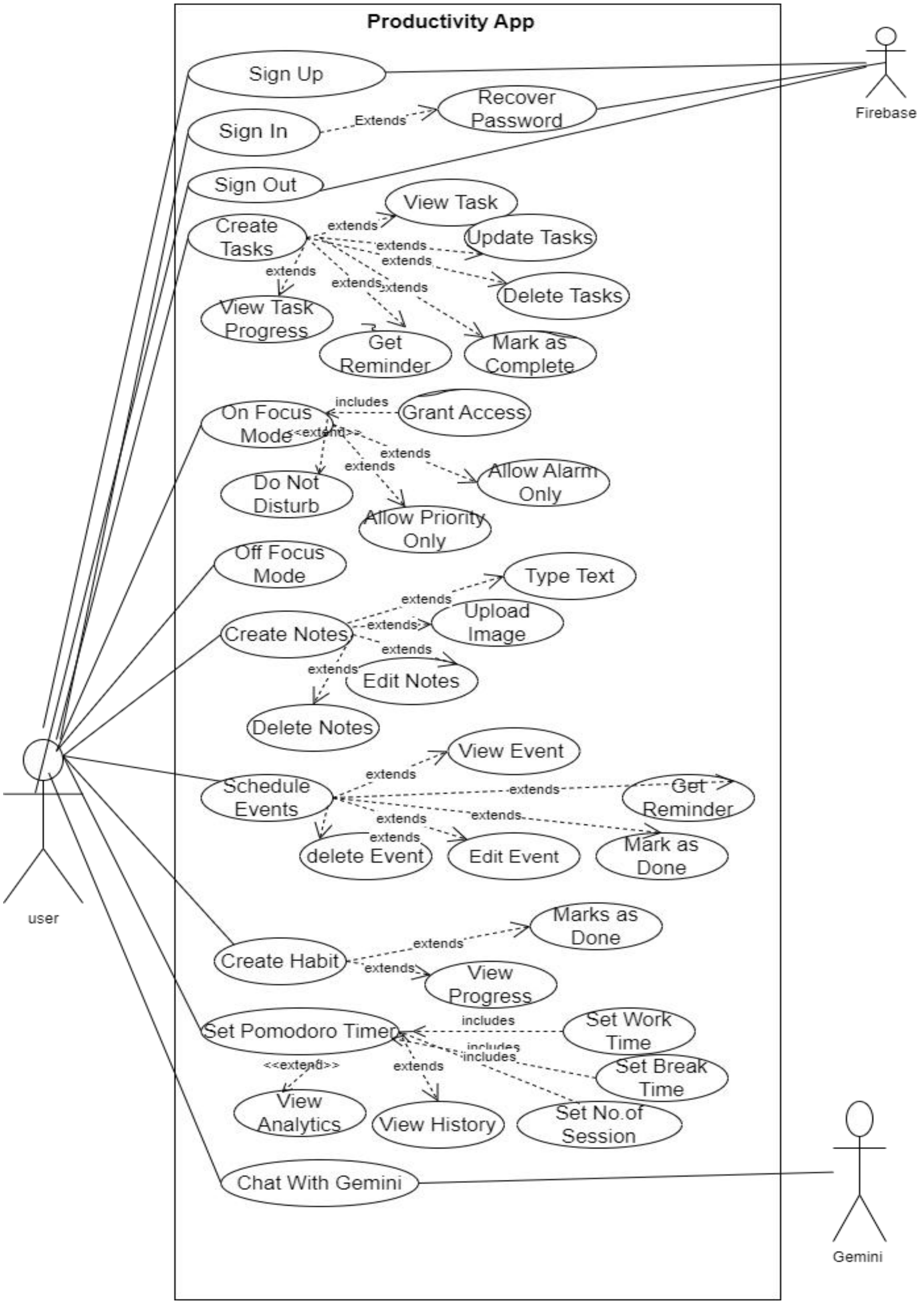


**Figure 1: Use Case Diagram**

**Table 7 : Use Case table of Sign Up**

| | |
|---|---|
| **Use Case ID:** | **UC-1** |
| **Use Case Name:** | Sign Up |
| **Actors:** | User |
| **Description:** | The "Sign Up" module allows a user to create a new account in the "Productivity App" by providing the necessary registration details, email and password. |
| **Trigger:** | User launches the app and selects the "Register" option. |
| **Preconditions:** | The user has installed the app on their device.<br>The user does not have an existing account in the system. |
| **Post conditions:** | If the registration is successful, a new user account is created, and the user gains access to the app's features.<br>If the registration fails, the user is informed of the failure, and they do not gain access |
| **Normal Flow:** | 1. The user launches the app.<br>2. The app displays the registration form.<br>3. The user enters their registration details, including email, username, and password.<br>4. The user submits the registration request.<br>5. The app validates the registration details.<br>6. If the details are valid and not already registered, a new account is created.<br>7. The app grants the user access to the main dashboard. |
| **Alternative Flows:** | If the user attempts to register with an email or username that is already in use, they receive an error message and are prompted to choose a different email or username. |
| **Exceptions:** | Network issues or server unavailability may prevent successful registration.<br>Registration details may not meet minimum requirements (e.g., a valid email address and password complexity). |
| **Business Rules** | Unique email addresses are required for each registered account. |
| **Assumptions:** | The user is providing valid registration details.<br>The registration process is secure and follows best practices for account creation and data protection |

**Table 8 : Use Case table of Sign In**

| | |
|---|---|
| **Use Case ID:** | **UC-2** |
| **Use Case Name:** | Sign In |
| **Actors:** | User |
| **Description:** | The "Login" module allows a user to access the "Productivity App" by providing valid authentication credentials. |
| **Trigger:** | User launches the app and selects the "Login" option. |
| **Preconditions:** | The user has installed the app on their device.<br>The user possesses a registered account in the system. |

| | |
|---|---|
| **Postconditions:** | If the user provides valid credentials, they are granted access to the app's features.<br>If the user provides incorrect credentials, they are denied access, and navigate back to the Screen |
| **Normal Flow:** | 1. The user launches the app.<br>2. The app displays the login screen.<br>3. The user enters their username and password.<br>4. The user submits the login request.<br>5. The app validates the user's credentials.<br>6. If the credentials are valid, the app grants access to the main dashboard.<br>7. If the credentials are invalid, the app displays an error message. |
| **Alternative Flows:** | If the user forgets their password, they can request a password reset. |
| **Exceptions:** | Network issues or server unavailability may prevent successful login. |
| **Business Rules** | User passwords must meet minimum security requirements, such as length greater than 8 . |
| **Assumptions:** | Users have registered accounts with valid credentials. |

**Table 9 : Use Case table of Create Task**

| | |
|---|---|
| **Use Case ID:** | **UC-3** |
| **Use Case Name:** | Create Task |
| **Actors:** | User |
| **Description:** | Enables the user to schedule a new task in the "Productivity App," with the ability to get reminders and track progress. |
| **Trigger:** | User selects the "Create Task" option from the app.. |
| **Preconditions:** | The user has logged into the app. |
| **Postconditions:** | A new task is scheduled, added to the user's task list, and includes a reminder and shows current progress. |
| **Normal Flow:** | 1.User selects the "Create Task" option.<br>2.The app displays the task Creation form.<br>3. User enters task details, including title, description, date, time.<br>4.User submits the task scheduling request.<br>5.The app validates the task details, reminder settings, and priority.<br>6.If the details are valid, a new task is created with the reminder and added to the task list. |
| **Alternative Flows:** | If the user attempts to schedule a task with missing or invalid details, they receive an error message and are prompted to provide the required information. User are not allowed to set a reminder for a date or time that has already passed. |
| **Exceptions:** | Network issues or server unavailability may prevent successful task scheduling. |
| **Business Rules** | Task details must include a title and a valid date and time |
| **Assumptions:** | The user is providing valid task scheduling details. |

**Table 10 : Use Case table of Update Task**

| | |
|---|---|
| **Use Case ID:** | **UC-4** |
| **Use Case Name:** | Update Task |
| **Actors:** | User |
| **Description:** | Allows the user to modify and update an existing task within the "Productivity App." |
| **Trigger:** | User selects the "Update Task" option for a specific task within the app |
| **Preconditions:** | The user has logged into the app. - There are existing tasks in the user's task list. |
| **Postconditions:** | The selected task is updated with the new information. |
| **Normal Flow:** | 1. User selects a specific task they wish to update.<br>2. The app displays the task details and an option to edit.<br>3. User makes the desired changes to the task, including title, description, date, time.<br>4. User confirms the changes.<br>5. The app validates the updated task details.<br>6. If the details are valid, the task is updated with the new information. |
| **Alternative Flows:** | If the user attempts to update a task with missing or invalid details, they receive an error message and are prompted to provide the required information. |
| **Exceptions:** | Network issues or server unavailability may prevent successful task scheduling. |
| **Business Rules** | Task details must include a title and a valid date and time |
| **Assumptions:** | The user is updating a valid task with accurate information |

**Table 11 : Use Case table of Delete Tasks**

| | |
|---|---|
| **Use Case ID:** | **UC-5** |
| **Use Case Name:** | Delete Tasks |
| **Actors:** | Users |
| **Description:** | Allows the user to remove a task from their task list in the "Productivity App. |
| **Trigger:** | User selects the "Delete Task" option for a specific task within the app. |
| **Preconditions:** | The user has logged into the app. - There are existing tasks in the user's task list |
| **Postconditions:** | The selected task is permanently removed from the user's task list |
| **Normal Flow:** | 1. User selects a specific task they wish to delete.<br>2. The app prompts the user to confirm the deletion.<br>3. User confirms the deletion.<br>4. The app removes the task from the user's task list. |
| **Alternative Flows:** | If the user cancels the task deletion, the task remains in the task list. |
| **Exceptions:** | Network issues or server unavailability may prevent successful task |

| | |
|---|---|
| | deletion. |
| **Business Rules** | Deletion of tasks is irreversible. - Users can only delete tasks that exist in their task list. |
| **Assumptions:** | The user is confirming the deletion of a specific task that already exists |

**Table 12 : Use Case table of Mark Task as Completed**

| | |
|---|---|
| **Use Case ID:** | **UC-6** |
| **Use Case Name:** | Mark as Completed |
| **Actors:** | Users |
| **Description:** | Allows the user to mark a task as completed in the "Productivity App." |
| **Trigger:** | User selects the "Mark as Completed" option for a specific task within the app. |
| **Preconditions:** | The user has logged into the app. - There are existing tasks in the user's task list. |
| **Postconditions:** | The selected task is marked as completed in the user's task list and progress is updated |
| **Normal Flow:** | 1. User selects a specific task they wish to mark as completed.<br>2. The app updates the task's status to "Completed." |
| **Alternative Flows:** | None |
| **Exceptions:** | Network issues or server unavailability may prevent successful task status update. |
| **Business Rules** | A task can only be marked as completed if it exists in the user's task list |
| **Assumptions:** | The user is marking a valid task as completed. |

**Table 13 : Use Case table of Turn on Focus Mode**

| | |
|---|---|
| **Use Case ID:** | **UC-7** |
| **Use Case Name:** | Turn on Focus Mode |
| **Actors:** | Users |
| **Description:** | Allows the user to activate the "Focus Mode" in the "Productivity App" with various notification options, including allowing only alarm notifications, activating the "Do Not Disturb" mode, and permitting notifications from priority apps only. |
| **Trigger:** | User selects the “Focus Mode" option within the app. |
| **Preconditions:** | User allows permission to app |
| **Postconditions:** | The selected focus mode settings are activated. |
| **Alternative Flows:** | None |
| **Exceptions:** | User didn’t grant permission to the app |
| **Business Rules** | Focus mode settings are based on user preferences. |
| **Assumptions:** | The focus mode activation process is secure and follows best practices for user preferences and system notifications. |

Table 14 : Use Case table of Turn off Focus Mode

| | |
|---|---|
| **Use Case ID:** | **UC-8** |
| **Use Case Name:** | Turn off Focus Mode |
| **Actors:** | User |
| **Description:** | Allows the user to deactivate or turn off the "Focus Mode" in the "Productivity App" and return to the regular notification settings. |
| **Trigger:** | User selects the "Turn off Focus Mode" option within the app. |
| **Preconditions:** | The user has logged into the app.<br>The app's "Focus Mode" is currently active. |
| **Postconditions:** | The "Focus Mode" is deactivated, and regular notification settings are restored. |
| **Normal Flow:** | 1. User selects "Turn off Focus Mode" within the app.<br>2. The app deactivates the active "Focus Mode." |
| **Alternative Flows:** | None. |
| **Exceptions:** | Network issues or server unavailability may affect the successful deactivation of "Focus Mode." |
| **Business Rules:** | Turning off "Focus Mode" restores regular notification settings. |
| **Assumptions:** | The user is deactivating an active "Focus Mode”. |

Table 15 : Use Case table of Take Notes

| | |
|---|---|
| **Use Case ID:** | **UC-9** |
| **Use Case Name:** | Take Notes |
| **Actors:** | User |
| **Description:** | Allows the user to create and save notes in text and image format. |
| **Trigger:** | User selects the "Take Notes" option within the app. |
| **Preconditions:** | The user has logged into the app and selected Note Taking Option |
| **Postconditions:** | The user has successfully created and saved a note in the selected format. |
| **Normal Flow:** | 1. User selects "Take Notes" within the app.<br>2. The app presents options for creating a note in text and picture,<br>3. User selects the desired format..<br>4. User confirms and saves the note. |
| **Alternative Flows:** | If the user denies storage access permission, the app displays an error and prompts the user to grant access. |
| **Exceptions:** | Storage full or assets missing may affect the successful storage |
| **Business Rules:** | Storage access is required for saving pictures. |
| **Assumptions:** | The user is creating notes in a valid format.<br>The storage access process is secure and follows best practices for user permissions and data handling. |

**Table 16 : Use Case table of Schedule Events**

| | |
|---|---|
| **Use Case ID:** | **UC-10** |
| **Use Case Name:** | Schedule Event |
| **Actors:** | User |
| **Description:** | Allows the user to schedule future tasks, view upcoming events, receive reminders for tasks, mark tasks as done, and view task progress percentage within the "Productivity App" calendar module. |
| **Trigger:** | User selects the "Calendar" option within the app. |
| **Preconditions:** | The user has logged into the app.<br>The app has a calendar feature with upcoming events. |
| **Postconditions:** | The user can see a list of upcoming events and access details for each event. |
| **Normal Flow:** | 1. User selects "Calendar within the app.<br>2. The app displays a list of upcoming events from the calendar.<br>3.User can Select a particular date to see the events on that date<br>3. User can click on an event to view additional details. |
| **Alternative Flows:** | None. |
| **Exceptions:** | Network issues or calendar unavailability may affect the successful retrieval of upcoming events. |
| **Business Rules:** | The app retrieves and displays events from the calendar.<br>Users can access event details through the app. |
| **Assumptions:** | The user relies on the calendar module for organizing and managing tasks and events effectively.<br>The calendar module follows standard conventions for task scheduling, reminder notifications, and progress tracking. |

**Table 17 : Use Case table of Create Habit**

| | |
|---|---|
| **Use Case ID:** | **UC-11** |
| **Use Case Name:** | Create Habit |
| **Actors:** | User |
| **Description:** | Allows the user to set goals for habits they want to cultivate within the "Productivity App." |
| **Trigger:** | User selects the "Habit Tracker" option within the app. |
| **Preconditions:** | The user has logged into the app.<br>The app has a habit Tracker feature. |
| **Postconditions:** | The user has set habit and can track their progress over time. |
| **Normal Flow:** | 1. User selects "Set Goal" within the app.<br>2. User sets habits they want to achieve.<br>3.The app records and displays the set goals for the user. |
| **Alternative Flows:** | None. |
| **Exceptions:** | Network issues or server unavailability may affect the goal-setting functionality. |
| **Business Rules:** | Habit must be measurable |
| **Assumptions:** | Users have a clear understanding of the habits they want to cultivate. |

Table 18 : Use Case table of Track Habits

| | |
|---|---|
| **Use Case ID:** | **UC-12** |
| **Use Case Name:** | Track Habits |
| **Actors:** | User |
| **Description:** | Allows the user to track their progress for building and maintaining habits within the "Productivity App." |
| **Trigger:** | User selects the "Track Habits" option within the app. |
| **Preconditions:** | The user has logged into the app. - The app has a habit tracking feature. |
| **Postconditions:** | The user has set habit goals and can track their progress over time. |
| **Normal Flow:** | 1. User selects "Track Habits" within the app.<br>2. User tracks their progress by marking completed habits.<br>3. The app records and displays progress over time. |
| **Alternative Flows:** | None. |
| **Exceptions:** | Network issues or server unavailability may affect the habit tracking functionality. |
| **Business Rules:** | Progress tracking is visual and informative. |
| **Assumptions:** | The user is using the habit tracking feature for personal improvement. - The habit tracking process is secure and follows best practices for goal setting and progress tracking. |

Table 19 : Use Case table of Pomodoro Timer

| | |
|---|---|
| **Use Case ID:** | **UC-13** |
| **Use Case Name:** | Pomodoro Timer |
| **Actors:** | User |
| **Description:** | Allows the user to utilize a Pomodoro timer within the "Productivity App" to set work intervals (Pomodoro) and take short and long breaks for effective time management. |
| **Trigger:** | User selects the "Pomodoro Timer" option within the app. |
| **Preconditions:** | The user has logged into the app.<br>The app has a Pomodoro timer feature. |
| **Postconditions:** | The user has completed the Pomodoro session and breaks as configured. |
| **Normal Flow:** | 1. User selects "Pomodoro Timer" within the app.<br>2. The app provides options to set Pomodoro<br>3. User customizes the timer settings.<br>4. User starts the Pomodoro timer.<br>5. The app counts down the Pomodoro session.<br>6. When the Pomodoro ends, the app notifies the user to take a short break.<br>7. After the short break, the app notifies the user to start work session again<br>8. The app resets the timer for the next Pomodoro session. |
| **Alternative Flows:** | None. |
| **Exceptions:** | Network issues or server unavailability may affect the Pomodoro timer functionality. |

|  |  |
|---|---|
| **Business Rules:** | Timer settings are customizable by the user. |
| **Assumptions:** | The user is utilizing the Pomodoro timer as a time management technique. |

**Table 20 : Use Case table of Use AI powered Assistant**

|  |  |
|---|---|
| **Use Case ID:** | **UC-14** |
| **Use Case Name:** | Use AI powered Assistant |
| **Actors:** | User |
| **Description:** | Allows the user to use the AI powered Chatbot within the "Productivity App" for text-based interactions, information retrieval, and assistance. |
| **Trigger:** | User selects the "AI Chatbot" option within the app. |
| **Preconditions:** | The user has logged into the app.<br>The app has integrated the Gemini API. |
| **Postconditions:** | The user has successfully used the Gemini for text-based interactions. |
| **Normal Flow:** | 1. User selects "AI Chatbot" within the app.<br>2. The app provides a text input interface for the user.<br>3. User inputs questions, requests, or queries.<br>4. The app sends the user's input to the Gemini API.<br>5. The Gemini API processes the input and provides responses.<br>6. The app displays Gemini responses to the user. |
| **Alternative Flows:** | None. |
| **Exceptions:** | Network issues or API unavailability may affect the successful use of the Gemini API. |
| **Business Rules:** | The Gemini API is integrated for user assistance and information retrieval. |
| **Assumptions:** | The user is using the Gemini API for various text-based interactions.<br>The Gemini API integration follows best practices for text input and response handling. |

## 3.3 Functional Requirements.

**Task Manager:**

- Users should be able to create, edit, and delete to-do lists with associated tasks.
- Users can set reminders for individual tasks or entire to-do lists.
- Users can track the completion status and progress of tasks.
- Users should be able to mark tasks as completed and view a history of completed tasks.

**Focus Mode:**

- Users can activate a focus mode that blocks distractions, limits, and creates a focused work environment.
- The app should suppress non-essential notifications during Focus Mode.

**Calendar:**

- Users can schedule appointments and events on the calendar.
- Users can set reminders for calendar events.

**Pomodoro Timer:**

- Users can set a timer for a specific work session duration.
- The app should provide breaks between work sessions during Pomodoro mode.

**Note-taking:**

- Users can take notes in text and image formats.
- The app should provide options to take assistance from chat bot

**Habit Tracker:**

- Users can set goals for habits they want to cultivate.
- The app should visually display progress towards habit goals.

**Analytics:**

- Users can access analytics to monitor productivity trends.
- Analytics should include information on the number of tasks completed.
- Users can view the time spent on tasks.

**Integration with Other Productivity Tools:**

- The app shall incorporate the Gemini API, allowing users to interact with the advanced natural language processing capabilities of Google Gemini Chatbot directly within the application.

