System Requirements Specification

for

StressMeOut

Prepared by

Armaan badhan 20103102

Priyanshu meharada 20103103

Sparsh jandial 20103004

Gaganjot singh 20103124

1. Introduction

This document is prepared in order to determine the software requirement specification for “StressMeOut” bot for Discord.
Discord is a VoIP and instant messaging social platform. Users have the ability to communicate with voice calls, video calls, text messaging, media and files in private chats or as part of communities called "servers". A server is a collection of persistent chat rooms and voice channels which can be accessed via invite links. Discord servers are organized into topic-based channels where you can collaborate and share.
Discord bots are AI-driven tools that can help you automate tasks on your Discord server. They make it a lot easier to build a community that is truly engaged and can be used to moderate your server and provide additional functionality.
This bot is to help keep track of all the deadlines and quiz dates for students. Using this bot the entities can see the deadlines and quiz dates by using a single command in Discord. In order to gain an overview about this, firstly a description of “StressMeOut” bot is followed.

1.1 Purpose

The SRS is needed to evolve as the development of the software product processes. The purpose of this document is to give a complete description about how StressMeOut can be developed. This document is to provide information about what the software product is to do to students. In addition to these, it provides a basis for validation and verification.

1.2 Intended Audience

This document is intended for the developers working on this project and the college instructors to evaluate the project.

1.3 Project Scope

The name of the software is StressMeOut . StressMeOut is a bot on discord that helps in making the academic life of students a bit easier. The aim of this system is to simplify the tedious task of keeping track of all the numerous deadlines that are scattered throughout different google classrooms and make this little yet crucial part of college easier for students. The user will be of one type i.e students on discord . Every batch has a discord channel of their own on which every student of the said batch is present . The students can use the system by typing in a simple command on discord which will provide them with a table of all the deadlines and important dates .

2. Overall Description

In this section, background information about what type of requirements the system should have will be provided briefly.

2.1 Product Perspective

StressMeOut is a discord bot that students can use to maintain and check deadline records. It is a part of a larger system, it is not an independent system.

2.2 Product Features

There are numerous classrooms for different classrooms, which means students have to go through every classroom to check and keep track of every deadline and important dates. This application provides an easy interface through which students can easily keep track of deadlines.

2.3 User Classes and Characteristics

StressMeOut doesn't require any specific computer knowledge to use it except the developers and the administrators of it. Standard users are thought to be from any age, any gender, any nationality. On the other hand, administrators and potential developers need a high level of expertise to understand technologies.

2.4 Design and Implementation Constraints

Misinformation about these important dates of quizzes, countdown reminders & deadlines can affect a student's grade. There might be cases when there is some issue with the bot which can lead to academic trouble. Accessing the API of discord will be done very carefully and methodologically so no such case arises and the bot functions perfectly at all times.

3. System Features

3.1 System Feature 1

Shows deadlines and important dates in a tabular form with the simple command / StressMeOut

3.1.1 Description

The user can scroll through the list where the deadline of every assignment is shown next to a countdown timer.

3.1.2 Stimulus/Response Sequences

The user acknowledges the deadlines and can discuss about it on discord.

3.1.3 Functional Requirements

The capabilities that must be present in order to execute the use case is database of deadlines.

3.2 System Feature 2

A reminder warning notification is shown an hour before every deadline of assignments/quizzes.

3.1.1 Description

The user will get a notification an hour before every deadline on discord messaging interface.

3.1.2 Stimulus/Response Sequences

Initially uninformed user now gets aware of the deadline and the deadline table is shown unasked for.

3.3 System Feature 3

New entries(deadlines, important dates, quizzes) can be added. older entries can be modified or deleted.

3.1.1 Description

The administrator will give permission to certain users to become superusers who can further make changes to or modify the entries according to the probable changes in deadlines and changes in dates.

3.1.2 Stimulus/Response Sequences

The superuser will be able to see his/her new added entry.

3.1.3 Functional Requirements

The functional requirement will be that there has to be some kind of extension or change in the deadlines.

4. External Interface Requirements

4.1 User Interfaces

Various interfaces for StressMeOut could be:

1. Discord Portable Application
2. Discord Web Application

4.2 Hardware Interfaces

The System must run over the internet, all the hardware is required to be connected to the internet. As for e.g. Modem, WAN-LAN, Ethernet Cross-Cable.

4.3 Software Interfaces

The user needs discord web app or discord mobile portable application to interact with the system.

4.4 Availability

The system should be available at all times, meaning the user can access it using a web browser or mobile application only restricted by the down time of the server on which the system runs. In case of a hardware failure or database corruption, an error message will be sent. Also in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the administrator. Then the service will be restarted. It means 24 X 7 availability.

4.5 Accuracy

The system should be accurate.

