Study Buddy

(Version 0.1 Beta)

- **PROJECT NAME:** Study Buddy
- **PHASE:** Detailed Requirements
- **DATE:** 10/21/2022
    - **Revision 01**: 12/09/2022
- **PREPARED BY:** DNM

Table of Contents

1.0 - Project Plan

1.1 DESCRIPTION/PURPOSE OF PROJECT:

The project will consist of creating an intuitive, and reliable task scheduling web app. The purpose of this product is to help post-secondary students with time management.

The key aspects include;

- To create a visual interface to aid students with scheduling and time management
- To categorise events based on relevant data, in an inclusive-intuitive manner
- To provide the user with multiple sorting options based on relevant data

1.2 SCOPE OF PROJECT:

The scope of the project entails developing a web application using a suitable programming language and tools to facilitate the use of the web application through multiple browsers and with the ability to store data online. The break down of the sope as follows;

- Development of a web application
- Web app to run on Google Chrome, Mozilla Firefox, Internet Explorer or any other modern browser
- Work with an online database
- Hosting a simple website

2.0 - Project Description

2.1 PROJECT STAKEHOLDERS

Based on the description and scope of the project, the main stakeholders are the end users, which are;

- Students enrolled in any post-secondary education program

2.2 PROJECT GOALS

The goal of this project is to fulfil shared goals for all stakeholders, which consist of having a reliable and easy to use task manager with deadlines and brief descriptions of each task. For these to be achieved;

- The product should have a user-friendly interface and intuitive controls
- The information displayed needs to be consistent and easy to read
- The final product in its first iteration should have enough features to aid students in scheduling tasks

2.3 CONSTRAINTS & RESTRICTIONS

The development of a web application with the specified description and scope, presents a set of constraints that are imperative to resolve prior to implementation. The constraints and planned solutions as follows:

CONSTRAINT	ISSUE/SOLUTION
User has multiple computers, and does not want to have separate instances of an application on each computer.	**Issue->** Stakeholder uses multiple devices. **Solution ->** Operate database and application through servers, service is accessed through a web browser.
Tier-one GUI contains a colour coded legend to categorise groups	**Issue->** Stakeholders may have visual impairments. **Solution ->** Addition of features to adjust colour coded text fields, ensure colours are always labelled.

2.4 ASSUMPTIONS AND DEPENDENCIES

For the development team to develop a product as described above, the team has compiled a list of assumptions based on the functionality that the product needs to provide to end users and the type of platforms that the end users may utilise in order to interact with the product.

- Users connected to the product are using a modern web browser without plugins that prevent JavaScript from being run.
- Google Cloud Services must be up for the database to work. (A Google cloud based database service called Firebase is used to store, sort and modify data).
- Azure services that run Githhub to host a website/domain must be up for the application to run. (Github is being used to host the web app.)
- The client and server must always have access to the internet for the application to work properly.
- The client must have a modern internet browser to access the website. (IE6 will not be supported)

3.0 - Project Requirements

There is only one primary user group in the system; post-secondary students.

3.1 FUNCTIONAL REQUIREMENTS

The functional requirements of the product are dictated by the intended behaviour of it for the end users through the product’s life cycle. The team has recognized that the functional requirements of a minimum viable product are as follows:

- The student should be able to add tasks, or groups to the to-do list.
- Each task needs to have the option to enter a description, type, group, and due date

- The web application will have an option to sort tasks based on their due-date.
- User will be able to select a colour when adding a group to the to-do list
- Default Types will exist as follows: Assignments, Tests, Projects, Labs, and Other
- A checklist field will exist that allows objects to be marked and then sorted as either completed or incomplete.
- A legend, indicating the colours of existing groups will be displayed on the homepage

3.2 USER INTERFACE REQUIREMENTS

For the UI, the team has identified some key aspects that will suit the basic user interface requirements and also need to satisfy the core requirements of the product, which are to provide an intuitive and easy to use task manager. The UI requirements dictate that;

- Options, controls, defaults, and the display of information must be concise, intuitive, and explicit
- Explicit error handling for when the user inputs wrong data in text boxes or leaves text boxes empty, such as; “Error, this field cannot be left empty”.

3.3 SYSTEM REQUIREMENTS

System requirements are not based on end-user requirements. Instead, they are mandated by the planned functionality of the product, which means that system requirements constrain the use of the product based on the following requirements:

- User’s system must have a modern web browser that supports the latest version of JavaScript, and HTML5.
- User must have an internet connection

3.4 ERROR HANDLING

Alongside the development of the product, sets of tests cases will be thought of and developed by the testing team, which will attempt to catch bugs in the code and cases where the user interacts with the product in a way that malfunctions, therefore, user input UI error handling will consist of;

- Handling user input of dates that are in the past relative to current date, if they have not entered a date, or the date is not in the specified format
- If the name of the task is left empty, an error message will be displayed and the task entry to the list will not be processed.
- If the user tries to add a pre-existing group an error message will be displayed reading "This group name ' group name ' is already in use".

Another important aspect of error handling considers the functional error handling which consist of;

- Handling the case that the database is not accessible. An error message will be displayed reading "Sorry, database is offline, please try again later"
- If user input data is corrupted, an error message "Database contains invalid data - data is corrupted."