### 3.3.1 Functional Requirement X

**Table 21 : FR table Description of Create Task**

| **Identifier** | **FR-1** |
|---|---|
| **Title** | Task Manager: Create Task |
| **Requirement** | Users should be able to create, edit, and delete to-do lists with associated tasks. |
| **Source** | User Perspective |
| **Rationale** | To enable users to organize their tasks efficiently and manage their workload effectively. |
| **Business Rule** | Each to-do list must have a unique name. |
| **Dependencies** | FR2, FR3,FR4 |
| **Priority** | High |

Table 22 : FR Table Description of Get Reminders

| **Identifier** | **FR-2** |
|---|---|
| **Title** | Task Manager: Get Reminders |
| **Requirement** | Users get reminders for tasks |
| **Source** | User Perspective |
| **Rationale** | To help users stay organized and on schedule by providing timely reminders for their tasks. |
| **Business Rule** | Reminders must be set in advance by a minimum of 1 minutes. |
| **Dependencies** | |
| **Priority** | Medium |

Table 23 : FR Table Description of Tracking Progress

| **Identifier** | **FR-3** |
|---|---|
| **Title** | Task Manager: Track Progress |
| **Requirement** | Users can track the completion status and progress of tasks. |
| **Source** | User Perspective |
| **Rationale** | To provide users with insights into their task management and productivity. |
| **Business Rule** | Progress tracking is based on the completion percentage of each task. |
| **Dependencies** | |
| **Priority** | Medium |

Table 24 : FR Table Description of  Mark Tasks as Completed

| **Identifier** | **FR-4** |
|---|---|
| **Title** | Task Manager: Mark Tasks as Completed |
| **Requirement** | Users should be able to mark tasks as completed and view a history of completed tasks. |
| **Source** | User Perspective |
| **Rationale** | To help users keep a record of completed tasks and track their overall productivity. |
| **Business Rule** | Completed tasks cannot be edited or deleted. |
| **Dependencies** | |
| **Priority** | High |

Table 25 : FR Table Description of Activate Focus Mode

| **Identifier** | **FR-5** |
|---|---|
| **Title** | Focus Mode: Activate Focus Mode |
| **Requirement** | Users can activate a focus mode that blocks distractions, limits app usage, and creates a focused work environment. |
| **Source** | User Perspective |
| **Rationale** | To help users concentrate on their work by minimizing distractions and creating an environment conducive to focus. |
| **Business Rule** | Focus Mode cannot be activated without user granting permission. |

| | |
|---|---|
| **Dependencies** | FR-6 |
| **Priority** | High |

**Table 26 : FR Table Description of Deactivate Focus Mode**

| | |
|---|---|
| **Identifier** | **FR-6** |
| **Title** | Focus Mode: Deactivate Focus Mode |
| **Requirement** | Users can deactivate a focus mode by turning it off on the focus mode screen |
| **Source** | User Perspective |
| **Rationale** | User can deactivate focus mode any time |
| **Business Rule** | Focus Mode cannot be deactivated without user consent |
| **Dependencies** | FR7, FR8 |
| **Priority** | High |

**Table 27 : FR Table Description of Schedule Events**

| | |
|---|---|
| **Identifier** | **FR-7** |
| **Title** | Calendar: Schedule Events |
| **Requirement** | Users can schedule appointments and events on the calendar. |
| **Source** | User Perspective |
| **Rationale** | To help users manage their time effectively by scheduling and organizing their tasks and Events. |
| **Business Rule** | Each calendar event must have a valid date and time that is . |
| **Dependencies** | FR-8 |
| **Priority** | High |

**Table 28 : FR Table Description of View Events**

| | |
|---|---|
| **Identifier** | **FR-8** |
| **Title** | View Upcoming Events |
| **Requirement** | Users should be able to view upcoming events for any dates they choose. |
| **Source** | |
| **Rationale** | To keep users informed about their upcoming schedule and help them plan accordingly. |
| **Business Rule** | Events must be displayed based on the user's selected date. |
| **Dependencies** | |
| **Priority** | High |

**Table 29 : FR Table Description of Set Pomodoro Timer**

| | |
|---|---|
| **Identifier** | **FR-9** |
| **Title** | Set Pomodoro Timer |
| **Requirement** | Users can set a timer for a specific work session duration. |
| **Source** | User Need to stay productive |
| **Rationale** | To implement the Pomodoro technique, helping users manage work sessions and breaks effectively. |
| **Business Rule** | Each Work session is followed by a break Session |

| | |
|---|---|
| **Dependencies** | FR-10 |
| **Priority** | High |

**Table 30 : FR Table Description of  Provide Breaks**

| | |
|---|---|
| **Identifier** | **FR-10** |
| **Title** | Pomodoro Timer: Provide Breaks |
| **Requirement** | The app should provide breaks between work sessions during Pomodoro mode. |
| **Source** | User Perspective |
| **Rationale** | To ensure users have designated breaks for relaxation and recharging during work sessions. |
| **Business Rule** | Breaks must come after work session |
| **Dependencies** | |
| **Priority** | Medium |

**Table 31 : FR Table Description of Taking Notes in Various Formats**

| | |
|---|---|
| **Identifier** | **FR-11** |
| **Title** | Note-taking: Note in Various Formats |
| **Requirement** | Users can take notes in text and image |
| **Source** | User Perspective |
| **Rationale** | To provide flexibility in capturing thoughts, ideas, and information in different formats. |
| **Business Rule** | Image notes must be in standard image formats |
| **Dependencies** | FR-12 |
| **Priority** | High |

**Table 32 : FR Table  Description of Edit, Delete Notes**

| | |
|---|---|
| **Identifier** | **FR-12** |
| **Title** | Edit and Delete Notes |
| **Requirement** | Users should have the ability to edit and delete notes. |
| **Source** | |
| **Rationale** | To allow users to manage their notes effectively and make necessary changes as needed. |
| **Business Rule** | Deleted and Edited notes are deleted properly |
| **Dependencies** | |
| **Priority** | Medium |

**Table 33 : FR Table  Description of Setting Goals**

| | |
|---|---|
| **Identifier** | **FR-13** |
| **Title** | Habit Tracker: Set Goals |
| **Requirement** | Users can set goals for habits they want to cultivate. |
| **Source** | User Perspective |
| **Rationale** | To enable users to define and work towards cultivating positive habits. |

| | |
|---|---|
| **Business Rule** | Each habit goal must have a defined success criterion. |
| **Dependencies** | FR14 |
| **Priority** | High |

**Table 34 : FR Table Description of Tracking Habit Progress**

| | |
|---|---|
| **Identifier** | **FR-14** |
| **Title** | Habit Tracker: Track Progress |
| **Requirement** | The app should visually display progress towards habit goals. |
| **Source** | User Perspective |
| **Rationale** | To motivate users by visually showcasing their progress in cultivating positive habits. |
| **Business Rule** | Progress tracking is based on daily completion of habit-related tasks. |
| **Dependencies** | |
| **Priority** | Medium |

**Table 35 : FR Table Description of Integrating with Gemini**

| | |
|---|---|
| **Identifier** | **FR-15** |
| **Title** | Integration with Other Productivity Tools: Integrate with Gemini |
| **Requirement** | The app shall incorporate the Gemini API, allowing users to interact with the advanced natural language processing capabilities of Gemini directly within the application. |
| **Source** | User Perspective |
| **Rationale** | To leverage Gemini's natural language processing for enhanced user interaction and assistance within the app. |
| **Business Rule** | Gemini integration is limited to text-based interactions within the app. |
| **Dependencies** | |
| **Priority** | Low |

## 3.4 Non-Functional Requirements

### 3.4.1 Reliability

Ensuring consistency within the productivity app is crucial due to the significance of data integrity and the potential repercussions of inaccuracies. Failure to save notes, inability to mark tasks as completed, and providing inaccurate analytics can disrupt user experience and undermine the app's reliability. Therefore, maintaining consistent data management processes is essential to prevent such issues and uphold user trust and satisfaction.

### 3.4.2 Usability

The productivity app's UI must be intuitive and easy to navigate.
*USE-1: This includes clear structures and straightforward task management processes.*
*USE-2: A visually appealing design with consistent layout and color schemes enhances usability.*
*USE-4: Compatibility with various screen sizes and devices ensures accessibility for all users.*

### 3.4.3 Performance

*PER-1: The Productivity App shall allow users to create a new task in less than 10 seconds, ensuring swift task entry for a seamless user experience.*
*PER-2: The calendar module shall load events and appointments within 10 seconds, providing users with quick access to their schedule.*
*PER-3: The app shall activate the Focus Mode within 5 second of user initiation, blocking distractions promptly to enhance user productivity.*
*PER-4: Analytics data, including task completion statistics, time spent, and productivity trends, shall be processed and displayed within 10 seconds of the user's request.*
*PER-5: The Pomodoro Timer shall start and provide breaks according to the user's settings without any delay, ensuring precise time management.*
*PER-6: Visual feedback from the Habit Tracker, indicating progress and achievements, shall be updated in real-time, providing users with immediate acknowledgment.*
*PER-7: The app's integration with external productivity tools, such as Gemini shall have a response time of less than 15 seconds for data retrieval and synchronization.*
*PER-8: The app shall allow users to take and save notes, including text and images within 5 seconds of initiating the process.*
*PER-9: The app shall maintain an overall responsiveness, with interactions and transitions responding within 5 second to provide a smooth user interface.*

### 3.4.4 Security

For the Productivity App, security is a non-functional criterion. All the data of the user is stored locally making it more secure, So no outsider can access user data .

## 3.5 External Interface Requirements

### 3.5.1 User Interfaces Requirements

- GUI Standards and Style Guides: material UI for android
- Font-Family: Ubuntu font family
- Screen Layout and Portrait.

### 3.5.2 Software interfaces

*SI -1: The app will be compatible with Android operating systems, specifically targeting Android 10 and above*

*SI-2: Calendar API is used for integrating calendar functionality, allowing users to schedule appointments and events.*

*SI-3: Firebase is utilized for user authentication and sign-in processes.*

*SI-4: Local Storage is used to store data.*

*SI-5: Integration with Google Gemini API for providing chat-based assistance and information retrieval within the app.*

### 3.5.3 Hardware interfaces

*HI-1: Supported device types include those running iOS and Android operating systems.*

*HI-2: Data interactions involve the storage and retrieval of app data from the device's storage system, including tasks, notes, and settings.*

*HI-3: Control interactions include user input via device touchscreens, keyboards, and mice, as well as output through displays and audio speakers.*

### 3.5.4 Communications interfaces

*CI-1. The app will send user request to the  Gemini API using internet and display the response to the user .*
*CI-2: All the data is stored locally on the mobile phone and will be fetched locally.*

# 4 Design and Architecture

## 4.1 Architectural Design

The box-and-line diagram provides a visual representation of the architecture and interconnections of the productivity app's modules. Each box represents a distinct module or feature, while the lines denote the interactions and dependencies between them.

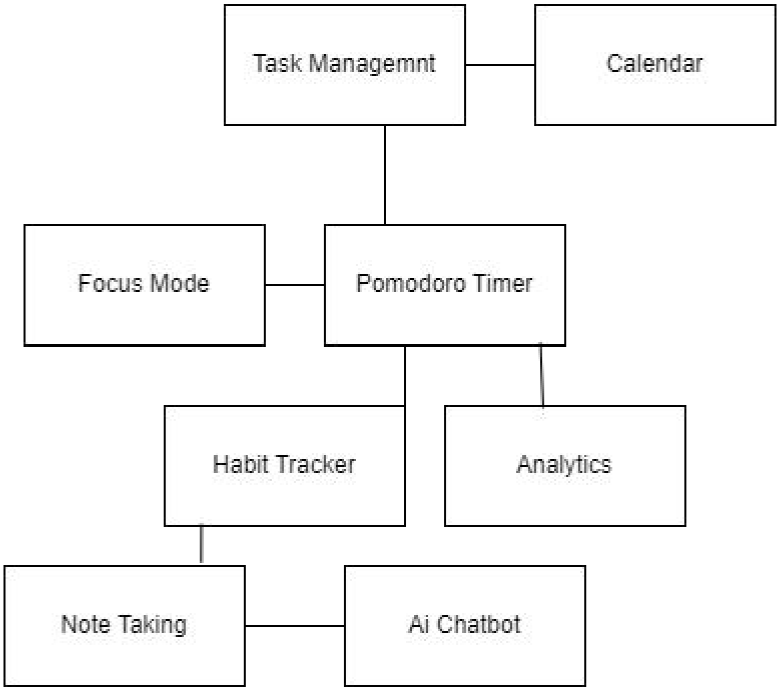


**Figure 2 : Box-Line diagram**

Our app's architecture is designed with a modular approach, prioritizing code encapsulation and flexibility. Instead of following a strict MVC pattern, each module operates independently, allowing for easier maintenance and scalability. Widgets and components are shared across modules for consistency, while module-specific elements remain confined, reducing dependencies and enhancing code reusability. This structure streamlines development, ensuring efficient integration of new features and updates.

## 4.2 Design Models

The activity diagram illustrates the seamless flow of interactions within the productivity app, depicting how users navigate through various features and functionalities. From task management to habit tracking, note-taking, and more, the diagram showcases the app's comprehensive capabilities in enhancing productivity and organization

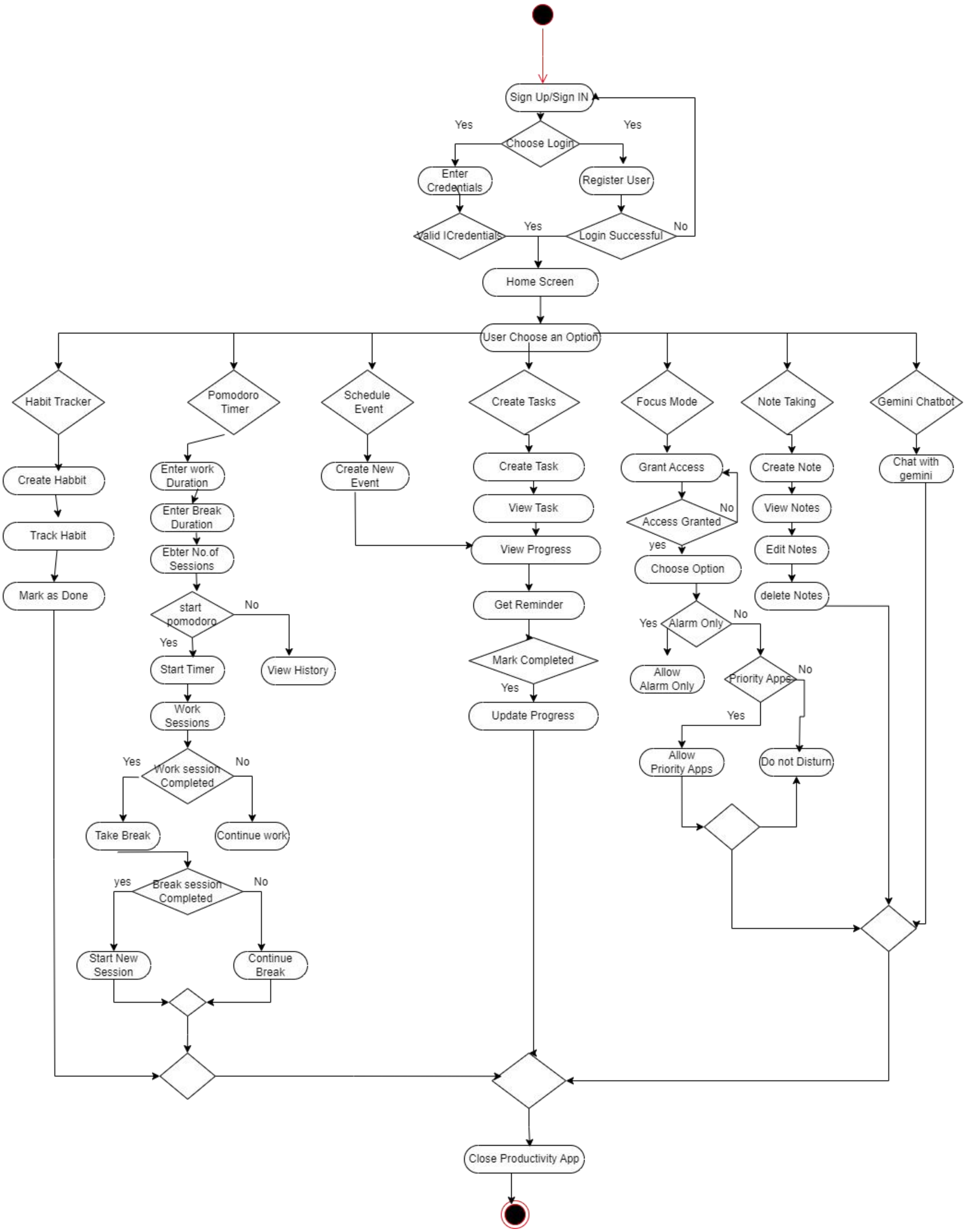


**Figure 3 :Activity Diagram**

- **Login Sequence Diagram**

The login sequence diagram illustrates the flow of interactions between the user and the productivity app during the login process. It captures the steps involved in user authentication, submitting them for verification, and receiving a response from the app.

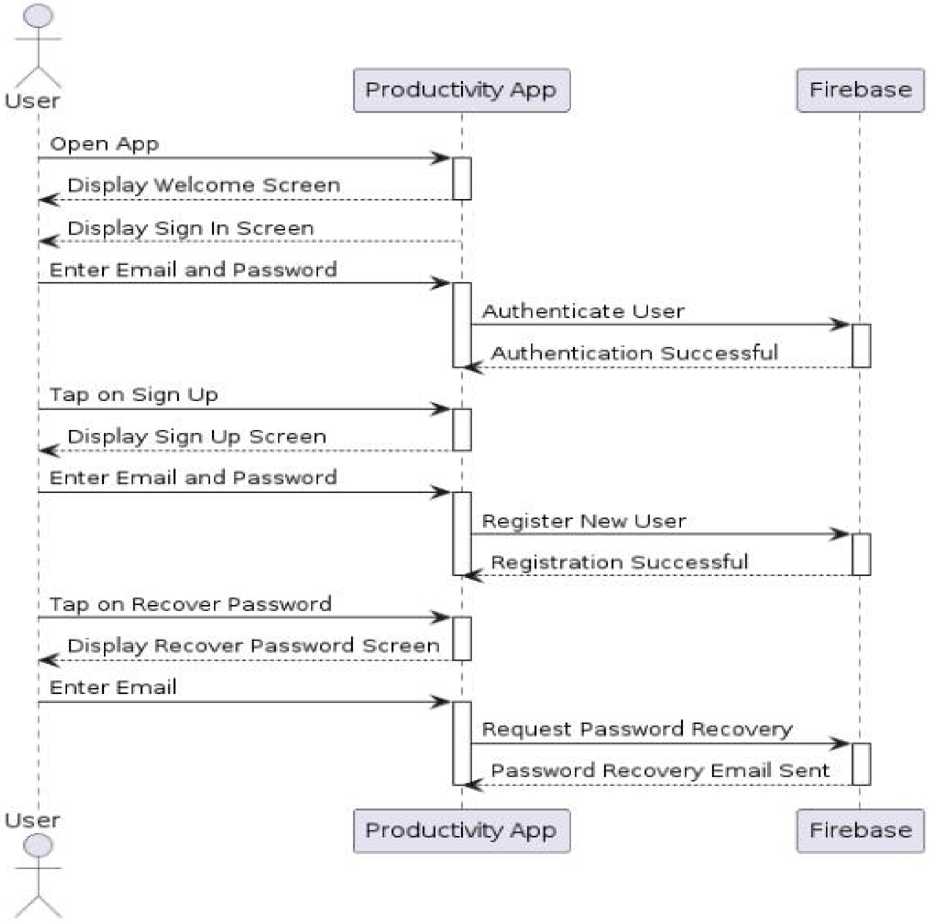


**Figure 4 :Login Sequence Diagram**

- **Focus Mode  Sequence Diagram**

The Focus Mode sequence diagram depicts the steps involved when a user activates the Focus Mode feature within the productivity app.This diagram provides insight into how the app requests permission from the user, processes their response, and configures the Focus Mode settings accordingly.

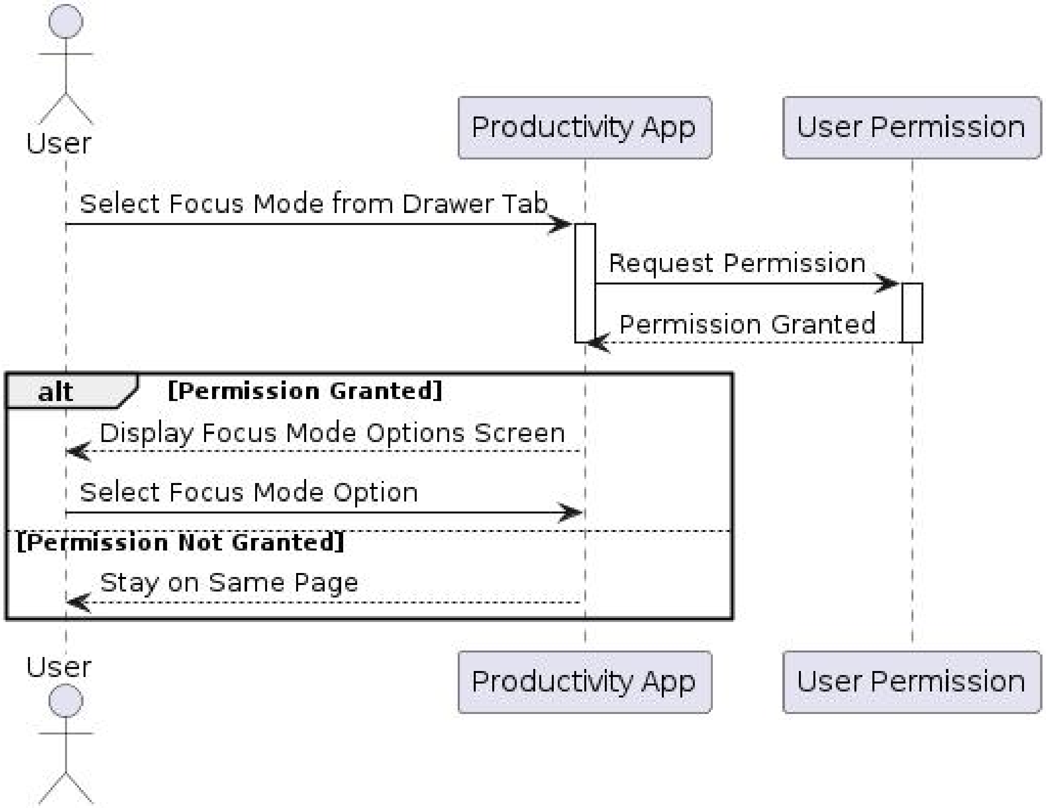


**Figure 5 : Focus Mode Sequence Diagram**

- **Task Management Sequence Diagram**

The Task Management Sequence Diagram illustrates the sequence of interactions between the user and the productivity app when performing various tasks management operations. It showcases how users can create, update, delete tasks, and mark them as complete. Additionally, it demonstrates how the app processes these actions and updates the task management database accordingly.

