**SRS DOCUMENT OF**

**DOCKET**

**(A JAVA BASED APPLICATION FOR THE PREPRATION FOR THE EXAM OR ANY EVENT)**

**INTRODUCTION**

The application, which is intended for college students, aims to assist with the management of their study schedule by providing a personalized study plan based on the syllabus, exam dates, and other academic events. The application takes into account the time remaining before an event, such as an exam, and suggests an appropriate time to start preparing for it.

The application allows users to input parameters such as the time they typically take to complete a module or unit of a subject. By taking these parameters into account, the application can calculate an approximate time that will be required to complete a task and suggest daily targets to help the user stay on track.

To achieve this, the application uses an algorithm that takes into account the syllabus, the dates of the exams and other events, along with the user-specified parameters. The algorithm then calculates the number of days remaining before an event, and based on the user's input it suggests a time to start preparation and sets a target for them.

**summary**

the application is designed to help college students manage their study schedule by providing a personalized study plan that takes into account the syllabus, exam dates, and other academic events. The application's algorithm calculates an appropriate time to start preparing for an event, and suggests daily targets to help users stay on track and reach their goals. Additionally, the application provides a reminder feature to keep the user on schedule and prevent them from forgetting important tasks.

**Functional Requirements**

* The application should allow users to input the syllabus, exam dates, and other academic events.
* The application should be able to calculate the number of days remaining before an event.
* The application should be able to suggest an appropriate time to start preparing for an event based on the number of days remaining and user-specified parameters.
* The application should be able to set daily targets for the user based on the time remaining before an event and user-specified parameters.
* The application should allow users to input their daily study progress and adjust the daily targets accordingly.
* The application should provide a reminder feature for upcoming exams, assignments, and other important events.

**Non-Functional Requirements**

* The application should have a user-friendly interface.
* The application should be compatible with different devices and operating systems.
* The application should have minimal load time.
* The application should be responsive and easy to navigate.
* The application should be scalable to accommodate future expansion and updates.

**System Requirements for the Java-based application**

1. Operating System: Windows, Mac, or Linux
2. Hardware: A modern computer with at least 2GB of RAM and a 64-bit processor
3. Java Development Kit (JDK): The latest version of JDK should be installed on the system
4. Java Runtime Environment (JRE): The latest version of JRE should be installed on the system
5. Software: A Java Integrated Development Environment (IDE) such as Eclipse, IntelliJ IDEA, or NetBeans
6. Internet connection: The application may require an internet connection to access certain features or update the data.
7. Additional Libraries or frameworks: The application may require additional libraries or frameworks like SWING for GUI, JPA for ORM, or any other library or framework as per the requirement of the application.

**CONSTRAINTS**

1. Input: The application must be able to accept a syllabus in the form of a PDF or text document as input.
2. PDF Reading: The application must be able to read the PDF document and extract the syllabus content.
3. Algorithm: The application must have an algorithm that can calculate an appropriate time to start preparing for an event, and suggest daily targets to help users stay on track and reach their goals.
4. Reminder Feature: The application must have a reminder feature to keep the user on schedule and prevent them from forgetting important tasks.
5. User-Friendly Interface: The application must have a user-friendly interface that is easy to navigate and use for students.
6. Platform: The application must be compatible with commonly used platforms such as Windows, Mac, and Linux.
7. Error Handling: The application must have proper error handling mechanisms in place to handle any unexpected situations.
8. Security: The application must have secure mechanisms in place to protect the user's data and information.