**Dream Team**

**Smart Calendar**

**Written Requirements**

***Revision History***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Authors** | **Description of Change** | **Sections** | **Rev** | **Date** |
| Henry Yang | Initial Document |  | O | 2/28/18 |
|  |  |  |  |  |

**Table of Contents**

[1 Team Description 5](#_gjdgxs)

[2 Terminology 6](#_30j0zll)

[3](#_1fob9te) Smart Calendar 7

[3.1 Overview 7](#_3znysh7)

[3.1.1 <Section 1> 7](#_2et92p0)

[3.1.2 <Section 2> 7](#_tyjcwt)

# Team Description

|  |  |
| --- | --- |
| **Team Member Name** | **Email Address** |
| Eric Guzman | eg70889@csu.fullerton.edu |
| Adam Weesner | klaviam@gmail.com |
| Jesus Rodriguez | jrjssrdgz@csu.fullerton.edu |
| Michael Perez | mperez980089@yahoo.com |
| Henry Yang | hkyang@csu.fullerton.edu |
|  |  |

# Terminology

The following table defined terms used within this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| User | Person using the application. |
| Primary schedule | List of involuntary activities the user will engage in throughout the week. |
| Block | Segment of time represented on the calendar. |
| Task | A voluntary activity the user must complete. |
| Ripping the bandaid | Algorithm type; assigns tasks in large, segments. |
| Slow and steady | Algorithm type; assigns tasks in small, frequent segments. |
| Free time | The time where the user is available to schedule tasks. |

# Smart Calendar

## Overview

<give an overview of what is being described below. These should not be requirements>

Detailed Description: This application will allow the user to input their “primary schedule” for the week. The primary schedule is the user’s list of activities that represent the times they are unavailable to perform tasks. The activities are represented as “blocks” on the calendar’s UI. After inputting their primary schedule, the user can create “tasks” to be scheduled: activities that the user needs to complete. Tasks have a name, description, due date, list of start sessions, and estimated amount of time it takes. When a task is created, it is put into the “unassigned task” list. All tasks can have the system sort them into the freetime on the user’s calendar. Two algorithm types will be available: “Slow and steady”, which will cut the task into small segments and spread the segments out evenly, and “Ripping the bandaid”, which will assign the tasks in large segments. Users will be reminded an hour before each task start session that it is approaching. Users are able to add or subtract the time it takes to complete a task, as well as delete a task entirely.

### Input Primary schedule

The user shall enter in their primary schedule for every week.

### Edit primary Schedule

The user can edit their primary schedule.

**3.1.3 Create Task**

The user will be able to create a task. Each task will have to have a name, amount of time needed for that task, due date for the task.

**3.1.4 Assign task to Calendar**

Once the task is created the user will be able to allocate a the specific days and times he or she would like to work on the task that will fit around the primary schedule.

**3.1.5 Unassign currently assigned task to calendar**

The USer will be able to unassign a current task. The user can switch a task to do or assign the current task in another time slot.

**3.1.6 Modify Task Due Date**

The User will be able to modify the due date of all the tasks that are already created.

**3.1.7 Add Time**

The user will be able to add time to any of the tasks that are already created.

**3.1.8 Subtract Time**

The user will be able to subtract time to any of the tasks that are already created.

**3.1.9 Delete Task**

The user will be able to delete any of the tasks that are already created.

**3.1.10 Start Task Timer**

The user can start the timer for a specific task that is allocated to a certain time block.

**3.1.11 Stop Task Timer**

The user will be able to stop the task timer during a task and at the end of the task.

**3.1.12 Remind User of Upcoming Task Start Session**

The system will remind the user that a task time limit to be completed is going to be up and the user must finish the task before it is due.

**3.1.13 Alert User when due date is reached**

The System will alert the user that the task has expired and will ask user to delete or change the due date of the task if able.

**Guidelines** – remove these from your document

Write at least ten to twenty text requirements that describe the key features of your system. This description should encompass all important functionality of your system. A person reading the document should be able to understand what the system does without any external references. Assume the audience for this document has the typical knowledge of a software engineer. Do not reference any implementation details. Describe what the system does, not how it does it.

Use grammatically correct and concise language.

Use a uniform level of detail.

Once defined, terminology such as “the user” and “the system” are generally appropriate.

Reuse of material from use-case and class diagram is OK, but make sure it is all shaped into the proper form.

The word “shall” indicates a testable requirement. Bold this word.

Use indentation to clarify the structure of the documents.

Do not write “shall not” requirements. They are problematic to test.

AND Requirements

The action of inputting a primary weekly schedule shall be taken when **all of** the following conditions are met:

* Schedule has not yet been input

The action of creating a task shall be taken when **all of** the following conditions are met:

* User allots blocks of free time to the schedule
* User inputs “due date”, “time”, “algorithm type” and “label” into the prompt

The action of editing the user’s schedule shall be taken when **all of** the following conditions are met:

* At least one (1) block of free time is allotted

The action of assigning a task to calendar shall be taken when **all of** the following conditions are met:

* The user has one (1) or more task in their queue

The action of adding time to the task timer shall be taken when **all of** the following conditions are met:

* A task is selected
* The time of the task does not conflict with other tasks

The action of subtracting time to the task timer shall be taken when **all of** the following conditions are met:

* A task is selected
* The amount of time being subtracted does not make the timer count less than 0

The action of reminding the user of an upcoming task shall be taken when **all of** the following conditions are met:

* The task falls within a user defined interval of when they wish to be reminded

The action of alerting the user when a due date is reached shall be taken when **all of** the following conditions are met:

* The task has not been completed by the user
* The current system time matches or is later than that of the task due date

XYZ action shall be taken when **all of** the following conditions are met:

OR Requirements

The action of unassigning a task shall be taken when **any of** the following conditions are met:

* The user manually deletes the task
* The system timer finishes its countdown and deletes the task

The action of starting the task timer shall be taken when **any of** the following conditions are met:

* The task scheduled is at the current time and the calendar executes the timer
* The user manually executes the task timer

The action of stopping the task timer shall be taken when **any of** the following conditions are met:

* The task scheduled ends at the current time and the calendar finishes executing the timer
* The user manually stops the task timer

XYZ action shall be taken when **any of** the following conditions are met:

**Your system will be tested according to these requirements.**