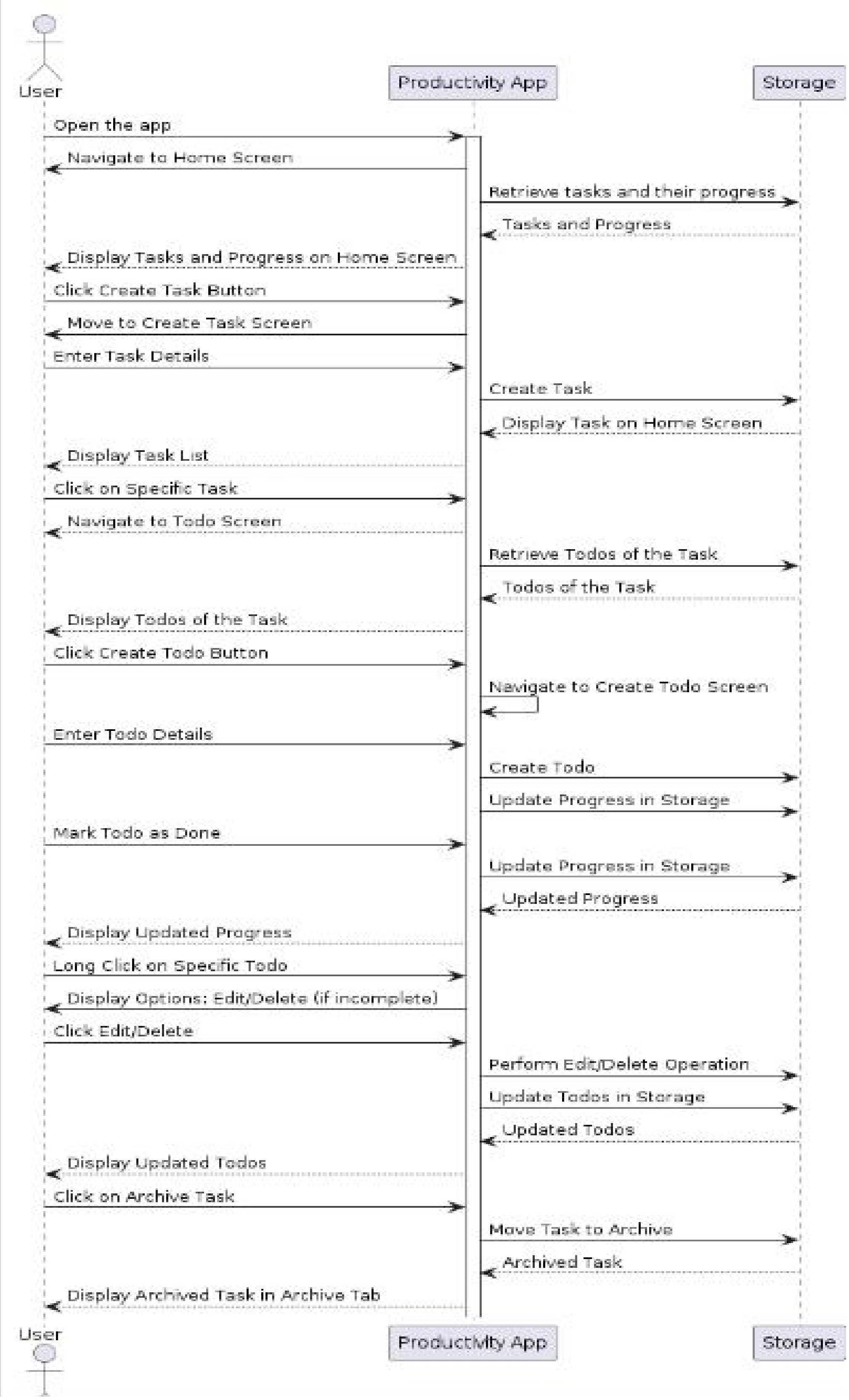


**Figure 6 : Task Management Sequence Diagram**

- **Note Taking Sequence Diagram**

The Note Taking Sequence Diagram depicts the flow of interactions between the user and the productivity app during note-taking activities. It outlines how users can create, view, edit, and delete notes of various types, including text, images. Additionally, it illustrates how these notes are stored and managed within the app's storage system.

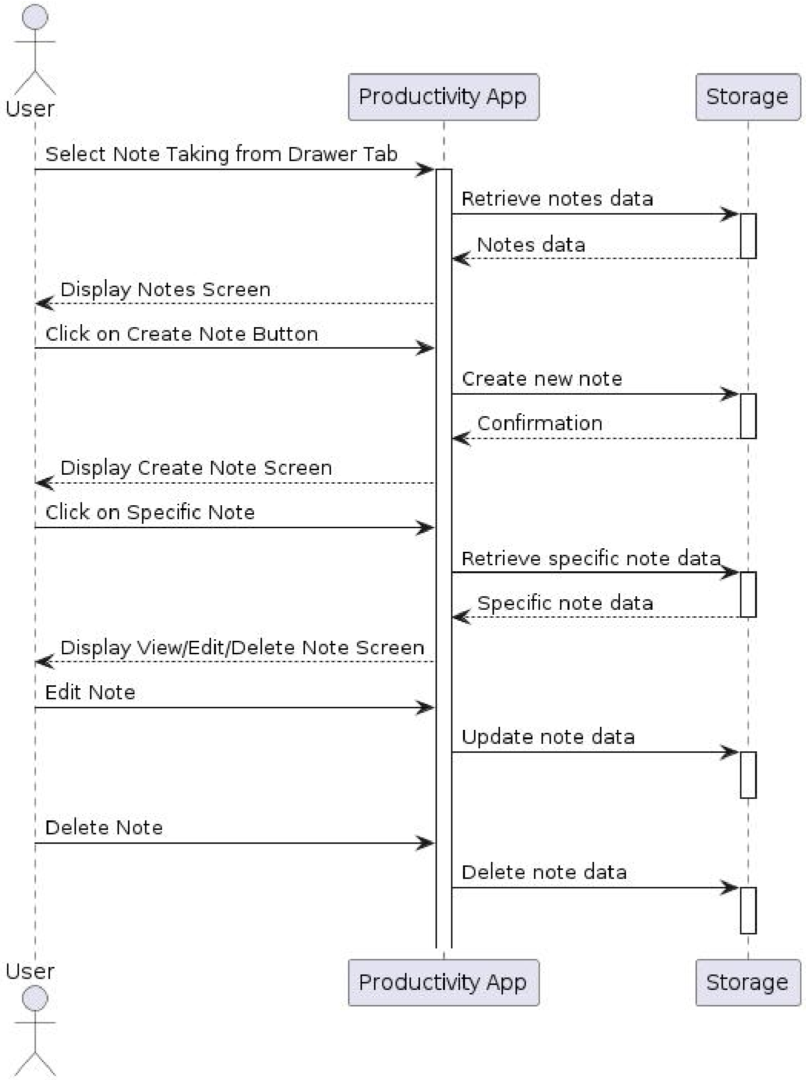


**Figure 7 : Note Taking Sequence diagram**

- **Pomodoro Sequence Diagram**

The diagram showcases the process of inputting work duration, break duration, and the number of sessions. It also illustrates the transition between work and break sessions, the completion of Pomodoro cycles, and the management of analytics related to session durations and productivity metrics.

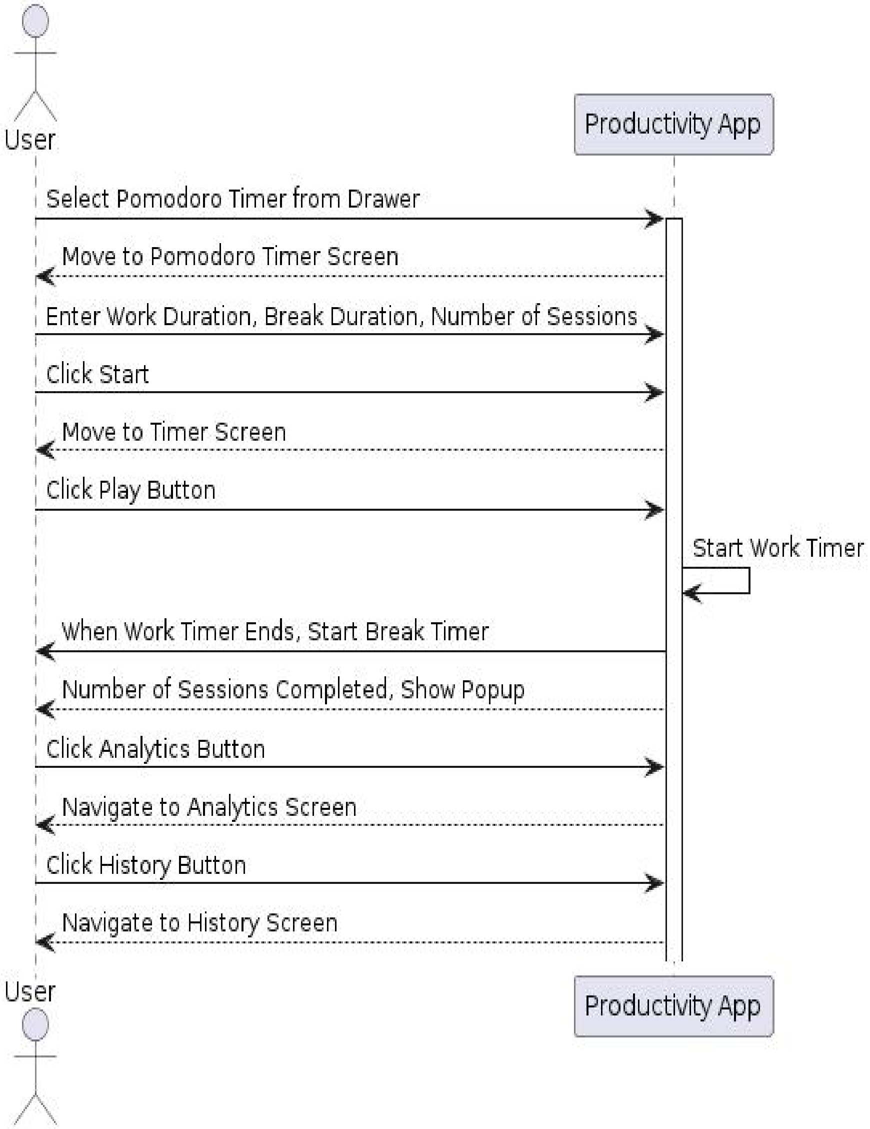

User
Productivity App
Select Pomodoro Timer from Drawer
Move to Pomodoro Timer Screen
Enter Work Duration, Break Duration, Number of Sessions
Click Start
Move to Timer Screen
Click Play Button
Start Work Timer
When Work Timer Ends, Start Break Timer
Number of Sessions Completed, Show Popup
Click Analytics Button
Navigate to Analytics Screen
Click History Button
Navigate to History Screen
User
Productivity App

**Figure 8 : Pomodoro Sequence Diagram**

- **Calendar Sequence Diagram**

The diagram captures the steps involved in selecting specific dates, viewing event details, marking events as done, and managing event notifications. Additionally, it highlights the communication between the user interface and the underlying data storage system to ensure accurate event management and synchronization across devices.

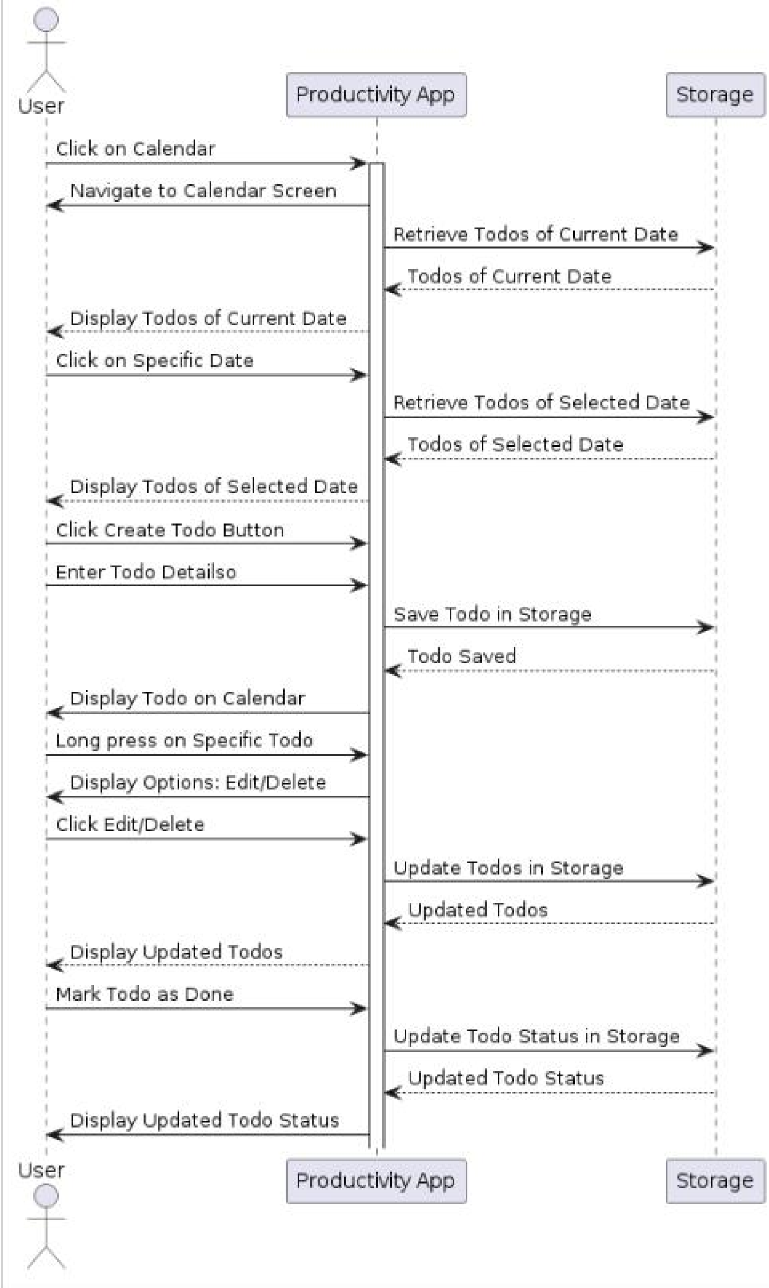


**Figure 9 : Calendar Sequence Diagram**

- **Habit tracker Sequence Diagram**

The Habit Tracker Sequence Diagram depicts the user's interactions with the app's habit tracking functionality. It illustrates how users can view existing habits, mark them as done, and create new habits.

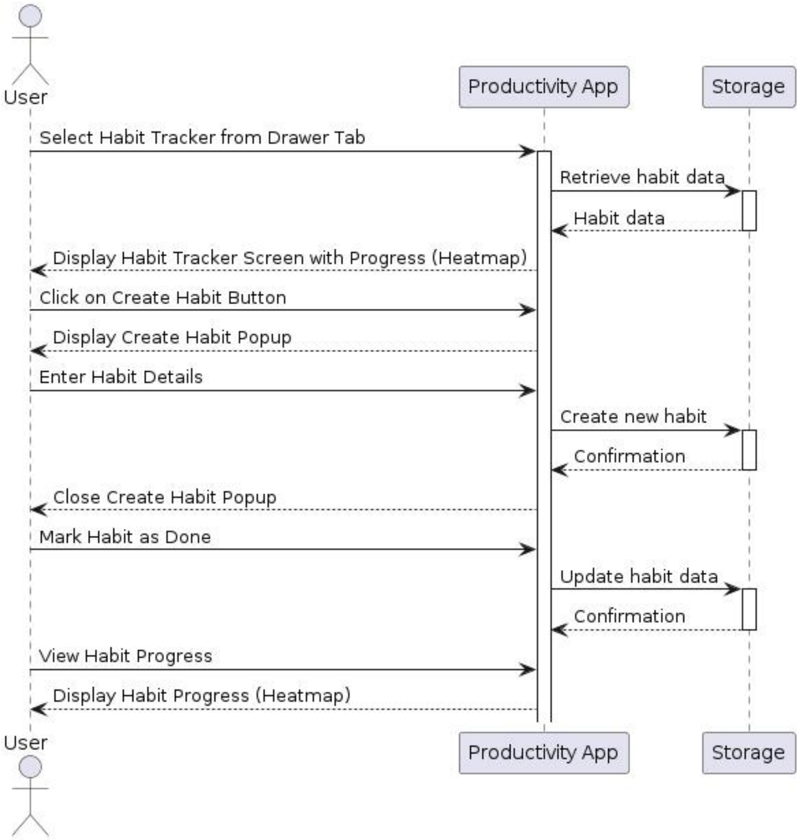


**Figure 10 : Habit tracker Sequence Diagram**

- **Chatbot Sequence Diagram**

The Chatbot Sequence Diagram illustrates the interaction between the user and the chatbot feature of the app. It outlines the steps involved in initiating a conversation with the chatbot, sending a message or query, and receiving responses.

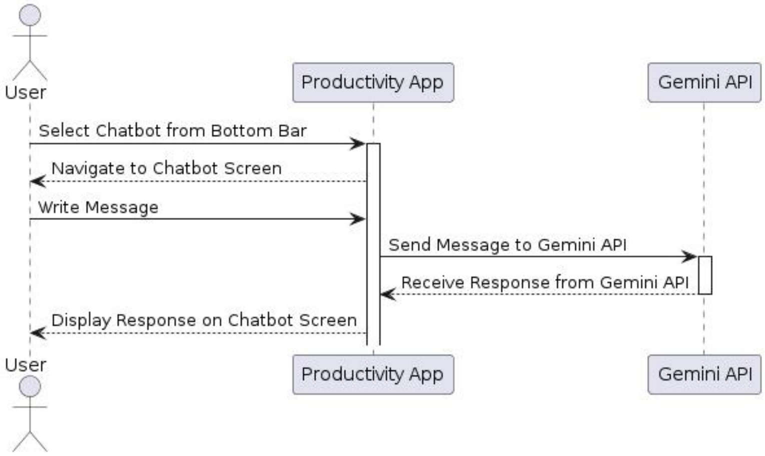

User
Productivity App
Gemini API
Select Chatbot from Bottom Bar
Navigate to Chatbot Screen
Write Message
Send Message to Gemini API
Receive Response from Gemini API
Display Response on Chatbot Screen
User
Productivity App
Gemini API

**Figure 11 : Chatbot Sequence Diagram**

- **Class diagram**

The Class Diagram provides a visual representation of the structure and relationships of the classes within the productivity app. It showcases the various entities and their attributes, methods, and associations

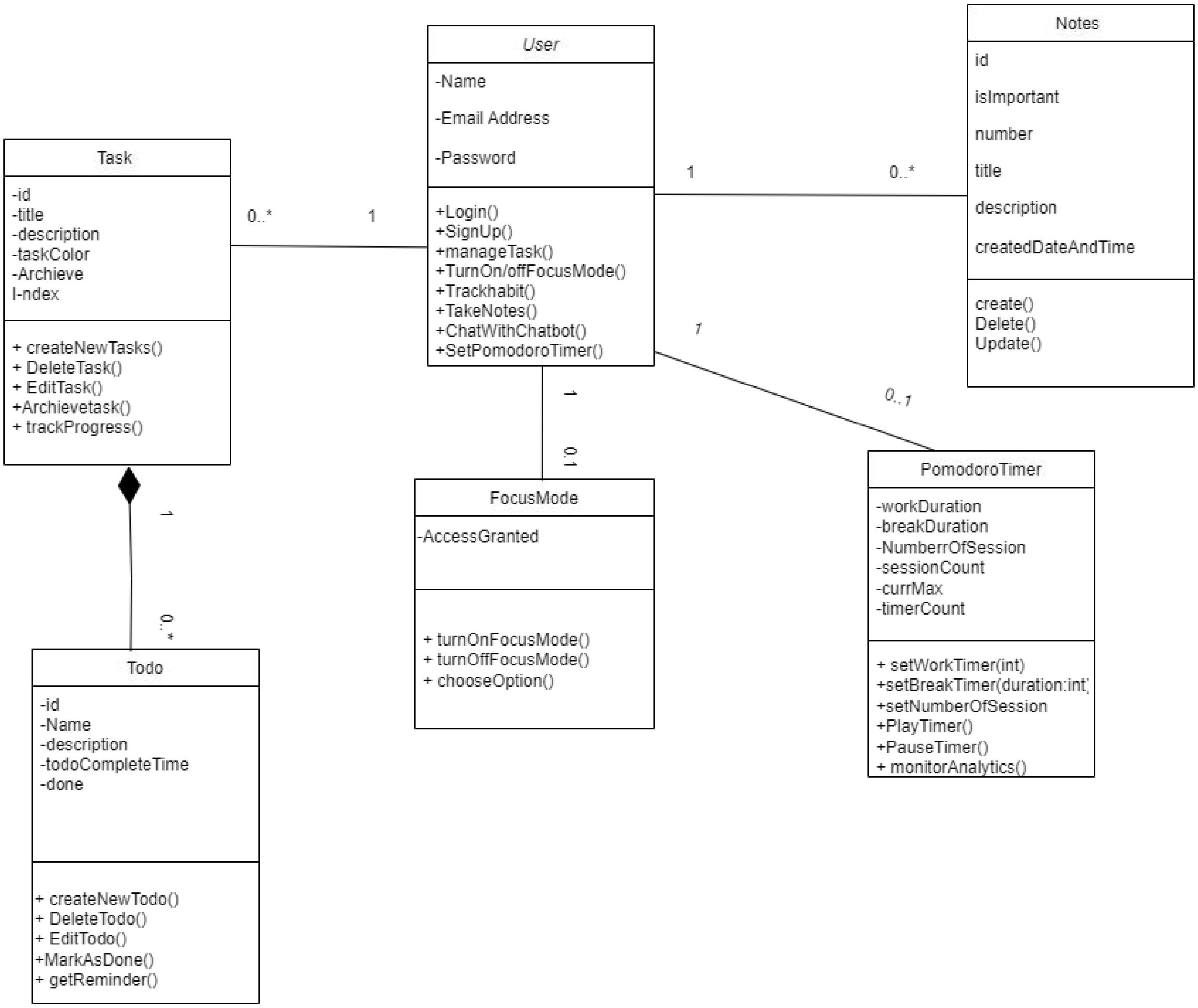


**Figure 12 : Class Diagram**

## 4.3 Data Design

The information domain of the productivity app encompasses various entities such as tasks, notes,events,habits, and user preferences. These entities are transformed into data structures within the app's database to facilitate efficient storage, processing, and organization.
The data will be stored locally on the user phone and user details are stored on Firebase.

### 4.3.1 Data Dictionary

- **User:**

1. **Name:** String, the name of user
2. **Email:** String, the email user wants to use for registration purpose.
3. **Password**: String, the password of user account.

- **Tasks:**

1. **id** (Type: Id) An auto-generated unique identifier assigned to each task entry for database storage and retrieval purposes.
2. **Title** (Type: String) The title or name of the task, providing a concise description of the task's purpose or objective.
3. **Description:**(Type: String) Additional details or information about the task
4. **Task Color**:(Type: Color) The color selected by the user to associate with the task, allowing for visual differentiation or categorization of tasks.
5. **Archive:**(Type: Bool) A Boolean value indicating whether the task has been archived (true) or is currently active (false), providing users with the option to hide completed or less relevant tasks.
6. **Index**:(Type: int) The index or position of a particular task within a list or collection, facilitating sorting or prioritization of tasks based on their order or sequence.

- **Todo**

1. **Id** (Type: Id) A unique identifier assigned to each task, used for database storage and retrieval purposes.
2. **Name** (Type: String) The title or name of the task, providing a concise description of the task's purpose or objective.
3. **description:** (Type: String) Additional details or information about the task, providing context or instructions for completing the task.
4. t**odoCompletedTime:** (Type: Date Time) The timestamp indicating the date and time when the task was marked as completed, used for tracking task completion history.
5. **done**: (Type: bool) A Boolean value indicating whether the task has been completed (true) or is still pending (false).