4.0 - Use Cases & Scenarios

The intent of developing a set of product specific test cases, during the requirements gathering stage, allows the team to think of the details of the implementation in a holistic manner, at a high level, without getting into any particular details of the design or implementation. Furthermore, with a clear idea the minimum viable product within the project scope, the team was able to specify a set of preliminary test cases that describe user interactions with the UI and system responses, as follows:

- Add Task

User	System
User clicks the add task button.	Pop-up window is brought up, "add task" is displayed, a text input box is displayed for task type, task group and task due date.

User clicks the ok button.	Displays “Task has been added” and adds the task.

- Remove task

User	System
User clicks the remove task button	System prompts user for confirmation of remove
User confirms the request	System displays that the task has been deleted

- Complete Task

User	System
User clicks the complete task button.	System displays that the task has been moved to the completed section.

- Add group

User	System
User clicks add group button.	A popup appears, “add group” is displayed, a text box for the name of the group appears, and a colour wheel to select the colour of the group appears.
User clicks the ok button.	“Group has been added” is displayed, system adds group with colour choice to database.

- Sort

User	System

User clicks on the tasks heading.	The tasks column on the main page is sorted according to date due, and any task that has been added since the last sort is added to the display of tasks.

5.0 - Sign-offs

Signature: David Botero Date: 10/21/2022

Print Name: David Botero

Title: Student

Role: Test Lead, UI Co-Lead,

Signature: Joshua Lade Date: 10/21/2022

Print Name: Joshua Lade

Title: Student

Role: UI Lead, Test Co-lead

Signature: Hayden Lister Date: 10/21/2022

Print Name: Hayden Lister

Title: Student

Role: Contact Person,

Project Management

Signature: Abhay Parashar Date: 10/21/2022

Print Name: Abhay Parashar

Title: Student

Role: Version Control Lead,

Dev Lead,

Project Management Co-lead

Signature: Robert Stewart Date: 10/21/2022

Print Name: Robert Stewart

Title: Student

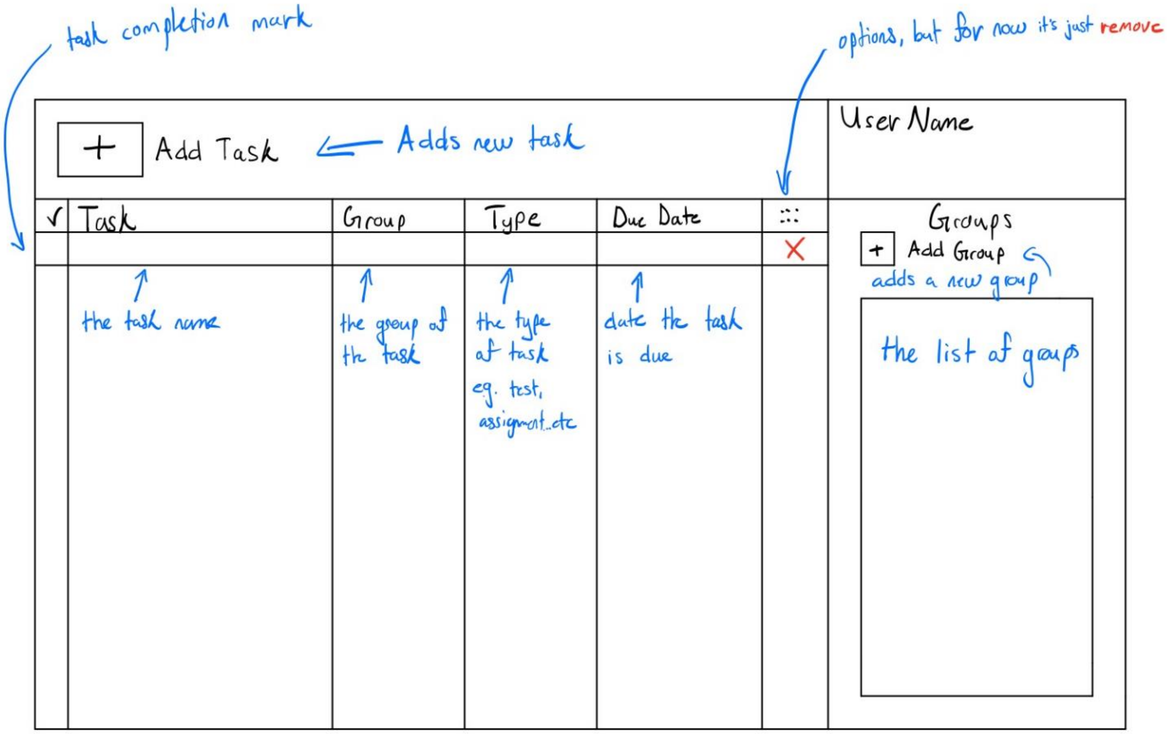
Role: Documentation, Dev Co-Lead

6.0 - Glossary

Group: A category that groups different tasks together who share a common goal. (e.g. a course)

Type: A particular kind of action that needs to be accomplished by a task.
Defaults: Assignments, Quizzes, Projects, Exams, Labs
Task: A specific action or activity that needs to be accomplished by a specific time.

7.0 - Appendix


task completion mark
options, but for now it's just remove
+ Add Task
Adds new task
User Name
✓
Task
Group
Type
Due Date
:::
✕
Groups
+ Add Group
adds a new group
the task name
the group of the task
the type of task
eg. test, assigmnt..etc
date the task is due
the list of groups

FIG_UI