**Prodigenius**

**Software Requirements Specification**

Version 1.0



**Group Id: F24PROJECTA13FA**

**Supervisor Name: Muhammad Bilal**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| 03/12/2024 | 1.0 | Introduction of the project | BC190410762,BC190409617 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. [Scope (of the project)](#scope)
2. [Functional Requirements Non Functional requirements](#FRNFR)
3. [Use Case Diagram](#UCD)
4. [Usage Scenarios](#UCS)
5. [Adopted Methodology](#Adopted)
6. [Work Plan (Use MS Project to create Schedule/Work Plan)](#Gantt)

**SRS Document**

**Scope of Project:**

The scope of the **Prodi Genius** project encompasses the development of an AI-powered task management mobile application using Flutter, integrating basic machine learning functionalities for enhanced user experience. The app is designed to streamline task management by offering features such as task creation, categorization, prioritization, scheduling, and productivity tracking. Users can manually input tasks, assign categories (e.g., work, study, or personal), and set due dates. To enhance efficiency, the app incorporates AI-powered task prioritization using Firebase ML Kit, which evaluates urgency and deadlines to suggest optimal task execution orders. Furthermore, it leverages TensorFlow Lite's pre-trained models to estimate task durations based on type, helping users better allocate their time.

A standout feature is the basic AI-driven scheduling mechanism, which uses Firebase ML Kit to analyze user availability and task importance, offering tailored scheduling recommendations. This includes prioritizing high-importance tasks or allocating shorter tasks for quick completion. To keep users on track, the app employs Flutter Local Notifications to send smart reminders for high-priority tasks and those nearing deadlines. Additionally, the app tracks task completion rates using Firebase ML Kit and provides simple productivity insights, such as identifying peak productivity days or weekly task trends.

The project also includes non-AI elements, such as progress visualization through intuitive charts and dashboards created with Flutter Charts, showcasing completed tasks, pending items, and overall productivity metrics. The application focuses on leveraging free and pre-trained AI models to ensure accessibility, avoiding complex custom models or expensive cloud dependencies. By offering a simple, cost-effective solution, Prodi Genius aims to enhance productivity and task management for users while maintaining a clean and user-friendly interface.

**Functional and non Functional Requirements:**

**Functional Requirements for Each User**

**1. End Users (Task Managers)**

End users are individuals who use the app to manage their tasks and schedules. Their functional requirements are:

**Task Management**

* Ability to manually create tasks with details such as title, description, category, and due dates.
* Option to edit, delete, or mark tasks as completed.
* Categorize tasks into predefined or custom categories (e.g., work, study, personal).

**Task Prioritization**

* Set task priority levels (high, medium, low) manually.
* View AI-suggested task priorities based on deadlines and urgency.

**Task Duration Estimation**

* View AI-estimated time required for task completion based on the task type.

**Task Scheduling**

* Receive personalized scheduling recommendations from the app based on task priorities and available time.
* Adjust or override AI-suggested schedules manually.

**Reminders and Notifications**

* Get smart reminders for approaching deadlines or high-priority tasks.
* Customize notification preferences (e.g., frequency, timing).

**Productivity Tracking**

* View daily, weekly, and monthly reports on task completion rates.
* Receive AI-generated insights such as peak productivity days and trends.

**Dashboard and Visualizations**

* Access a visual dashboard showing task statuses (completed, pending) and progress trends.
* View simple charts and graphs to analyze productivity.

**2. Administrators (App Developers/Managers)**

Administrators oversee the application’s operation and ensure a seamless user experience. Their functional requirements include:

**User Management**

* Monitor active users and manage user access or accounts if necessary.
* Track user activity trends (e.g., feature usage, app engagement rates).

**AI Model Updates**

* Integrate and test updates for AI models used in task prioritization, scheduling, and duration estimation.
* Ensure AI models remain lightweight and on-device compatible.

**Bug Reporting and Fixing**

* Monitor and resolve user-reported bugs or app crashes.
* Implement diagnostic tools to identify and debug performance issues.

**Data Management**

* Manage task data storage securely, ensuring user data integrity and privacy.
* Monitor data usage for on-device and local storage capacities.

**Notifications Management**

* Configure the default settings for notifications and reminders.
* Monitor the performance and reliability of the notification system.

**Performance Optimization**

* Test and optimize app performance to handle multiple users and large datasets.
* Ensure compatibility with newer versions of Android and iOS operating systems.

**Feature Updates and Enhancements**

* Roll out periodic updates for app features and add new functionalities.
* Gather user feedback to prioritize feature improvements.

**Non-Functional Requirements**

1. **Performance**
   * The app should load tasks and respond to user interactions within 1-2 seconds.
   * AI-powered features like task prioritization and scheduling should complete computations within 3-5 seconds.
2. **Usability**
   * The interface must be user-friendly, with clear navigation and minimal learning curve.
   * Users should easily input tasks, view schedules, and understand productivity insights.
3. **Reliability**
   * The app should function smoothly without frequent crashes, especially during AI computations.
   * Task data and user settings must remain intact in case of unexpected shutdowns.
4. **Portability**
   * The app must be compatible with both Android and iOS devices, ensuring smooth cross-platform performance.
5. **Scalability**
   * The system should handle an increasing number of tasks and user interactions without significant performance degradation.

Use Case Diagram(s):

<Provide here the use case diagram of your system>

Usage Scenarios:

<Provide here the usage scenarios of all use cases in table format explaining Use Case title, Use Case Id, Actions, Description, Alternative Paths, Pre and Post Conditions, Author, Exceptions. You are supposed to provide a usage scenario for each of use case shown in use case diagram>

Adopted Methodology

<Write here detail about the adopted methodology for your project. You must choose VU Process Model which is a combination of waterfall and spiral models. More or combining two or more methodologies is explained in SE2 (CS605) handouts. >

Work Plan (Use MS Project to create Schedule/Work Plan)

<Provide Gantt chart of your final project>