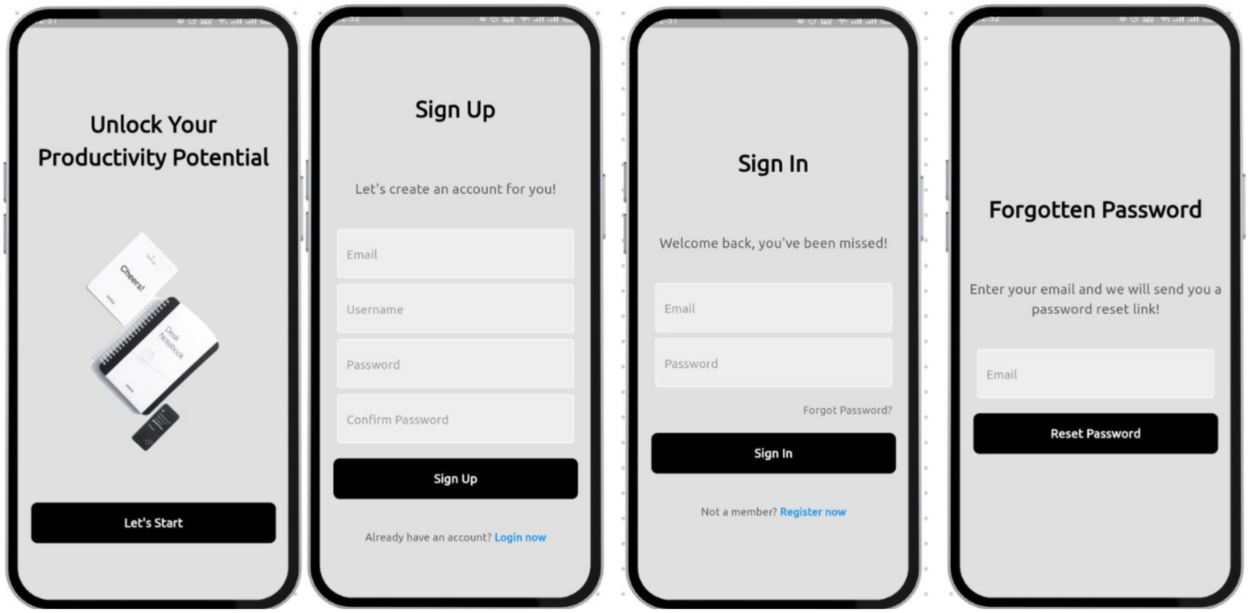
- **Note**

1. **id:** (Type: int?) An optional value representing the unique identifier for each task entry.
2. **is Important:** (Type: bool) A indicator to denote whether the task is marked as important or not.
3. **number:** (Type: int) An integer value representing a numerical identifier or code associated with the task.
4. **title:** (Type: String) A string value representing the title or name of the task.
5. **description:**(Type: String) A string value providing additional details or information about the task.
6. **created Time:** (Type: DateTime) A date and time value indicating the timestamp when the task was created or added to the system.

## 4.4 Human Interface Design

### 4.4.1 Screen Images

- **Sign up/Sign In**



Unlock Your Productivity Potential

Let's Start

Sign Up

Let's create an account for you!

Email

Username

Password

Confirm Password

Sign Up

Already have an account? Login now

Sign In

Welcome back, you've been missed!

Email

Password

Forgot Password?

Sign In

Not a member? Register now

Forgotten Password

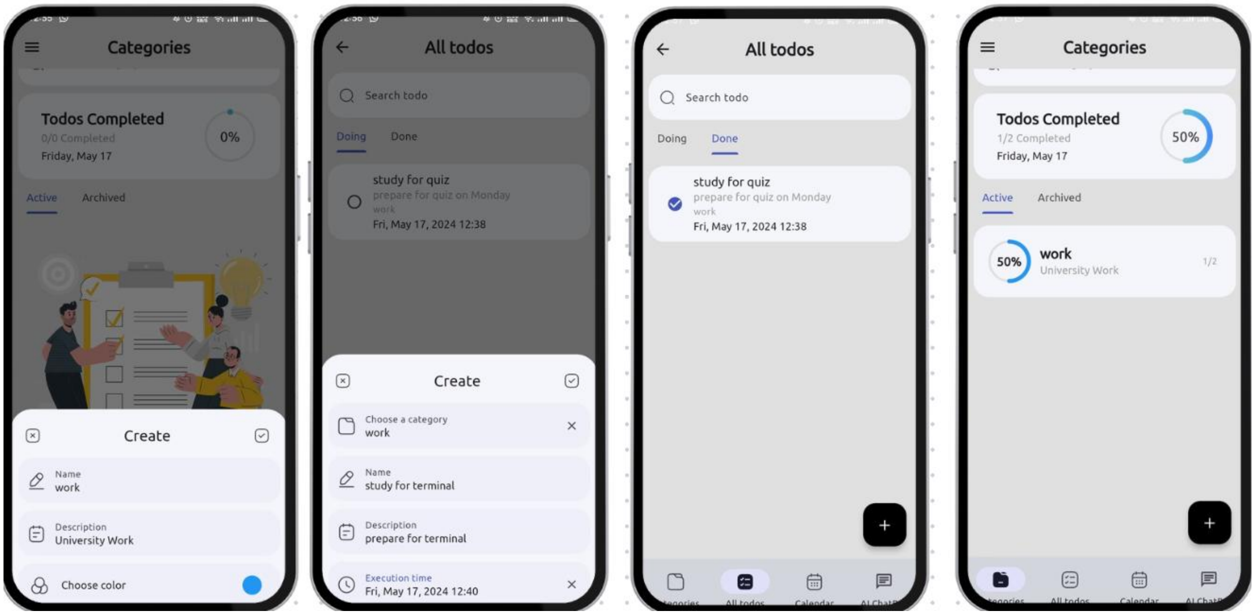
Enter your email and we will send you a password reset link!

Email

Reset Password

**Figure 13 : Interface of Sign Up/Sign in**

- **Tasks Screen**



Categories

Todos Completed

0/0 Completed

Friday, May 17

0%

Active Archived

Create

Name
work

Description
University Work

Choose color

All todos

Search todo

Doing Done

study for quiz
prepare for quiz on Monday
work
Fri, May 17, 2024 12:38

Create

Choose a category
work

Name
study for terminal

Description
prepare for terminal

Execution time
Fri, May 17, 2024 12:40

All todos

Search todo

Doing Done

study for quiz
prepare for quiz on Monday
work
Fri, May 17, 2024 12:38

Categories

Todos Completed

1/2 Completed

Friday, May 17

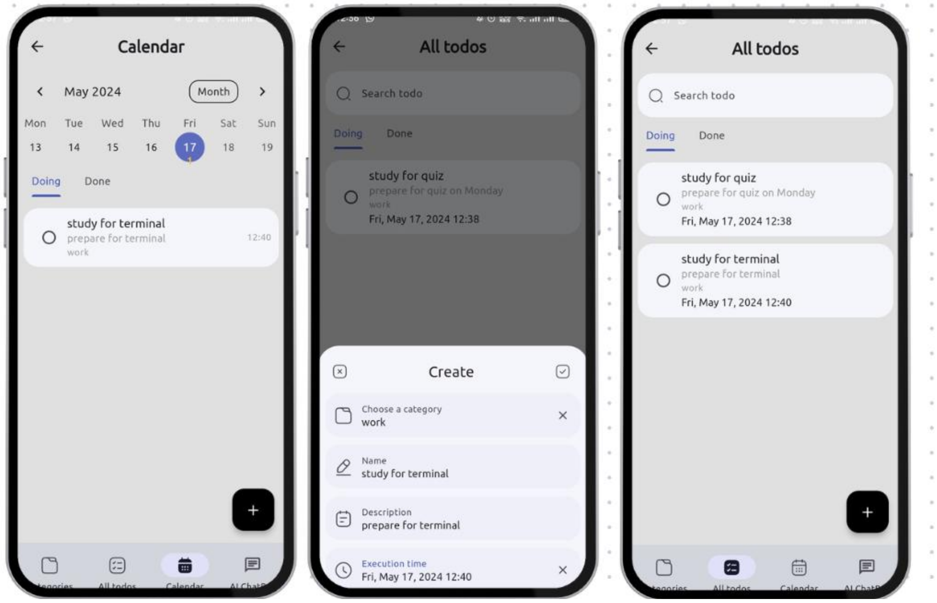
50%

Active Archived

50% work
University Work
1/2

**Figure 14 : Interface of Task Management**

- **Calendar**



study for terminal
prepare for terminal
work
12:40

study for quiz
prepare for quiz on Monday
work
Fri, May 17, 2024 12:38

Choose a category
work

Name
study for terminal

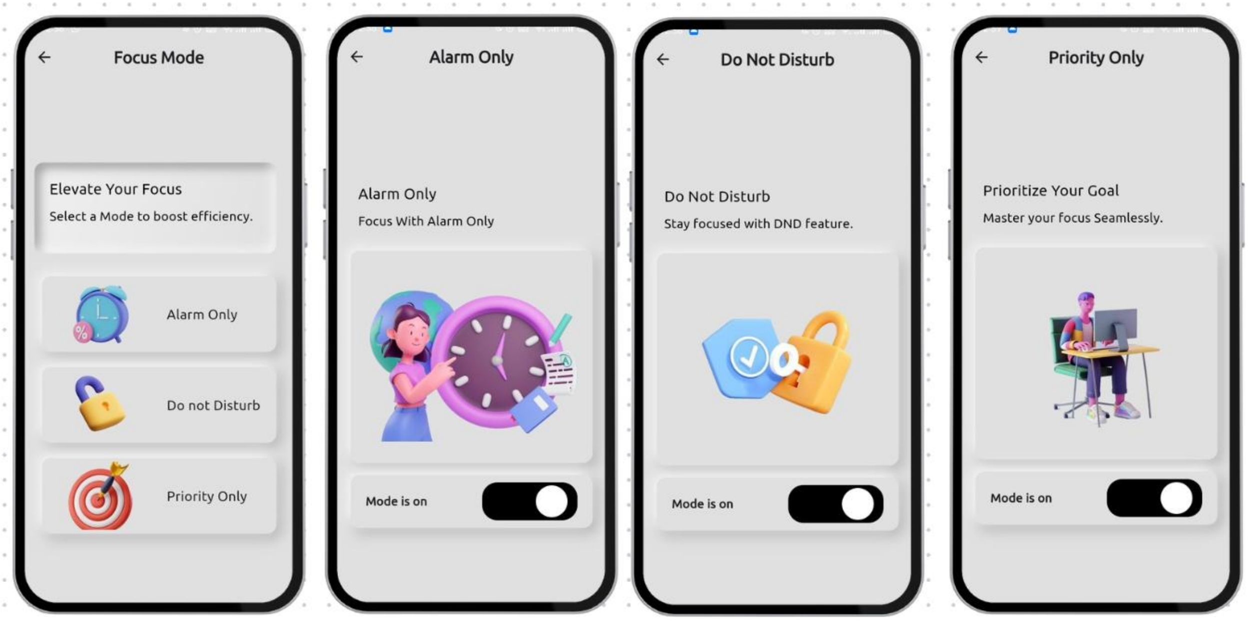
Description
prepare for terminal

Execution time
Fri, May 17, 2024 12:40

study for terminal
prepare for terminal
work
Fri, May 17, 2024 12:40

**Figure 15 : Interface of Calendar Event Scheduling**

- **Focus Mode**



Elevate Your Focus
Select a Mode to boost efficiency.

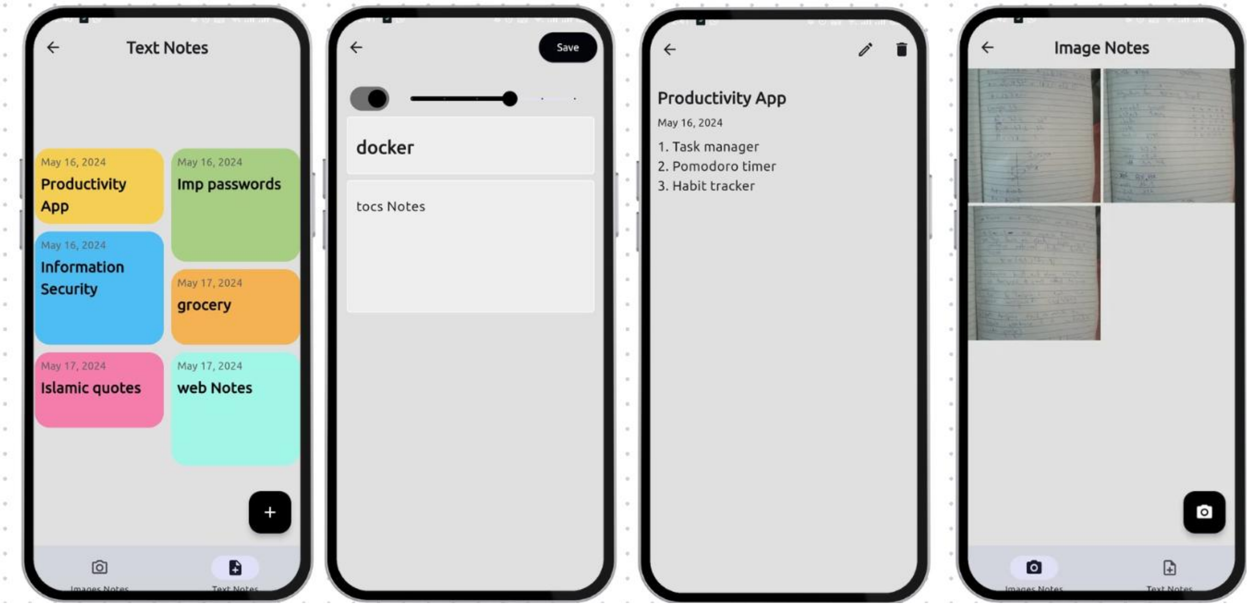
Alarm Only
Focus With Alarm Only
Mode is on

Do Not Disturb
Stay focused with DND feature.
Mode is on

Prioritize Your Goal
Master your focus Seamlessly.
Mode is on

**Figure 16 : Interface of Focus Mode**

- **Note Taking**



May 16, 2024 Productivity App
May 16, 2024 Imp passwords
May 16, 2024 Information Security
May 17, 2024 grocery
May 17, 2024 Islamic quotes
May 17, 2024 web Notes

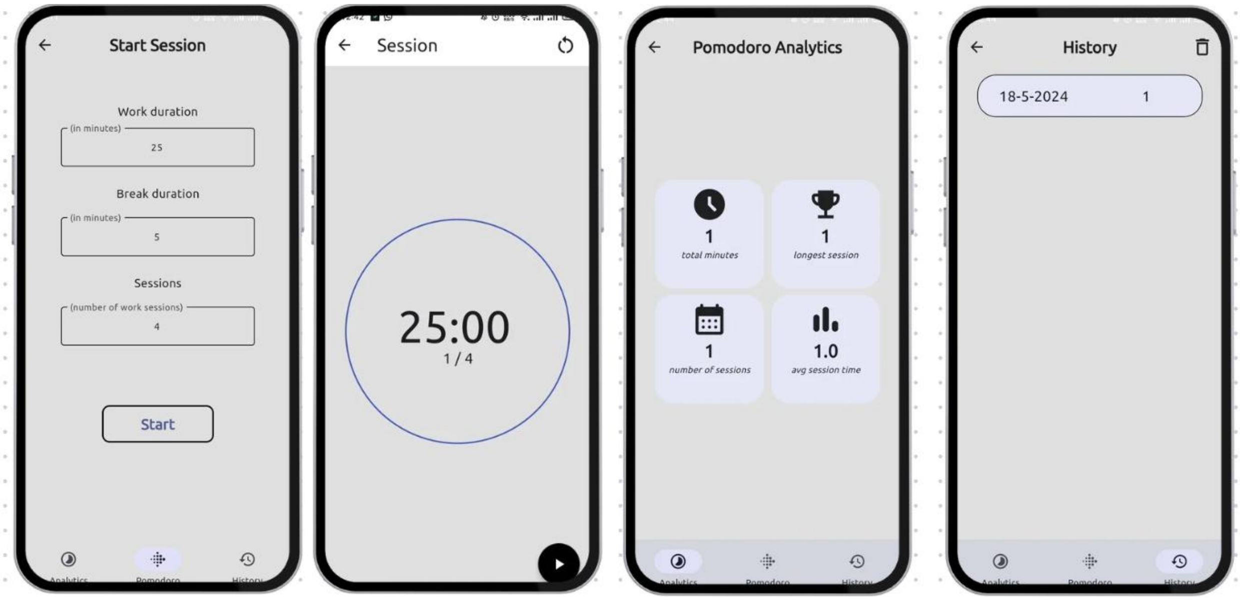
docker

tocs Notes

Productivity App
May 16, 2024
1. Task manager
2. Pomodoro timer
3. Habit tracker

**Figure 17 : Interface of Note Taking**

- **Pomodoro**



Work duration
(in minutes) 25
Break duration
(in minutes) 5
Sessions
(number of work sessions) 4

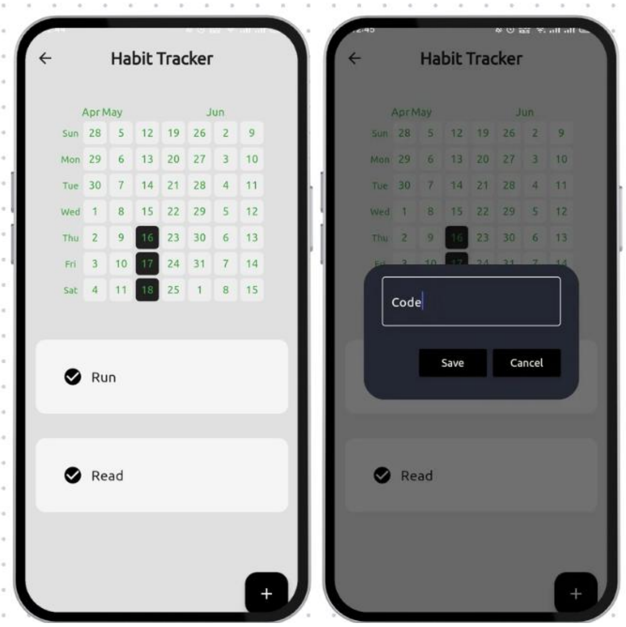
25:00
1 / 4

1 total minutes
1 longest session
1 number of sessions
1.0 avg session time

18-5-2024 1

**Figure 18 : Interface of Pomodoro Timer**

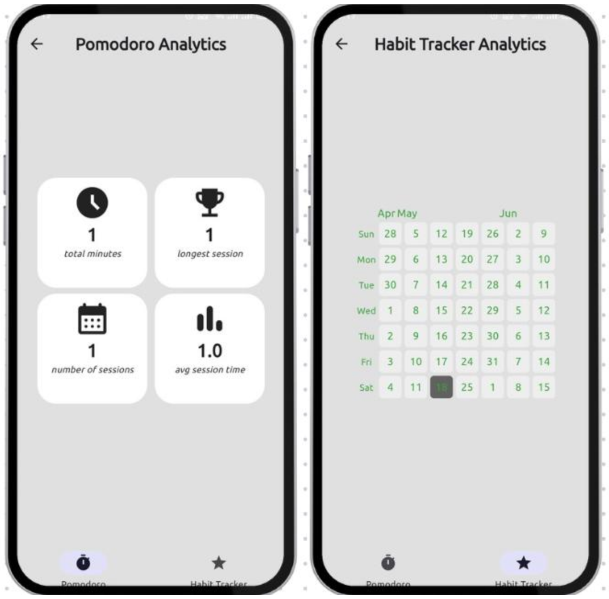
- **Habit Tracker**



| | Apr May | | | | Jun | | |
|---|---|---|---|---|---|---|---|
| Sun | 28 | 5 | 12 | 19 | 26 | 2 | 9 |
| Mon | 29 | 6 | 13 | 20 | 27 | 3 | 10 |
| Tue | 30 | 7 | 14 | 21 | 28 | 4 | 11 |
| Wed | 1 | 8 | 15 | 22 | 29 | 5 | 12 |
| Thu | 2 | 9 | 16 | 23 | 30 | 6 | 13 |
| Fri | 3 | 10 | 17 | 24 | 31 | 7 | 14 |
| Sat | 4 | 11 | 18 | 25 | 1 | 8 | 15 |

**Figure 19 : Interface of Habit Tracker**

- **Analytic**



1 total minutes

1 longest session

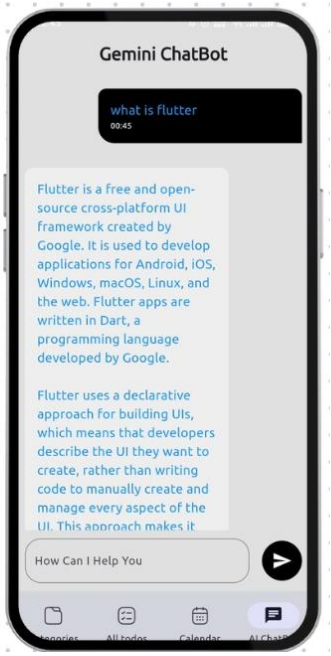
1 number of sessions

1.0 avg session time

| | Apr May | | | | Jun | | |
|---|---|---|---|---|---|---|---|
| Sun | 28 | 5 | 12 | 19 | 26 | 2 | 9 |
| Mon | 29 | 6 | 13 | 20 | 27 | 3 | 10 |
| Tue | 30 | 7 | 14 | 21 | 28 | 4 | 11 |
| Wed | 1 | 8 | 15 | 22 | 29 | 5 | 12 |
| Thu | 2 | 9 | 16 | 23 | 30 | 6 | 13 |
| Fri | 3 | 10 | 17 | 24 | 31 | 7 | 14 |
| Sat | 4 | 11 | 18 | 25 | 1 | 8 | 15 |

**Figure 20 : Interface of Analytics**

- **AI Powered Chat bot**



what is flutter
00:45

Flutter is a free and open-source cross-platform UI framework created by Google. It is used to develop applications for Android, iOS, Windows, macOS, Linux, and the web. Flutter apps are written in Dart, a programming language developed by Google.

