TimeDrop
Supplementary Specifications

Version <1.1>

Revision History

Date	Version	Description	Author
<10/09/2022>	<1.0>	First Draft	Sai Mittapalli, Mohamed Aql, Muhammad Momin Rahman, Abdalla Eltom

Table of Contents

Supplementary Specifications

1. Introduction

1.1 Purpose

The purpose of this document is to outline the requirements specifications, all of which are non-functional; likewise, it is also used to identify possible constraints when implementing these requirements.

1.2 Scope

The scope of this documentation includes all the non-functional specifications of the Calendar Software model with its corresponding requirements.

1.3 Definitions, Acronyms, and Abbreviations

API
Application Programming Interface: this is a software system allowing the server and client of a program to connect with each other through the form of requests.

Camel Case
A writing convention for all programming languages, to which the first word is lowercase followed by another word in the form of uppercase.

HTML
Hypertext Markup Language: a type of language describing the orientation, distribution, and dimensions of how a browser should appear in front-end programming.

IDE
Integrated Development Environment: a virtual facility providing software developers a platform on which coding operations may be performed on.

CSS
Cascading Style Sheets: an embedded language within HTML that styles the browser in an appealing and organized manner.

Refer to Glossary Document (upedu_gloss.dot)

1.4 References

1. UPEDU Template Examples

1.5 Overview

To complement the functional requirements, factors such as aesthetics, memory consumption, and performance are considered.

2. Assumptions and Dependencies

It is assumed that the user is familiar with using websites and web applications and that the data received from the user and Django Login Authentication is accurate.

3. Usability

3.1 User's Knowledge

The user is familiar with using an internet web browser and the system is easy enough for the user to navigate and use without any learning curve.

3.2 User Interface

The user will only interact and see the user interface so the user interface should be friendly and have all the necessary functions to be easily accessible with the use of buttons, headings, and other features. The user must also be able to have access to a keyboard and mouse in order to interact with the user interface.

3.3 User Performance

The system should have its performance optimized to be smooth and efficient. The user must be able to see all their desired information when they need, and the system shouldn't have unnecessary aspects that decrease the user performance.

4. Performance

This application should be able to run and function as long as the user has a stable connection to the HTTP. The application will execute on an internet web browser and the application shouldn't take up more than 500MB.

5. Supportability

The user can access the website using any modern internet web browsers. And along with that, camel case naming conventions will be used for all variables and file names for the software code.

6. Design Constraints

The programing languages used will be Python, JavaScript, HTML, and CSS for the web content and user interface and SQL will be used for the databases where all the user information is stored.

7. Security

Since the user provides their credentials like their username, password, and their email, the system must be secure and be able to store all this information in a database. All these need to be protected from getting hacked and being stolen. Along with that, the user's tasks and events stored in the calendars also need to be secure.

8. Interfaces

The user interface for this system is what the user will interact with most often to navigate through the system. The user interface will be friendly and easy to navigate to accomplish all the main functions. Along with that, the system will interact with the databases where the user information is stored through the User Information database interface.