4.6 Reliability

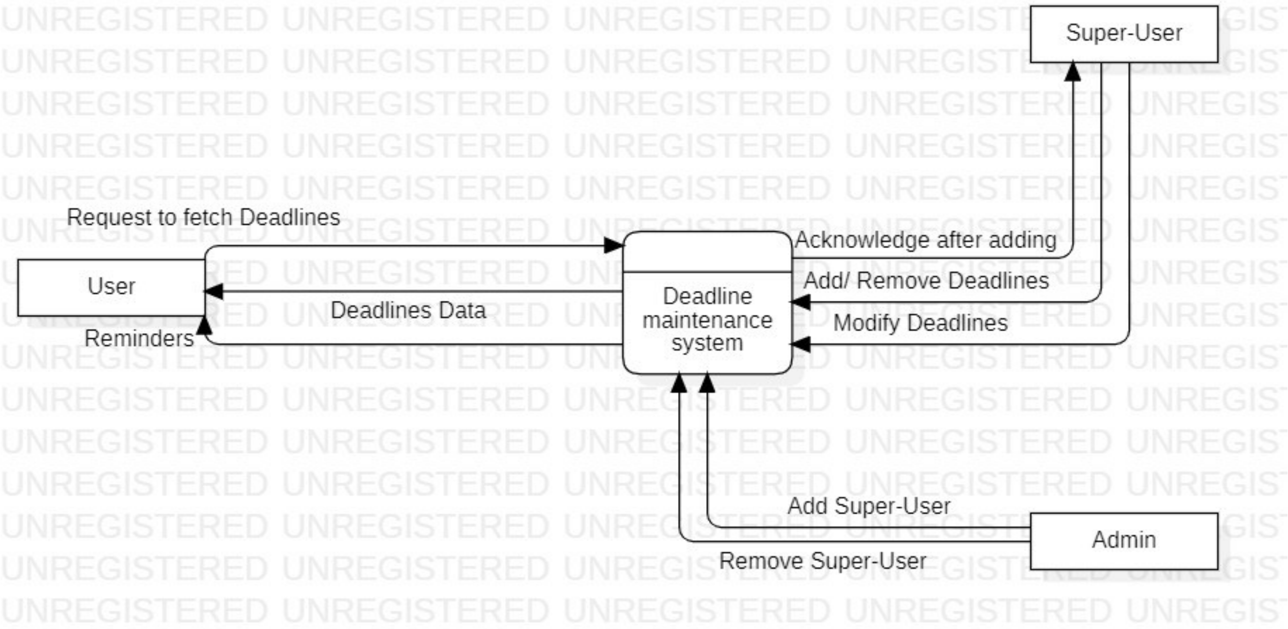
The system provides storage of all databases on redundant computers with automatic switch over. The reliability of the overall program depends on the reliability of the separate components. The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes. Thus the overall stability of the system depends on the stability of the container and its underlying operating system.

4.7 Portability

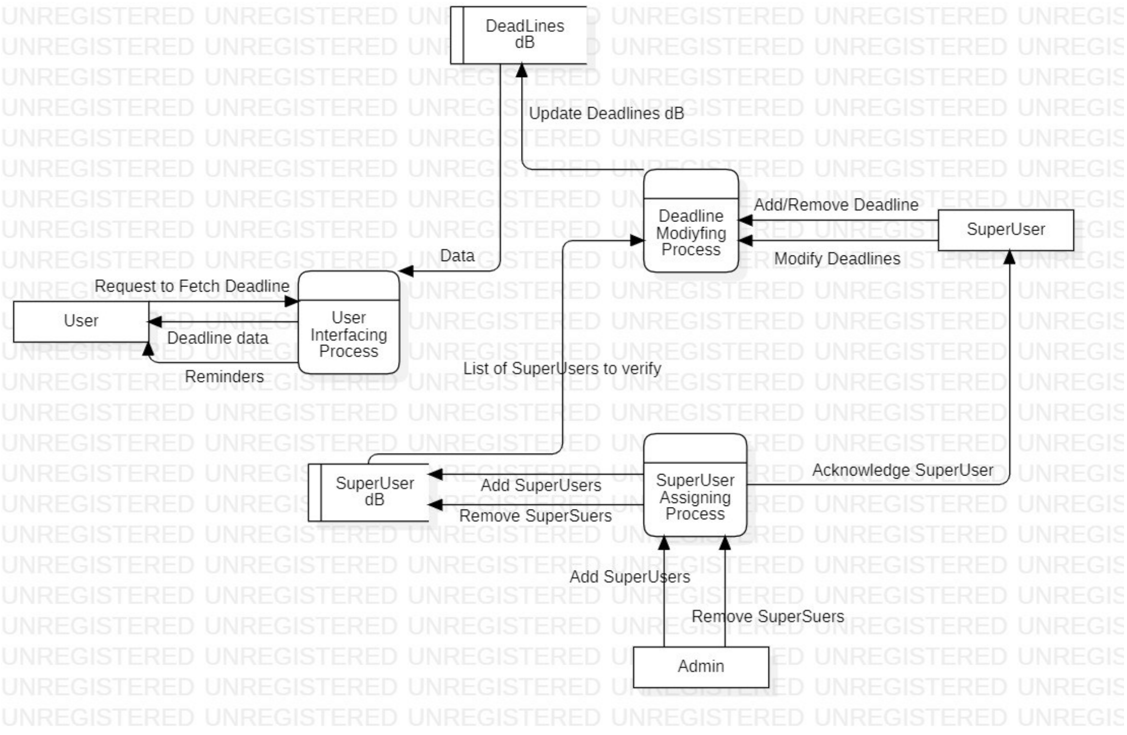
The application is python based. As the usage of the bot is throuh Discord, end user part is fully portable and any system using any web browser or mobile application should be able to use the features of the system, including any hardware platform that is available or will be available in the future. An end-user can use this system on any OS; either it is Windows or Linux. The system shall run on PC, Laptops etc.