Flutter uses a declarative approach for building UIs, which means that developers describe the UI they want to create, rather than writing code to manually create and manage every aspect of the UI. This approach makes it

**Figure 21 : Interface of Chat bot**

### 4.4.2 Screen Objects and Actions

**Home Screen:**
Bottom Bar: The home screen includes a bottom bar featuring various functionalities for quick access, including the calendar, AI assistant, and more.
Floating Action button: Create new Task Category.
Drawer Tabs: The drawer tab contains additional features and functionalities of the app.
**Todo Screen:**
Floating Action Button: Create a new Todo.
Search Bar: Search a Todo.
Tab bar: Tab bar has two tabs done and doing which list the tasks that are completed in the done tab and incomplete task in the doing tab.
**Calendar Screen:**
Floating Action Button: Schedule new event on the calendar.
Tab bar: Tab Bar has two tabs done and doing which list the tasks that are completed in the done tab and incomplete task in the doing tab.
**AI chat bot Screen:**
Text Box**:** User can type their message in the text box.
Send Icon Button: Send Icon button send the user request to the Gemini Api.
**Focus Mode Screen:**
Allow Button: Ask for user to grant permission.
Option Buttons: User can choose an option that are listed on the screen.
Slider: Slide the slider to turn it on and off.

**Note Taking:**
Floating Action Button: Create New Note.
Bottom tab bar : User can switch between text notes and image notes.
**Analytic:**
Bottom bar: User can Navigate between different option.
**Pomodoro Screen:**
Textbox: User can enter work, break duration and number of Session.
Start Button: Starts the timer.
Bottom bar: User can navigate to analytic and history screen of pomodoro.
**Habit tracker:**
Floating Action Button: Create New Habit.
Checkbox: Mark Habit as Completed.
Heat Map: Heat Map shows the Habit Progress.

# 5 Implementation

## 5.1 Algorithm

**Table 36 : Algorithm**

**Algorithm 1 Pomodoro Timer**

**Input:Work duration, Break Duration, Number of Session.**

**Output:** Pomodoro Timer.

```
1. Class PomodoroTimer:
        Private Variables:
        sRunning: boolean
        time: Duration
        break: Duration
        timeInt: int
        counter: int
        sessionCount: int
        timerCount: int
        currMax: int
        timer: Timer
        prefs: SharedPreferences

2. Constructor:
         Initialize private variables with default values.

3. Public Methods:
        startTimer(): Start the timer.
        stopTimer(): Stop the timer.
        resetTimer(): Reset the timer.
        storeTime(): Store the logged time in SharedPreferences.
        getPrefs(): Get the SharedPreferences instance.

4. .Private Methods:
        initState(): Initialize the state of the timer.
        getPrefs(): Retrieve SharedPreferences instance.
        storeTime(): Store logged time in SharedPreferences.
        startTimer(): Start the countdown timer.
        stopTimer(): Stop the countdown timer.

5. Function _initState():
        Initialize private variables with provided or default values.

6. Define a method to start the timer:
           Function _startTimer():

7. Start a Timer.periodic to update the time every second.
        Update _time and _timerCount accordingly.
        Handle session completion and stop the timer if necessary.
```

```
8. Function _stopTimer():
        Cancel the running Timer.

9. Define a method to reset the timer:
        Function resetTimer():
         Reset the timer to initial values.
        Stop the timer if running.

10. Define a method to store the logged time:
        Function storeTime():
         Retrieve SharedPreferences instance.
        Store logged time with session count and date.

11. Define a method to retrieve SharedPreferences instance:
        Function getPrefs():
        Return the SharedPreferences instance.

12. Define a method to handle session completion:
        Function _handleSessionCompletion():
        Display a message indicating session completion.
        Log the session details.
         top the timer.
13. Define a method to display messages:
        Function displayMessage(message):
        Display the message to the user using a snack bar or dialog.
```

**Algorithm 2 Habit Tracker**

**Input:** Habit

**Output: Habit Tracking**

```
1. Class HabitDatabase:
        todaysHabitList
        HeatMapDataSet

2. Function createDefaultData():
        SET todaysHabitList to default habits
        STORE current date in database

3. Function loadData():
        IF todaysHabitList not in database:
        LOAD habit list from database
        SET all habit completed flags to false
        ELSE:
        LOAD todays habit list from database

4. Function updateDatabase():
        STORE todays habit list in database
        STORE universal habit list in database
        CALCULATE habit completion percentages
        LOAD heat map

5. Function calculateHabitPercentages():
        COMPUTE count of completed habits
        COMPUTE percentage of completed habits
        STORE percentage in database

6. FunctionloadHeatMap():
        COMPUTE startDate from database
        COMPUTE daysInBetween
        FOR EACH day in daysInBetween:
        COMPUTE strengthAsPercent
        COMPUTE year, month, day
        STORE percentForEachDay in heatMapDataSet

7. Function checkBoxTapped(value, index):
        UPDATE todaysHabitList[index] with new completion status
        CALL updateDatabase()
```

```
8.  Function createNewHabit():
        GET new habit name from user input
        WHEN input received:
        ADD new habit to todaysHabitList
        CALL updateDatabase()

9.  Function saveNewHabit():
        ADD new habit to todaysHabitList
        CALL updateDatabase()

10.  Function deleteHabit(index):
        REMOVE habit from todaysHabitList
        CALL updateDatabase()
```

**Algorithm: Chat with AI powered Chatbot**

```
1. Define a class Message:
        isUser: boolean
        message: string
        date: DateTime

2. Define a widget Messages:
        Parameters:
       isUser: boolean
        message: string
        date: string

3. Function talkWithGemini():
        Get user input text from
        Add user's message to _message list
        Use GenerativeModel instance to generate a response based on user input
        Add generated response to _message list
```

**Algorithm 2 Task Management**

**Input: Task and Todos detail**

**Output: Task and Todos with progress**

```
1. class Tasks {
          Id id;
          String title;
          String description;
          int taskColor;
          bool archive;
          int? index;
          List<Todos>
}

2. class Todos {
          Id id;
          String name;
          String description;
          DateTime? todoCompletedTime;
          bool done;
          bool fix;
          Tasks task
}

3. Initialize Isar Database and Collections
       Isar openDB() {
          if (Isar.instanceNames.isEmpty) {
             // Create Isar instance with specified schemas
             return Isar.open([TasksSchema, TodosSchema, SettingsSchema]);
          }
```

```
            return Isar.getInstance();
          }

4. void addTask(String title, String desc, Color myColor) {
           Tasks newTask = Tasks(
               title: title,
               description: desc,
               taskColor: myColor.value,
               archive: false, // Assuming the task is not archived initially
            );

            // Add the new task to the tasks collection
            tasks.add(newTask); // Assuming 'tasks' is a list of Tasks objects

            // Save the new task to the Isar database
            isar.writeTxnSync(() => isar.tasks.putSync(newTask));

            // Optionally show a success message to the user
            print('Task added successfully!');

           }

5. void updateTask(Tasks task, String title, String desc, Color myColor) {
              // Assuming 'task' is an existing task object that needs to be updated
            // Modify the task properties based on the provided parameters
            task.title = title;
            task.description = desc;
            task.taskColor = myColor.value;

            // Update the task in the tasks collection (assuming 'tasks' is a list of Tasks objects)
            int taskIndex = tasks.indexOf(task);
            if (taskIndex != -1) {
               tasks[taskIndex] = task;
               tasks.refresh(); // Refresh the list to reflect the changes
            }

            // Save the updated task to the Isar database
            isar.writeTxnSync(() => isar.tasks.putSync(task));

            // Optionally show a success message to the user
            print('Task updated successfully!');

          }

6. void deleteTask(List<Tasks> taskList) {
            List<Tasks> taskListCopy = List.from(taskList);

            for (var task in taskListCopy) {
               // Delete associated todos
               List<Todos> todosToDelete = isar.todos.filter().task((q) => q.idEqualTo(task.id)).findAllSync();
               for (var todo in todosToDelete) {
                  // Optionally cancel any associated notifications (if needed)
                  if (todo.todoCompletedTime != null && todo.todoCompletedTime.isAfter(now)) {
                     await flutterLocalNotificationsPlugin.cancel(todo.id);
                  }
                  // Remove the todo from the todos list
                  todos.remove(todo);
                  // Delete the todo from the Isar database
                  isar.writeTxnSync(() => isar.todos.deleteSync(todo.id));
               }

               // Remove the task from the tasks list
               tasks.remove(task);
               // Delete the task from the Isar database
               isar.writeTxnSync(() => isar.tasks.deleteSync(task.id));

               // Optionally show a success message to the user
               print('Task deleted successfully!');
          }
```

```
7. void archiveTask(List<Tasks> taskList) {
            List<Tasks> taskListCopy = List.from(taskList);

           for (var task in taskListCopy) {
             // Archive the task
             isar.writeTxnSync(() {
               task.archive = true;
               isar.tasks.putSync(task);
             });

         }
```

## 5.2 External APIs/SDKs

Gemini API is utilized within the productivity app to empower users with AI-powered text-based interactions and assistance.

**Table 37 : Details of APIs used in the project.**

| **Name of API and version** | **Description of API** | **Purpose of usage** | **List down the API endpoint/function/class in which it is used** |
|---|---|---|---|
| Google Gemini | AI powered text-based assistance | AI powered Virtual Assistance. | within the AI Chatbot to handle text-based interactions. |

## 5.3 User Interface

### 5.3.1 Welcome Screen

Upon launching the application, users are greeted with a welcoming interface, serving as their initial point of interaction. The design is intuitive and visually appealing, ensuring a smooth and engaging start.

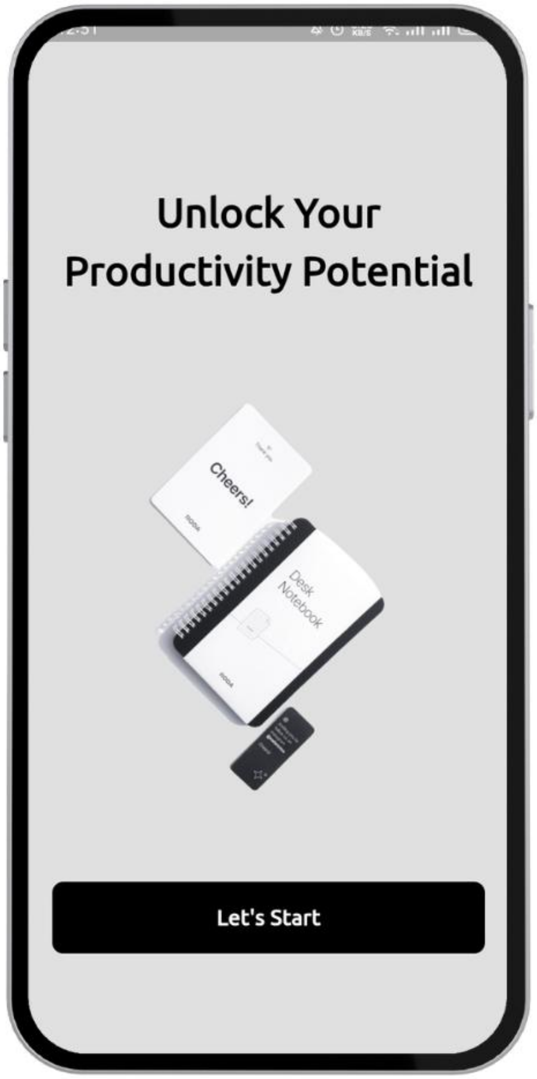
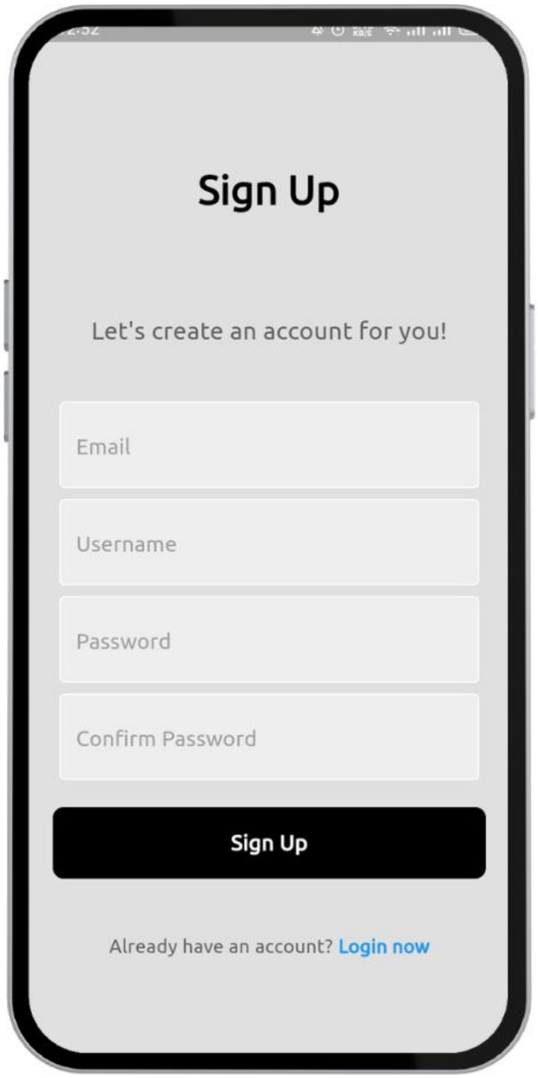

Unlock Your Productivity Potential

Figure 22 : Welcome Screen

### 5.3.2 Sign Up

The Sign-Up screen simplifies new user registration, guiding them through the process by collecting essential information such as name, email, and password. With its intuitive design, it ensures a seamless onboarding experience, setting the stage for users to explore and utilize the app's features effectively.



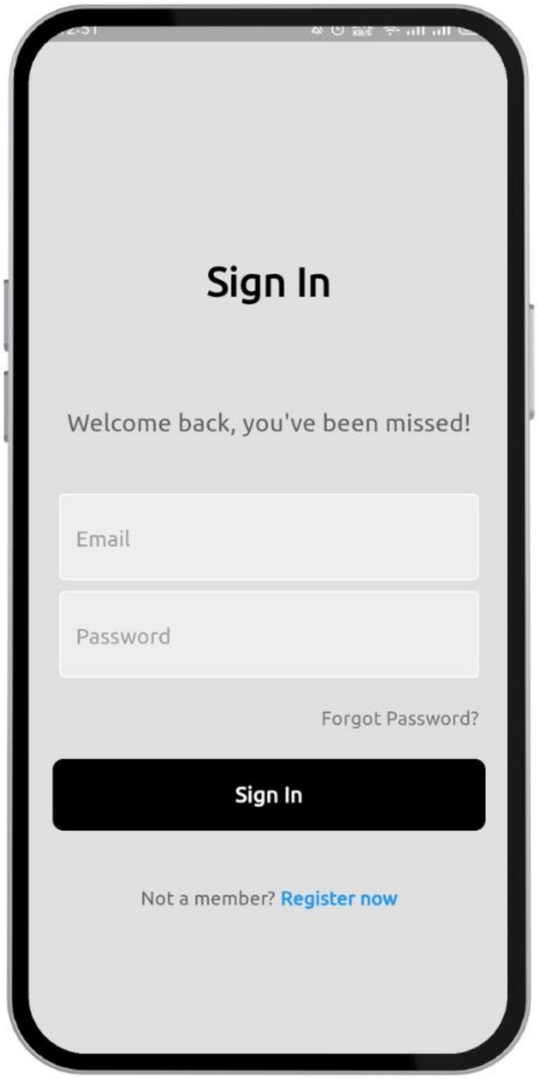
Let's create an account for you!

Already have an account? Login now

**Figure 23 : Sign Up Screen**

### 5.3.3 Sign In

The Sign In screen provides users with a safe and straightforward method to access their accounts, utilizing their email address and password for secure authentication. It offers a seamless login experience, ensuring users can quickly and confidently access the app's features and functionalities.

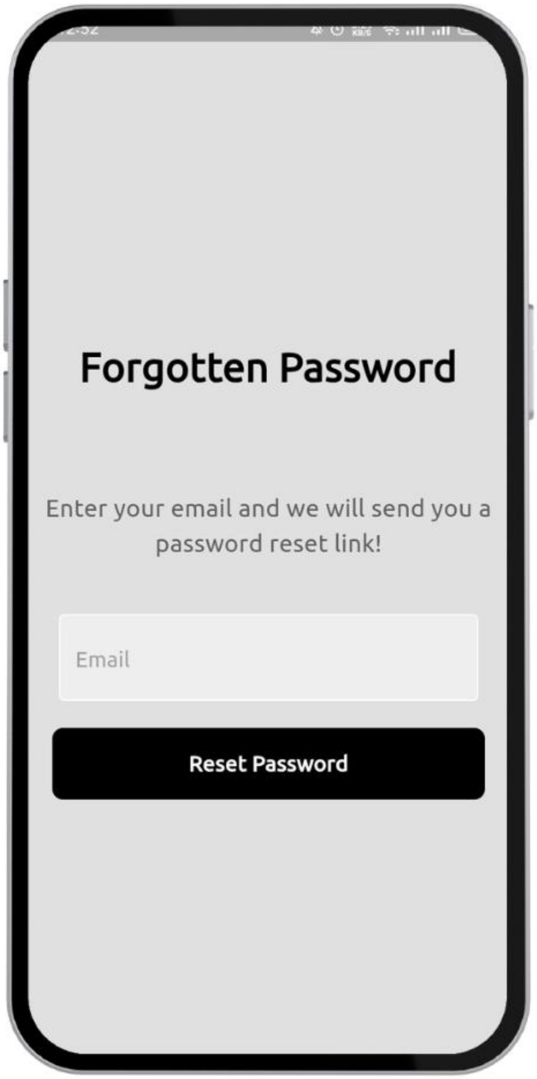


Welcome back, you've been missed!

**Figure 24 : Sign In Screen**

### 5.3.4 Forgot Password

The Forgot Password screen allows users to reset their password by providing their email address. Once submitted, users receive an email with instructions on how to reset their password, enhancing account security and accessibility.



Forgotten Password

Enter your email and we will send you a password reset link!

**Figure 25 : Forgot Password Screen**

### 5.3.5 Home Screen

The Home Screen showcases all user tasks along with their progress for quick monitoring. Users can seamlessly access additional features from the bottom bar, enhancing navigation and productivity.

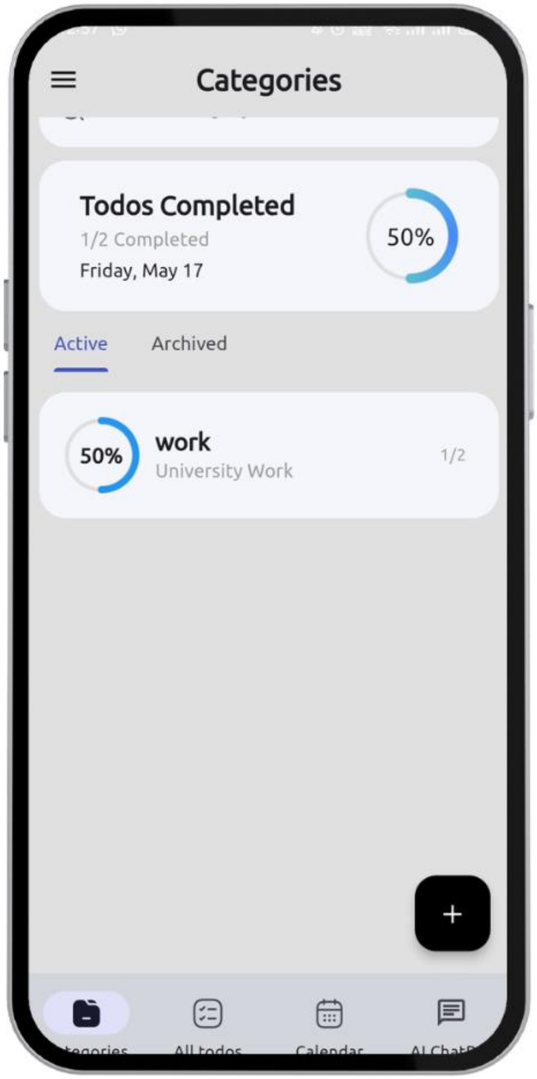


**Figure 26 : Home Screen**

### 5.3.6 Drawer Tab

The Drawer Tab List serves as a central hub for accessing all available app features, offering convenient navigation to various sections and functionalities. Users can effortlessly explore Focus Mode , habit tracking, note-taking, pomodoro Timer capabilities, enhancing their productivity .

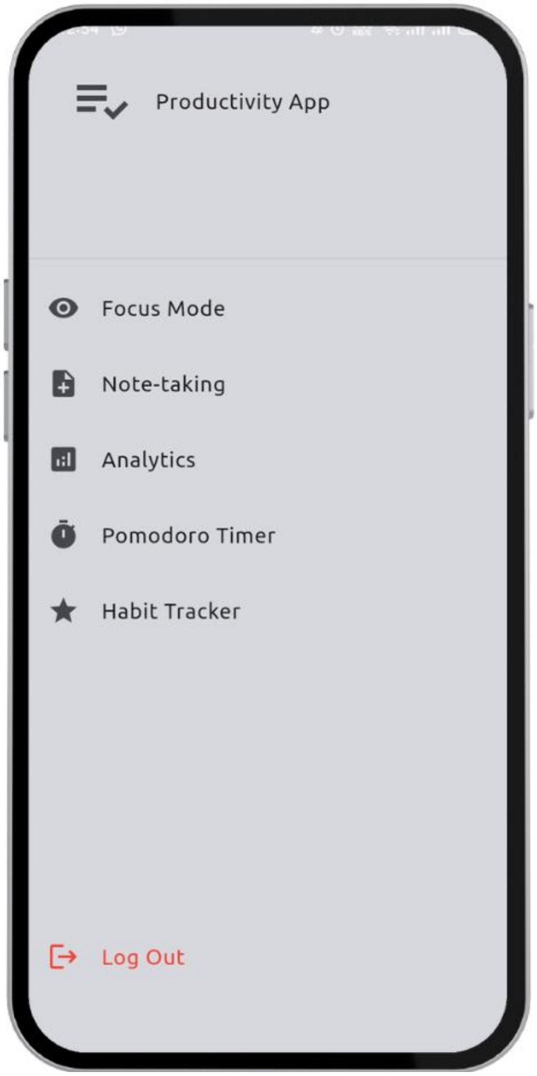
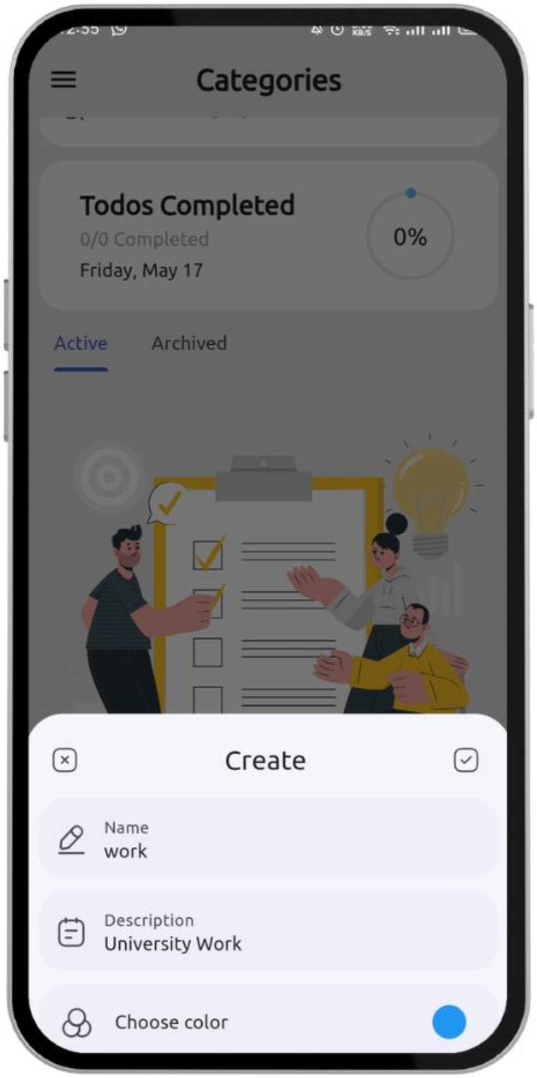


**Figure 27 : Drawer Tab Screen**

### 5.3.7 Create New task.

The Create New Task screen empowers users to input essential task details, including title, description, and color, facilitating organized and personalized task management. Additionally, users can track the progress of their tasks, providing valuable insights into their productivity and workflow management.



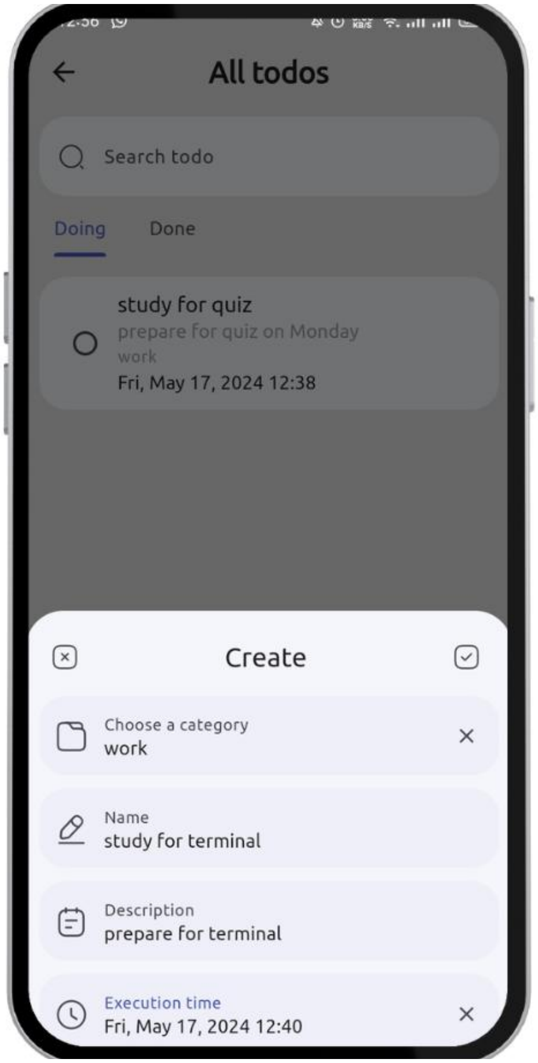
Name
work

Description
University Work

**Figure 28 : Create New Task Screen**

### 5.3.8 Create New Todo

The Create New Todo screen empowers users to seamlessly generate to-dos by specifying crucial details such as name, description, and execution time. Furthermore, users can categorize their to-dos under specific task categories, enhancing organization and management efficiency within the app.



study for quiz
prepare for quiz on Monday
work
Fri, May 17, 2024 12:38

Choose a category
work

Name
study for terminal

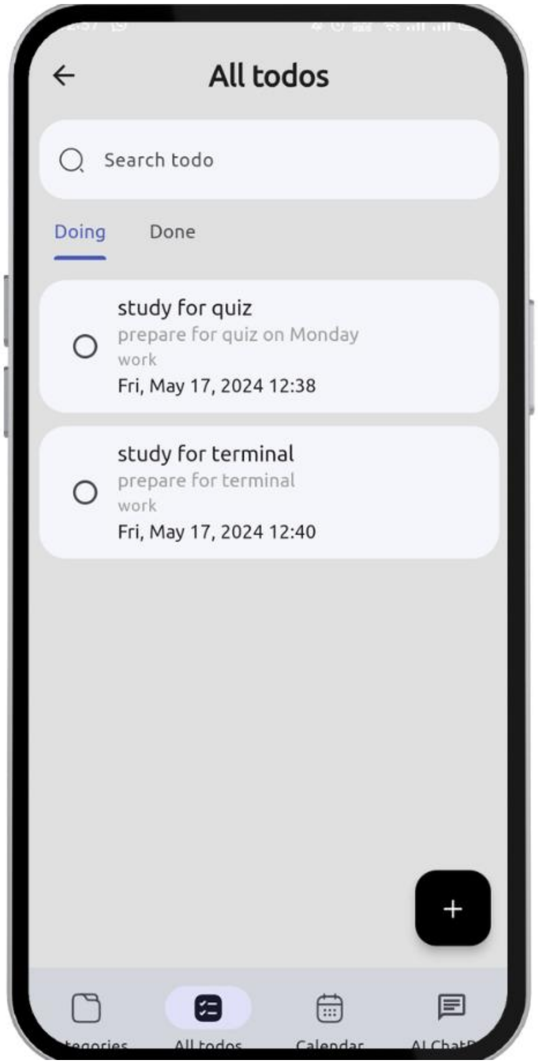
Description
prepare for terminal

Execution time
Fri, May 17, 2024 12:40

**Figure 29 : Create New Todo Screen**

### 5.3.9 Todo Screen

The Todo Screen displays all current to-dos, providing users with an overview of their pending tasks. Users can view the names, descriptions, and execution times of each to do, allowing for efficient task management .



All todos

Search todo

Doing Done

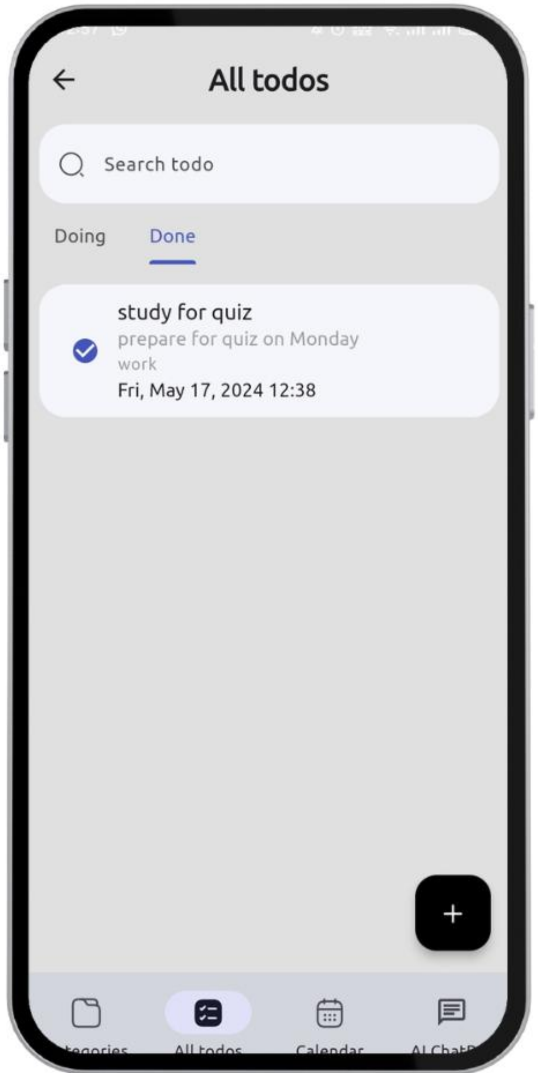
○ study for quiz
prepare for quiz on Monday
work
Fri, May 17, 2024 12:38

○ study for terminal
prepare for terminal
work
Fri, May 17, 2024 12:40

**Figure 30 : Todos Screen**

### 5.3.10 Done Todo Screen

The Done Todo Screen displays all the completed to-dos, providing users with a record of their accomplished tasks. Users can review the names, descriptions, and execution times of each completed to do, allowing for a sense of accomplishment and progress tracking.

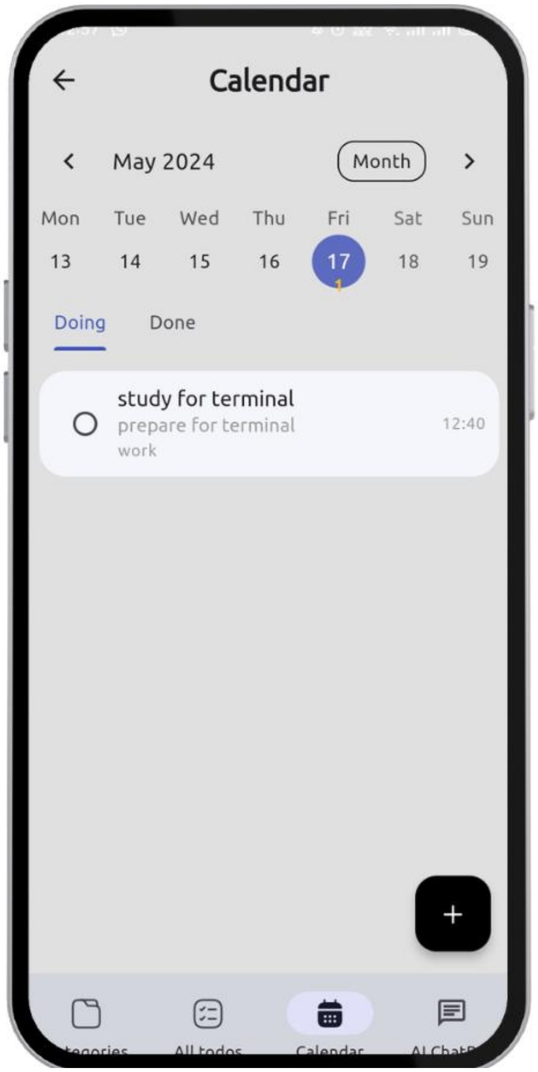


study for quiz
prepare for quiz on Monday
work
Fri, May 17, 2024 12:38

**Figure 31 : Done Todos Screen**

### 5.3.11 Calendar Screen

The Calendar Screen showcases a calendar interface where users can view their schedules and upcoming events. Users have the capability to schedule new events directly from the calendar, specifying details such as event name, date, time, and any additional notes.

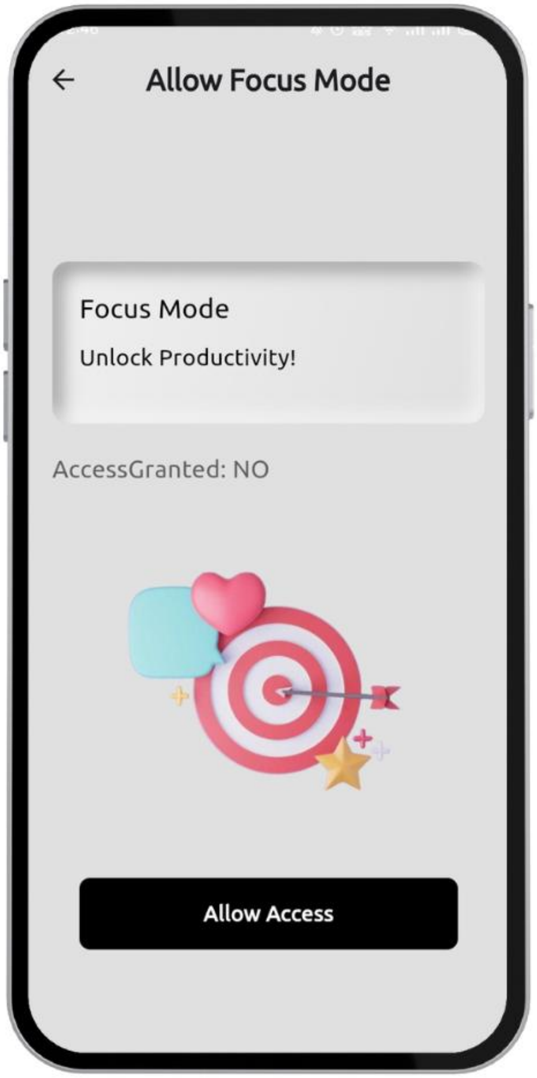


study for terminal
prepare for terminal
work
12:40

Figure 32 : Calendar Screen

### 5.3.12 Allow Focus Mode Permission

The Allow Focus Mode Screen prompts users to grant permission for the productivity app to activate its focus mode feature, facilitating uninterrupted concentration and enhancing productivity.



Allow Focus Mode

Focus Mode

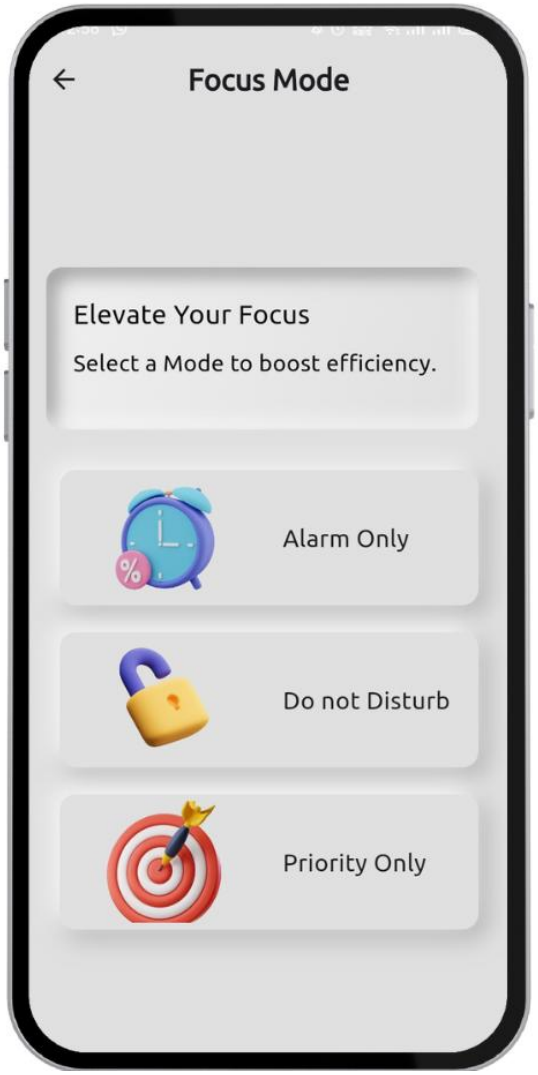
Unlock Productivity!

AccessGranted: NO

**Figure 33 : Allow focus Mode Permission Screen**

### 5.3.13 Select Focus Mode Option

Users can select from the following three options listed on the screen  based on their individual preferences and requirements.This personalized approach ensures that users can optimize their productivity according to their requirement.



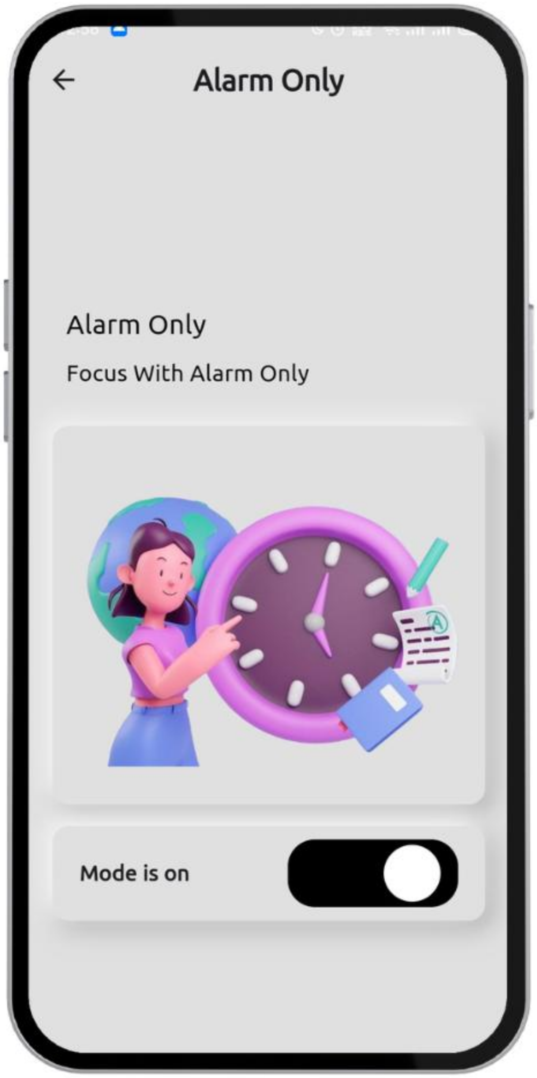
Elevate Your Focus

Select a Mode to boost efficiency.

**Figure 34 : Select Focus Mode Screen**

### 5.3.14 Alarm Only Screen

Users can select from a diverse range of options, to suit their unique requirements and preferences, ensuring a personalized experience tailored to their individual needs.



Alarm Only

Alarm Only

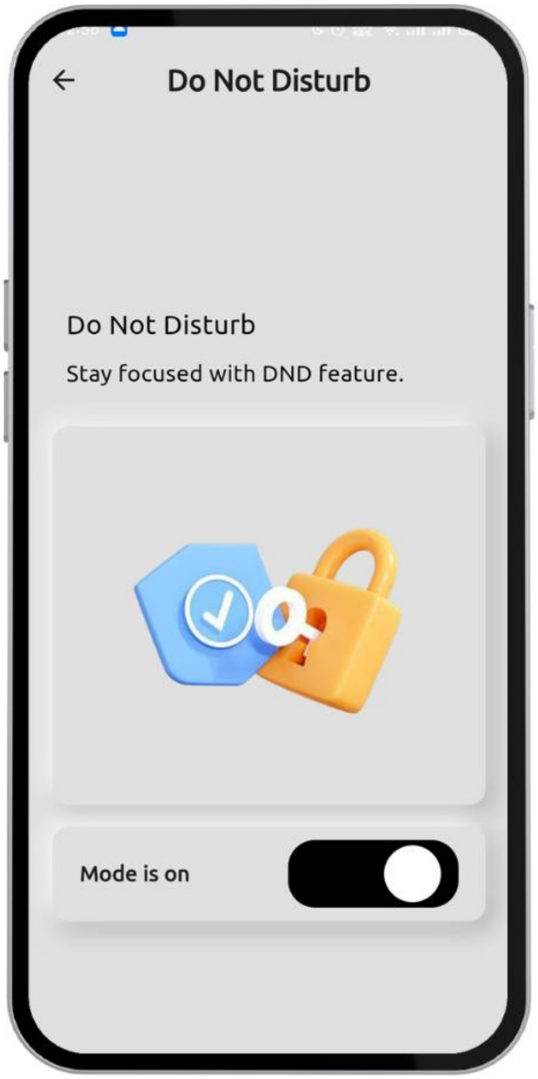
Focus With Alarm Only

Mode is on

**Figure 35 : Alarm Only Mode Screen**

### 5.3.15 Do not Disturb Mode Screen

Users have the flexibility to toggle the Do Not Disturb Focus mode on and off, empowering them to control their level of concentration and minimize interruptions as needed.



Do Not Disturb

Do Not Disturb

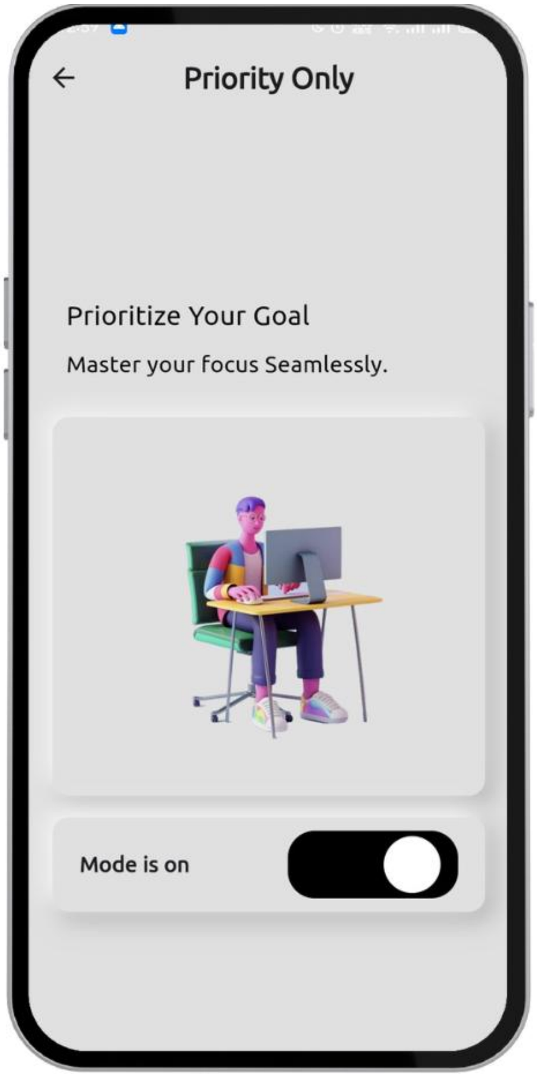
Stay focused with DND feature.

Mode is on

Figure 36 : Do not Disturb Mode Screen

### 5.3.16 Priority App Screen

Users can easily enable or disable the Allow Priority App Notifications Only Focus mode, granting them the ability to customize their focus experience based on their preferences and priorities.



Priority Only

Prioritize Your Goal

Master your focus Seamlessly.

Mode is on

**Figure 37 : Priority Only Screen**

### 5.3.17 Notes Screen

The Notes Screen presents users with an organized display of all notes, showcasing titles and creation dates for easy reference and efficient note management. Users can also create new notes directly from the screen, allowing for seamless integration of ideas and information into their workflow.