Appendix A: Analysis Models

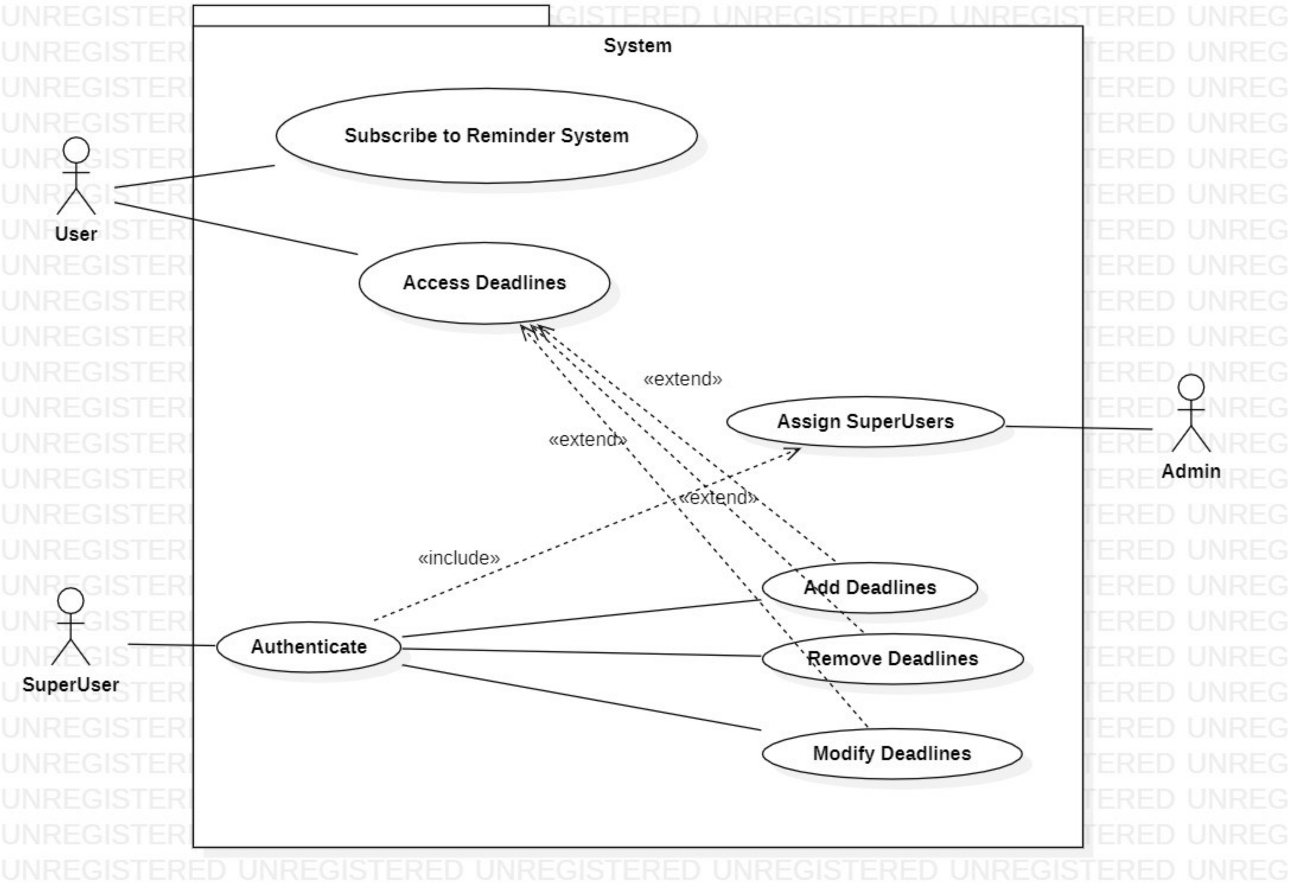
A.1 Level Zero Data Flow Diagram



A.2 Level One Data Flow Diagram



A.3 Use Case Diagram


System
Subscribe to Reminder System
User
Access Deadlines
«extend»
«extend»
Assign SuperUsers
Admin
«extend»
«include»
Add Deadlines
Authenticate
SuperUser
Remove Deadlines
Modify Deadlines