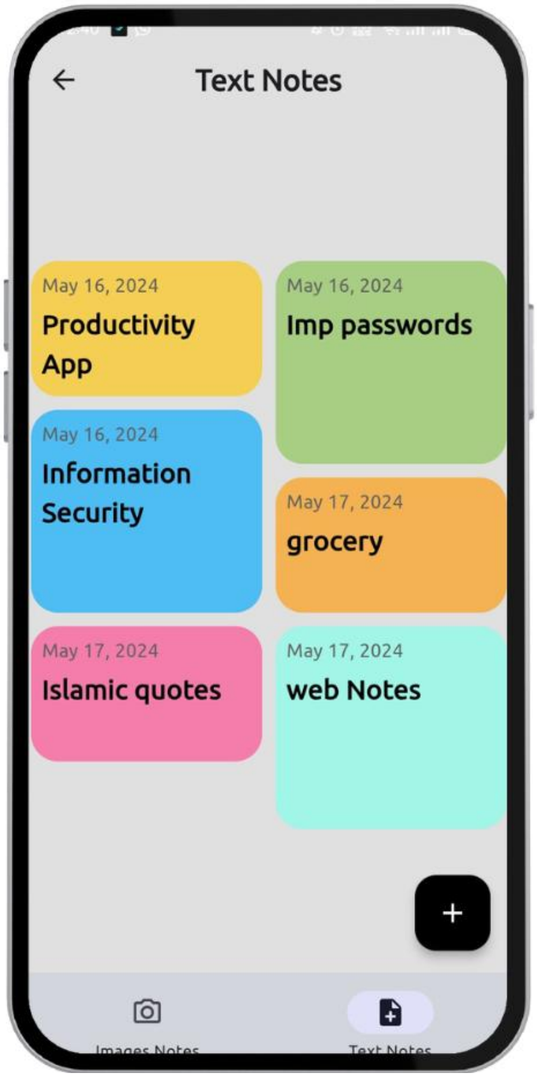
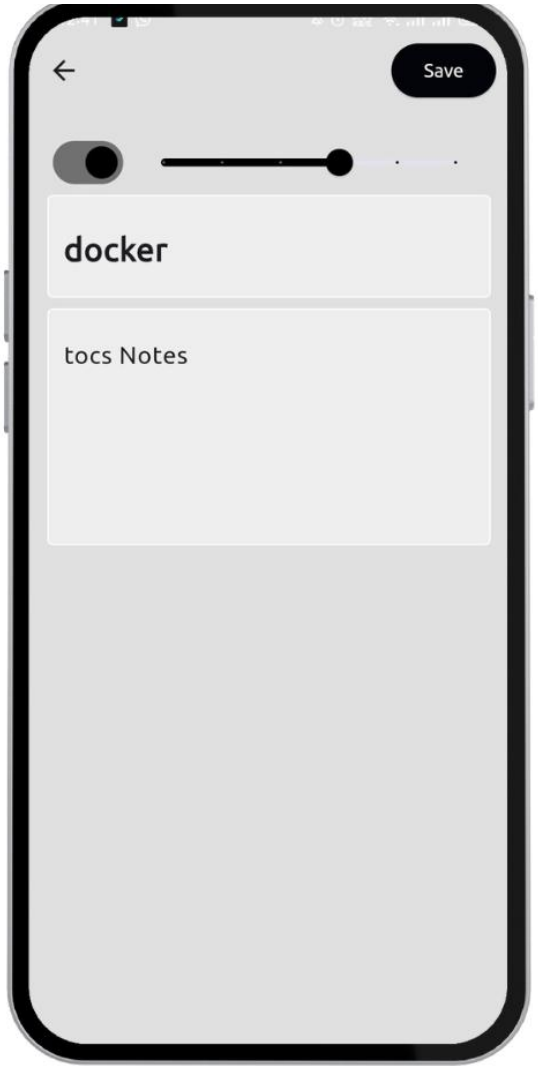


Figure 38 : Notes Taking Screen

### 5.3.18 Create New Notes

The Create New Notes screen allows users to compose and save new notes with titles, descriptions, and other relevant details. Users can effortlessly input their thoughts ensuring clarity and organization in their note-taking process.



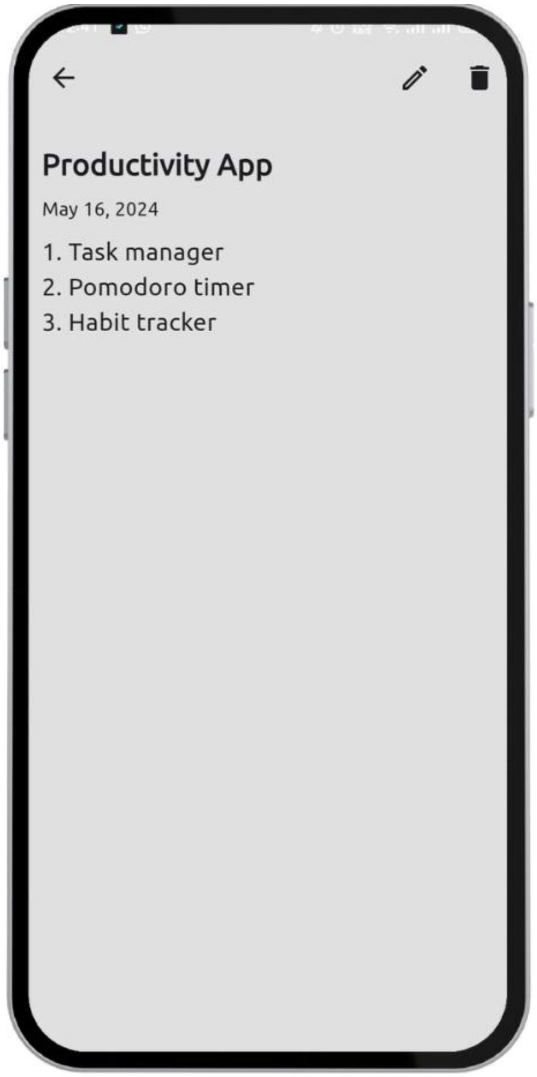
docker

tocs Notes

Figure 39 : Create New Notes Screen

### 5.3.19 View, Edit, or Delete Text Notes

Users have the option to edit or delete specific notes as desired, providing them with control over their note-taking and organization within the app. This flexibility ensures that users can refine their notes over time or remove outdated information, maintaining a streamlined and efficient note-taking experience.



Productivity App

May 16, 2024

1. Task manager
2. Pomodoro timer
3. Habit tracker

**Figure 40 : View ,Edit,Delete Note Screen**

### 5.3.20 Capture image Notes

The Image Note Capture Screen provides users with a seamless interface to capture visual notes, empowering them to preserve important information through images. Users can effortlessly capture images using their device's camera and store them for future reference. Additionally, the screen allows users to browse and view previously captured image notes, facilitating easy retrieval and organization of visual content.

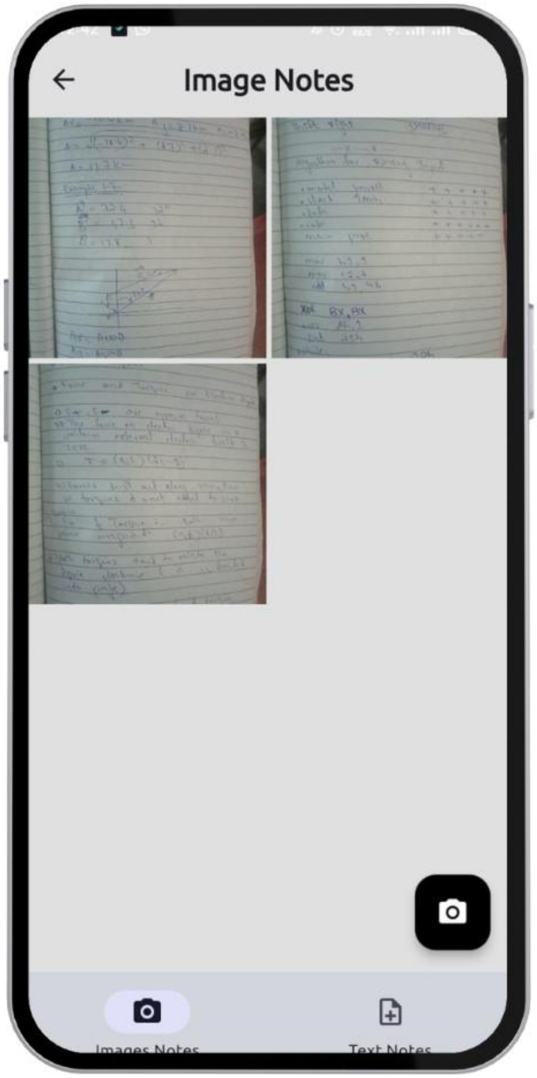
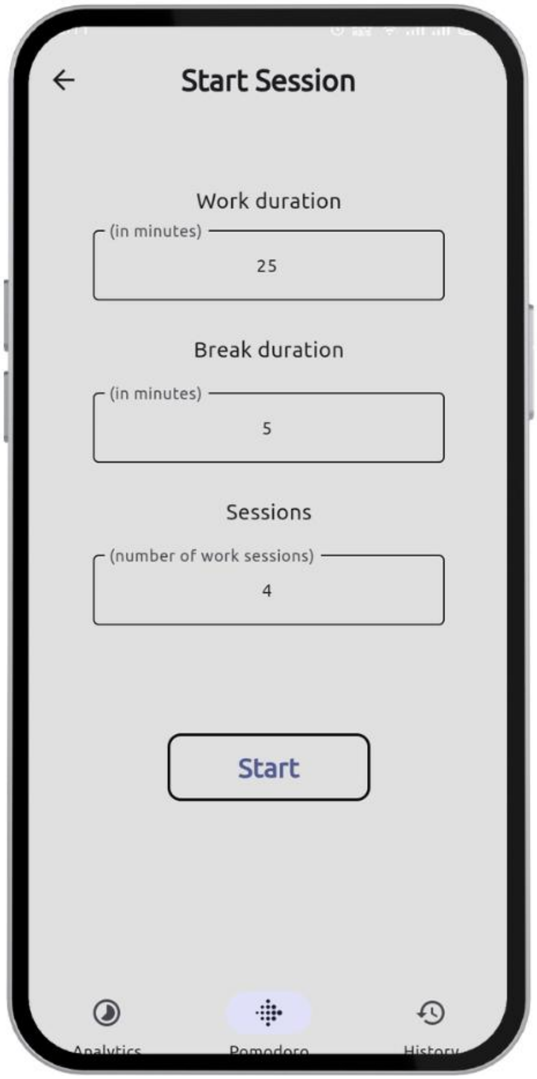


**Figure 41 : Capture Image Notes**

### 5.3.21 Set Pomodoro

The Set Pomodoro screen enables users to input the duration of work and break sessions, as well as the number of sessions they wish to complete to start a Pomodoro. Users can customize their workflow to match their productivity needs, ensuring a balanced and effective time management strategy.



Start Session

Work duration (in minutes): 25

Break duration (in minutes): 5

Sessions (number of work sessions): 4

**Figure 42 : Set Pomodoro Screen**

### 5.3.22 Timer Screen

The timer feature on the Pomodoro screen displays both work and break session timer, helping users stay on track with their productivity sessions. It visually distinguishes between work and rest periods, ensuring users maintain a balanced workflow.

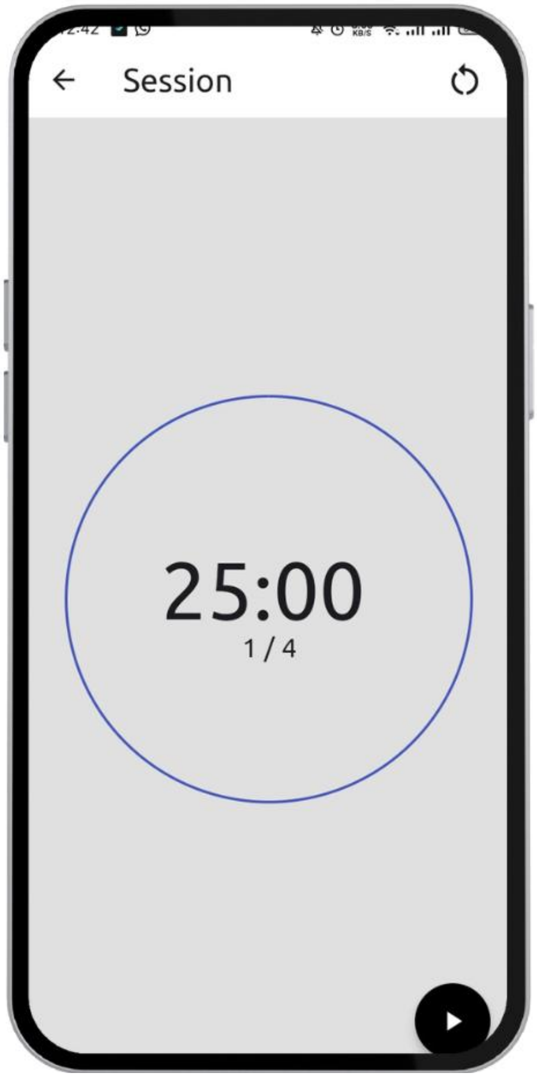
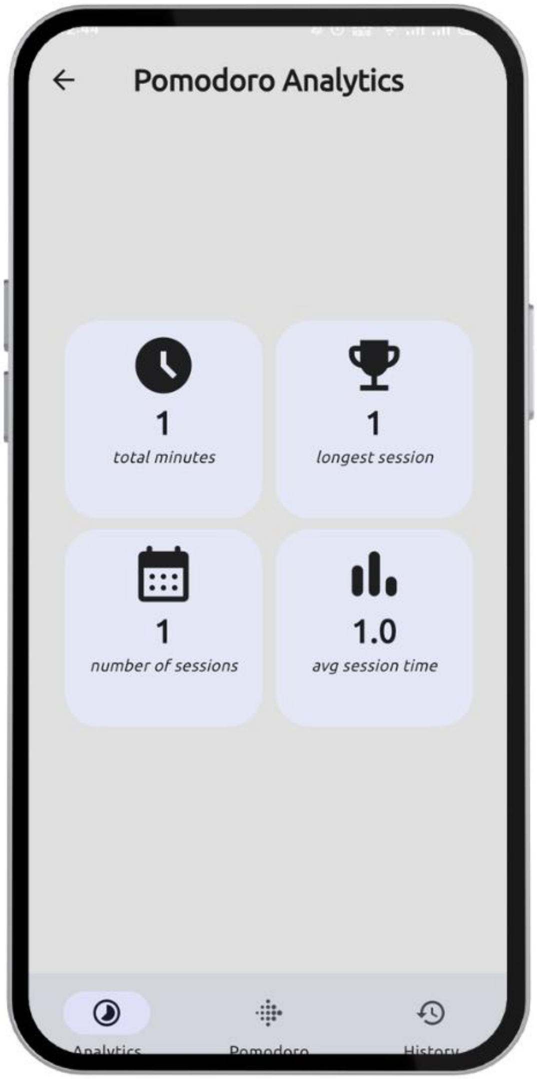


**Figure : Pomodoro Timer Screen**

### 5.3.23 Pomodoro Analytics

The "Pomodoro Analytics" screen presents users with insights such as total minutes spent, longest session duration, number of sessions completed, and average session time. Users can track their productivity trends.This comprehensive overview helps optimize time management and enhance overall efficiency.



Pomodoro Analytics

1 *total minutes*

1 *longest session*

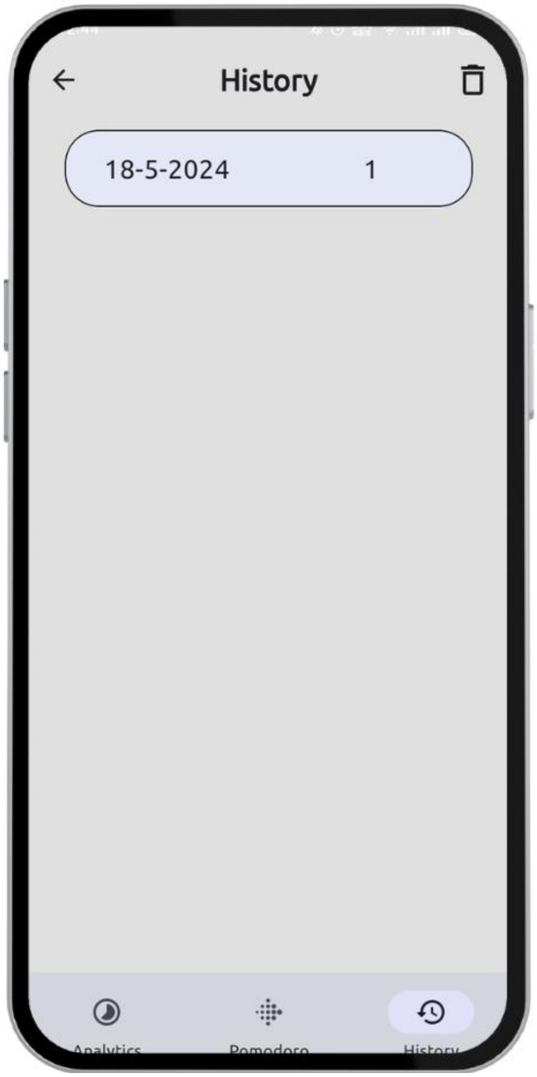
1 *number of sessions*

1.0 *avg session time*

Figure 43: Pomodoro Analytics Screen

### 5.3.24 Pomodoro History Screen

Displays completed Pomodoro sessions, including the dates and the number of sessions completed on each day, helping users track their productivity over time. By providing a clear historical record, users can identify their most productive days, and make adjustments to improve their efficiency. This feature supports better planning and goal-setting for future tasks.



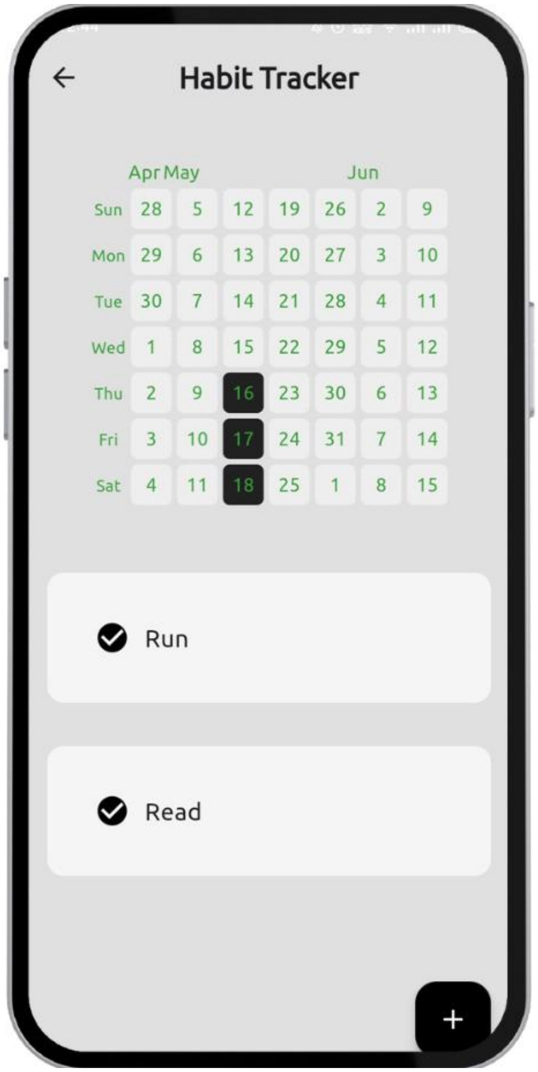
History

| | |
|---|---|
| 18-5-2024 | 1 |

Figure 44 : Pomodoro History Screen

### 5.3.25 Habit Tracker

The Habit Tracker screen enables users to monitor their daily, weekly, or monthly progress in establishing and tracking goals or habits. It provides visual representations of streaks and consistency, helping users stay motivated and accountable.



Habit Tracker

|  | Apr | May |  |  |  | Jun |  |
|---|---|---|---|---|---|---|---|
| Sun | 28 | 5 | 12 | 19 | 26 | 2 | 9 |
| Mon | 29 | 6 | 13 | 20 | 27 | 3 | 10 |
| Tue | 30 | 7 | 14 | 21 | 28 | 4 | 11 |
| Wed | 1 | 8 | 15 | 22 | 29 | 5 | 12 |
| Thu | 2 | 9 | 16 | 23 | 30 | 6 | 13 |
| Fri | 3 | 10 | 17 | 24 | 31 | 7 | 14 |
| Sat | 4 | 11 | 18 | 25 | 1 | 8 | 15 |

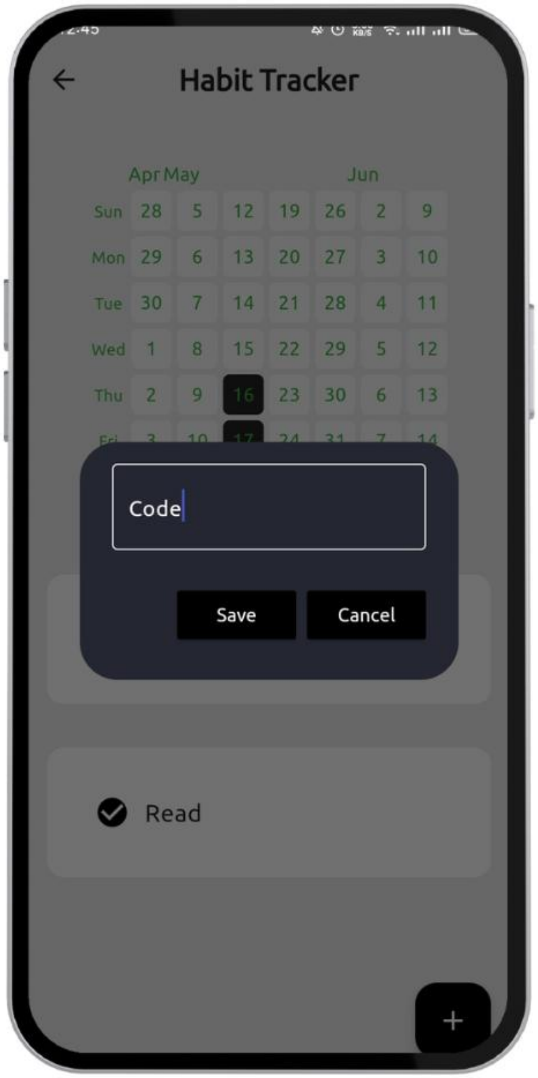
✓ Run

✓ Read

Figure 45 : Habit tracker Screen

### 5.3.26 Create New Habit

Users can create new habits, empowering them to cultivate and track positive behaviors within the app. The feature allows for personalized goal setting, making it easy to monitor progress and stay motivated.



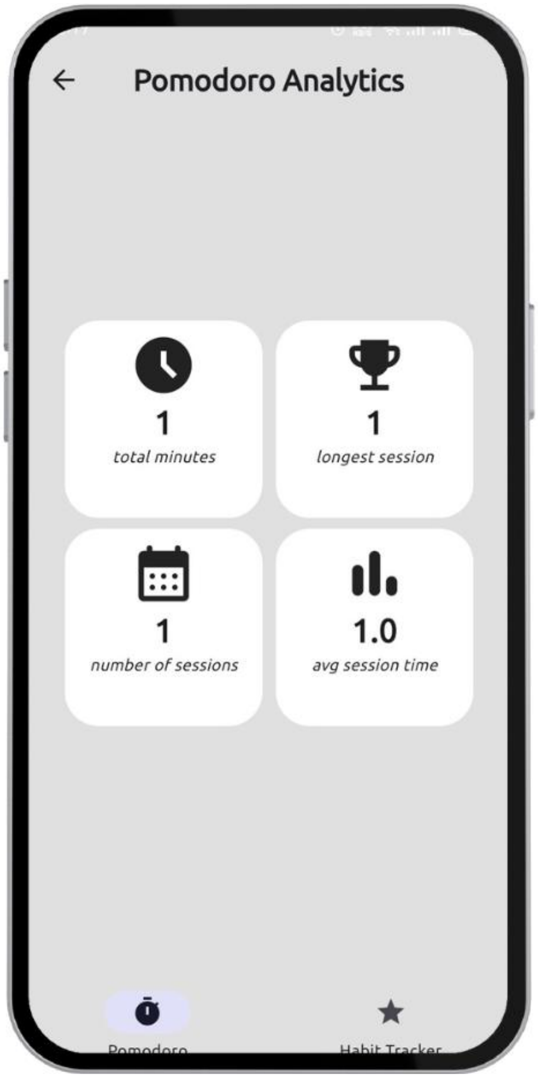
|  | Apr | May |  |  |  | Jun |  |
|---|---|---|---|---|---|---|---|
| Sun | 28 | 5 | 12 | 19 | 26 | 2 | 9 |
| Mon | 29 | 6 | 13 | 20 | 27 | 3 | 10 |
| Tue | 30 | 7 | 14 | 21 | 28 | 4 | 11 |
| Wed | 1 | 8 | 15 | 22 | 29 | 5 | 12 |
| Thu | 2 | 9 | 16 | 23 | 30 | 6 | 13 |
| Fri | 3 | 10 | 17 | 24 | 31 | 7 | 14 |

Code

**Figure 46 : Create New habit Screen**

### 5.3.27 Analytics

This screen features two bottom tabs, providing users with comprehensive insights into their productivity. The "Pomodoro Analytics" tab displays metrics such as total minutes spent, longest session duration, number of sessions completed, and average session time. The "Habit Analytics" tab offers a detailed view of habit tracking progress, allowing users to monitor their consistency and improvement over time.



Pomodoro Analytics

1
*total minutes*

1
*longest session*

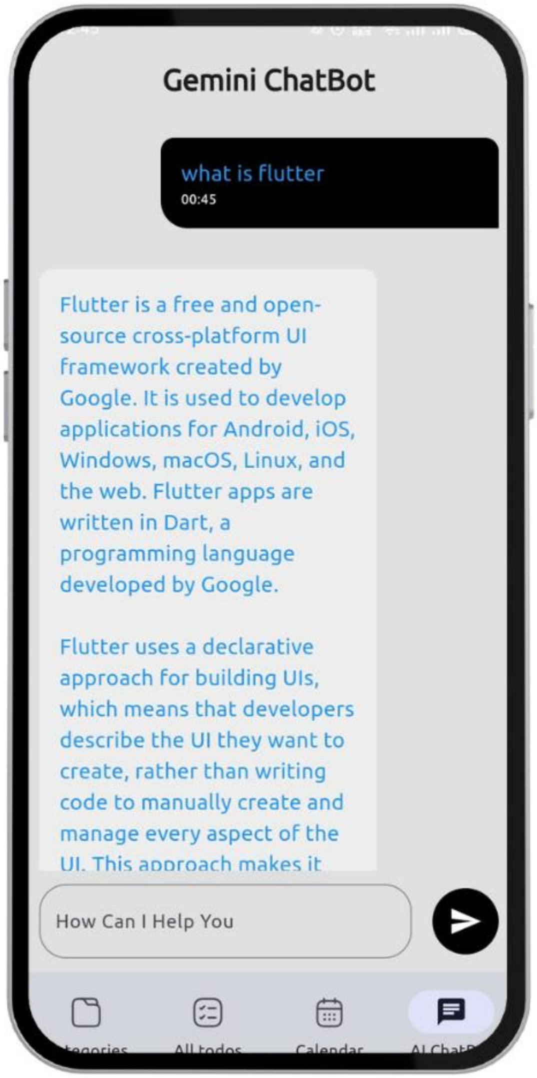
1
*number of sessions*

1.0
*avg session time*

**Figure 47 : Analytics Screen**

### 5.3.28 AI powered Chat bot Assistant

Users can engage in chat with Gemini, an interactive virtual assistant, receiving its responses directly on the screen for seamless communication and assistance. Gemini provides instant answers to queries. This real-time interaction enhances user engagement and support, making it easier to achieve their goals.



Gemini ChatBot

what is flutter
00:45

Flutter is a free and open-source cross-platform UI framework created by Google. It is used to develop applications for Android, iOS, Windows, macOS, Linux, and the web. Flutter apps are written in Dart, a programming language developed by Google.

Flutter uses a declarative approach for building UIs, which means that developers describe the UI they want to create, rather than writing code to manually create and manage every aspect of the UI. This approach makes it

Figure 48 :AI Powered Chat bot Screen

## 5.4 Deployment

1. Hosting Service:
The productivity app is currently not hosted on any platform but will soon be hosted on Google play store.
2. Available Software and Versions:
Flutter SDK (Version 3.19.6): Used for building the cross-platform mobile application.
Android Studio (Version 2024): Integrated development environment (IDE) for Android app development.
Firebase: Firebase Authentication for user authentication.
3. Operating Systems:
Development Machines: Windows, macOS, or Linux-based systems depending on developers' preferences.
Target Platforms: Android mobile devices for testing and deployment.
4.Other Tools:
IDE Plugins: Flutter and Dart plugins for Android Studio or Visual Studio Code.
Emulators: Android Emulator, I and physical devices for testing across different platforms.

# 6 Testing and Evaluation

## 6.1 Unit Testing

**Unit Testing 1:** Login with valid and invalid credentials.

**Testing Objective:** To ensure the login form is working correctly with valid and invalid credentials/inputs.

**Table 38 : Unit Test Login With Valid and invalid credentials.**

| **No.** | **Test case/Test script.** | **Attribute and value.** | **Expected result.** | **Result** |
|---|---|---|---|---|
| 1 | Ensure that user can login with valid email | Email: maryamamjad532@gmail.com<br>Password: 12345678 | Validates user credential and move to the home screen | Pass |
| 2 | If the email is wrong, it should generate an error message | Email: maryamamjad..532@gmail.com<br>Password: 12345678 | Display an error message | Pass |
| 3 | If the user credentials don't exist | Email: xyz@gmail.com<br>Password: xyz | Display a message that no such user exists or has expired | Pass |
| 4 | Register with already existing email | Email: maryamamjad532@gmail.com<br>Password: 12345678 | Display a message that email address is already in use | Pass |

**Unit Testing 2:** Add Todos to Task
**Testing Objective:** To ensure the task and to-dos are working properly.

**Table 39 : Unit Test Table Add Todos to Task**

| No. | Test case/Test script. | Attribute and value. | Expected result. | Result |
|---|---|---|---|---|
| 1 | Create unique task category title | Task category title: "Work | Task category title is created Successfully | Pass |
| 2 | Add to do to category | Todo Title: Work on UI | Added Successfully | Pass |
| 3 | If to do title is empty that it should display a warning | Todo title: “ ” | Display a message “Please enter a title” | Pass |
| 4 | App shouldn’t allow the user to pick an execution time and date of todo past the current date | Date: 12-jan-2024<br>Time: 12:00 | Automatically choose the current time | Pass |
| 5 | Mark todo as done | Todo status done | Updated the Task Progress | Pass |
| 6 | User can’t edit todos that are mark as done | User tries to edit Work on UI todo | No update option | Pass |
| 7 | User can edit and delete incomplete todos | Edit Work on UI to Work on code.<br>Delete: Go to gym | Work on UI edited.<br>Go to Gym Deleted | Pass |

**Unit Testing 3:** Test Focus Mode
**Testing Objective:** To ensure if Focus Mode Actually blocks Notifications.

**Table 40: Unit Test Table Test Focus Mode**

| No. | Test case/Test script. | Attribute and value. | Expected result. | Result |
|---|---|---|---|---|
| 1 | User grant permission to app | Permission granted to access | Focus Mode no has permission to block notification | Pass |
| 2 | Do not Disturb Mode block all notifications | Do not disturb mode is turn on | No notifications are allowed | Pass |
| 3 | Alarm Only mode | Only alarm notification is | No other | Pass |

| | | | | |
|---|---|---|---|---|
| | allows only alarm notification | granted, and the rest are blocked | notifications except alarm | |
| 4 | Priority Only mode | Priority mode is activated | Only the apps with high priority can send notification | Pass |

**Unit Testing 4:** Test Pomodoro Timer
**Testing Objective:** To ensure Pomodoro Timer Works Efficiently

**Table 41 : Unit Test Table Test Pomodoro Timer**

| No. | Test case/Test script. | Attribute and value. | Expected result. | Result |
|---|---|---|---|---|
| 1 | Minimum time for work and break session is 1 minute | Work duration=0<br>Break Duration =0<br>No.of session=0 | Alert Message is generated "Please enter Valid numbers" | Pass |
| 2 | Each Work session is followed by a break session | Work duration=25<br>Break Duration =5<br>No.of session=1 | After Work timer ends break timer start | Pass |
| 3 | If the number of sessions are completed. A Message is shown | Work duration=25<br>Break Duration =5<br>No.of session=1 | Message is shown "Number of Session Completed" | Pass |
| 4 | Priority Only mode | Priority mode is activated | Only the apps with high priority can send notification | Pass |

**Unit Testing 5:** Test Habit Tracker
**Testing Objective:** To ensure habits are created and Tracked Properly

**Table 42 : Unit Test Table Test Habit Tracker**

| No. | Test case/Test script. | Attribute and value. | Expected result. | Result |
|---|---|---|---|---|
| 1 | User can create a new habit. | Habit: "running " | Habit created and is showing in the list | Pass |
| 2. | User can mark habit as completed | Habit running mark as done for today | Heat map is updated to show progress | Pass |
| | | | | |

**Unit Testing 6:** Test Note Taking
**Testing Objective:** To ensure Notes are created Properly and stored Properly.

**Table 43 : Unit Test Table Test Note Taking**

| No. | Test case/Test script. | Attribute and value. | Expected result. | Result |
|---|---|---|---|---|
| 1 | User can create note with proper title and description | Title: Grocery List<br>Description: Buy Milk | Note Created Successfully | Pass |
| 2. | User cannot create note without proper title and description | Title:<br>Description: | Warning message is shown | Pass |
| 3. | User can capture image as notes | User captured an image | Image is visible | Pass |

**Unit Testing 7:** Test Gemini Response
**Testing Objective:** To ensure Gemini is responding to user request.

**Table 44 : Unit Test Table Test Gemini Response**

| No. | Test case/Test script. | Attribute and value. | Expected result. | Result |
|---|---|---|---|---|
| 1 | Check if Gemini sends a response and that is human readable | User: “Hello Gemini” | “Hello there, I am Gemini……” | Pass |

## 6.2 Functional Testing

**Functional Testing 1:** User Authentication

**Table 45 : Functional Test Table User Authentication**

| No. | Test case/Test script. | Attribute and value. | Expected result. | Actual result | Result |
|---|---|---|---|---|---|
| 1. | Sign up | Email: maryamajad4321@gmail.com<br>Password:12345678 | Signup successfully and redirected to home screen | Signup successfully and redirected home screen | Pass |
| 2. | Login User | Email: maryamajad532@gmail.com<br>Password:12345678 | Login successfully and redirected home screen | Logged in and redirected to home screen | Pass |
| 3. | Recover | Choose Forgot password | Email sends to | Email received | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | Password | option.<br>Email:<br>maryamajad532@gmail.com | user your email address | and password changed | |

## Functional Testing 2: Task Management

**Table 46 : Functional Test Table Task Management**

| No. | Test case/Test script. | Attribute and value. | Expected result. | Actual result | Result |
|---|---|---|---|---|---|
| 1. | Creating a new Task and Adding Todos | Task: Office Work<br>Todo: Finish Email<br>Make Presentation | Task and to-dos are created successfully and displayed with progress | Task and to-dos are created successfully and displayed with progress | Pass |
| 2. | Users get Notification for the to-do and can mark them as completed. | Task: Office Work<br>Todo: Finish Email<br>Make Presentation | User got reminder and mark the task as completed and the progress is updated | User got reminder and mark the task as completed and the progress is updated | Pass |
| 3. | User can see the progress of the task | Task: Office Work<br>Todo: Finish Email(done)<br>Make Presentation (not done) | Progress should be 50% | Progress is 50% | Pass |

## Functional Testing 3: Event Scheduling

**Table 47 : Functional Test Table Event Scheduling**

| No. | Test case/Test script. | Attribute and value. | Expected result. | Actual result | Result |
|---|---|---|---|---|---|
| 1 | User can Schedule and event in the calendar on future any date and timer | Category: Work<br>Name: Make PPT<br>Desc: Work on PPT<br>Time 2/May/2024 | User event is scheduled and can be viewed, Edited, and deleted | User event is scheduled and can be viewed, Edited and deleted | Pass |
| 2. | User will get a notification on the execution time and date | Category: Work<br>Name: Make PPT<br>Desc: Work on PPT<br>Time 2/May/2024 | User got notification | User got notification | Pass |
| 3. | Scheduled event will be displayed in the to-do list if it is scheduled for today | Category: Work<br>Name: Make PPT<br>Desc: Work on PPT<br>Time 2/May/2024 | Todo is displayed in the Today List | Todo is displayed in the Today List | Pass |

**Functional Testing 4:** Note Taking

**Table 48 : Functional Test Table Note Taking**

| No. | Test case/Test script. | Attribute and value. | Expected result. | Actual result | Result |
|---|---|---|---|---|---|
| 1 | Users create a note with proper title and description and note is saved in the storage | Note Title: Grocery<br>Desc: Buy Fruits | Note is saved in the storage and can be seen in the notes list | Note is saved in the storage and can be seen in the notes lis | Pass |
| 2 | User can Edit Note | Note Title: Grocery<br>Desc: Buy vegetables. | Note is Edited | Note is Edited | Pass |
| 3 | User can delete Notes from storage | Delete Note Title: Grocery<br>Desc: Buy vegetables. | Noted is Deleted from storage | Noted is Deleted from storage | Pass |

**Functional Testing 5:** Set Pomodoro Timer

**Table 49 : Functional Test Table Set Pomodoro Timer**

| No. | Test case/Test script. | Attribute and value. | Expected result. | Actual result | Result |
|---|---|---|---|---|---|
| 1 | User can enter break duration work duration and number of sessions | Work Duration=25<br>Break Duration=5<br>Sessions=4 | Pomodoro is set and can be started | Pomodoro is set and can be started | Pass |
| 2 | User can View History |  | History of Pomodoro can be Viewed in the history tab bar | History of Pomodoro can be Viewed in the history tab bar | Pass |
| 3 | User can Start and Stop a Timer |  | Tiner is paused and played | Tiner is paused and played | Pass |

**Functional Testing 6:** Focus Mode

**Table 50 : Functional Test Table Focus Mode**

| No. | Test case/Test script. | Attribute and value. | Expected result. | Actual result | Result |
|---|---|---|---|---|---|
| 1 | User is | Permission granted | User is moved to the | User is moved to | Pass |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  | prompted to allows permission to the app |  | focus mode screen where he/she can choose from different mode | the focus mode screen where he/she can choose from different mode. |  |
| 2. | User selected option is activated and working | Focus mode activated | notifications are blocked | notifications are blocked | Pass |
| 3. | User can deactivate Focus Mode | Focus mode deactivated | notifications are unblocked | notifications are unblocked | Pass |

**Functional Testing 7:** Habit Tracker

**Table 51: Functional Test Table  Habit Tracker**

| No. | Test case/Test script. | Attribute and value. | Expected result. | Actual result | Result |
|---|---|---|---|---|---|
| 1 | User can create new habit | Habit name: ‘Running” | A new habit is created and is showing in the list | A new habit is created and is showing in the list | Pass |
| 2 | User can mark habit as done | Habit name: ‘Running.”<br>Status: Done | Habit status is updated and shown on heat map | Habit status is updated and shown on heat map | Pass |

**Functional Testing 8: Chat with Gemini**

**Table 52 : Functional Test Table Chat With Gemini**

| No. | Test case/Test script. | Attribute and value. | Expected result. | Actual result | Result |
|---|---|---|---|---|---|
| 1 | User can chat with Gemini | User Message: “Hello: | Gemini should send an appropriate response | Gemini response is received | Pass |

## 6.3 Business Rules Testing

**Table 53 : Business Rule Test Table**

| Rule | condition | True | Error message |
|---|---|---|---|
| 1 | Duplicate email while creating account |  | ✓ |
| 2 | Wrong Credential while sign in or Sign Up |  | ✓ |
| 3 | Tasks with earlier deadlines are displayed first | ✓ |  |
| 4 | Completed tasks are displayed in done tab. | ✓ |  |
| 5 | Archived tasks are hidden from the main task list | ✓ |  |
| 6 | Task and to-dos are searchable | ✓ |  |

| | | | |
|---|---|---|---|
| 7 | Pomodoro timer switches between work and break sessions as expected | ✓ | |
| 8 | Events and tasks are displayed on the correct dates | ✓ | |
| 9 | Users receive reminder on correct time | ✓ | |
| 10 | Habit progress is displayed and updated correctly. | ✓ | |
| 11 | Focus mode is activated, and distractions are minimized | ✓ | |
| 12 | Response from Gemini is Valid and Readable. | ✓ | |

## 6.4 Integration Testing

**Integration Testing 1:** Scheduling Event and Performing Tasks

**Testing Objective:** To ensure the schedule event will be visible on the event date in the to-dos list and user will also be able to view it in the to-do list.

**Table 54 : Integration Testing Table Scheduling Event and Performing Tasks**

| No. | Test case/Test script. | Attribute and value. | Expected result. | Actual result | Result |
|---|---|---|---|---|---|
| 1. | User schedule event on Calendar for today and the Event is Now Visible in the tasks | Category: Work<br>Title: Work on UI<br>Desc: Improve Theme<br>Time: Today 6:00 pm | Task should be listed in the Todos, and user should be able to mark it has done or edit or delete it | Task is visible and user can mark it has done or edit or delete it | Pass |

# 7 Conclusion and Future Work

## 7.1 Conclusion

In conclusion, the productivity app emerges as a versatile and comprehensive solution crafted to address the multifaceted challenges of modern-day productivity. With a meticulous blend of innovative features, intuitive design, and seamless functionality, it aims to revolutionize how users manage tasks, organize their time, and optimize their workflows.

At its core, the productivity app serves as a centralized hub for users to orchestrate their daily activities efficiently. Through robust task management capabilities, users can effortlessly create, prioritize, and track tasks, ensuring nothing falls through the cracks. The app's intuitive interface empowers users to categorize tasks, set deadlines, and monitor progress, fostering a sense of control and accomplishment.

Moreover, the app's focus mode feature provides users with a conducive environment for deep work and uninterrupted concentration. By temporarily blocking distractions and notifications, users can immerse themselves in their tasks and enhance productivity levels. The integration of a pomodoro timer further enhances time management by breaking work into focused intervals, interspersed with short breaks, promoting sustained productivity and mental agility.

The productivity app also recognizes the importance of fostering positive habits and routines to drive long-term success. With its habit tracking functionality, users can establish, monitor, and reinforce desired behavior.Through visual progress tracking users gain valuable feedback and motivation to stay on track towards their goals.

In addition to task management and habit tracking, the productivity app offers robust note-taking capabilities to capture ideas, insights, and inspirations on the go. Whether it's jotting down thoughts, capturing images, users can effortlessly organize and access their notes across devices, ensuring seamless continuity and accessibility.

Furthermore, the app's calendar integration empowers users to plan and manage their schedules effectively. By syncing events, appointments, and deadlines, users can stay ahead of their commitments and make informed decisions about how to allocate their time. The app's reminder feature ensures that users never miss important dates or deadlines, enhancing reliability and accountability.

Through seamless integration with external productivity tools and APIs like the Gemini API, the productivity app expands its capabilities, offering users access to a wealth of additional features and resources. Whether it's retrieving information, automating tasks, or accessing personalized assistance, users can leverage these integrations to enhance their productivity and efficiency further.

In essence, the productivity app represents a paradigm shift in how individuals approach productivity and time management. By combining cutting-edge technology with user-centric design principles, it empowers users to take control of their schedules, maximize their potential, and achieve their goals with confidence. As we continue to refine and expand its capabilities, the productivity app remains committed to supporting users on their journey towards greater productivity, fulfillment, and success.

## 7.2 Future Work

In the realm of future work, the productivity app harbors immense potential for growth and evolution, paving the way for enhanced functionality, expanded features, and enriched user experiences. Here are some avenues for future development and improvement:

1. **Team Collaboration and Task Management:** One of the primary directions for future work involves extending the app's capabilities to facilitate team collaboration and project management. By implementing features such as shared task lists, collaborative workflows, and real-time communication channels, the app can empower teams to work together seamlessly, streamline coordination, and boost collective productivity.

2. **Advanced Habit Analysis and Personalization**: Building upon the existing habit tracking functionality, future iterations of the app could delve deeper into habit analysis and provide users with personalized insights and recommendations. Leveraging machine learning algorithms and data analytics, the app can offer tailored suggestions for habit formation, behavior modification, and goal achievement, thereby fostering continuous improvement and self-development.

3. **Personalized AI Assistant**: In future iterations, the AI assistant within the productivity app can undergo further refinement and personalization. By leveraging user preferences, behavior patterns, and historical data, the AI assistant can offer tailored recommendations, proactive suggestions, and contextually relevant insights. Users can customize the assistant's behavior, preferences, and interaction styles to align with their individual needs and preferences, fostering a more personalized and intuitive user experience. This personalized AI assistant can serve as a trusted companion, offering invaluable support, guidance, and assistance as users navigate their productivity journeys..

# 8. References

[1] 31 Best Productivity Apps & Tools To Work Faster In 2023: https://hive.com/blog/productivity-apps/

[2] The best productivity apps in 2023:
https://zapier.com/blog/best-productivity-apps/

[3] Microsoft Todo
todohttps://play.google.com/store/apps/details?id=com.microsoft.todos&hl=en&gl=US&pli=1

[4] Dart Packages. https://pub.dev/

[5] Flutter documentation. https://docs.flutter.dev/