**Dream Team**

**Smart Calendar**

**Use Case Report**

***Revision History***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Authors** | **Description of Change** | **Sections** | **Rev** | **Date** |
| Michael Perez | Initial use-case document creation |  | O | 2/18/2018 |
| Michael Perez | Updated use-case diagram and use-cases |  | 1 | 2/27/2018 |
| Michael Perez | Mega update which got rid of the timer system |  | 2 | 3/8/2018 |

**Table of Contents**

[1 Team Description 4](#_30j0zll)

[2 Project Description 4](#_1fob9te)

[2.1 Use Case Diagram 5](#_3znysh7)

[2.2 Use Case List 6](#_2et92p0)

[2.2.1 Use Case 1:](#_3dy6vkm) Create Free Time Block 6

[2.2.2](#_1t3h5sf) Use Case 2: Edit Free Time Blocks 7

2.2.3Use Case 3: Create Task 8

2.2.4 Use Case 4: Assign Task to Calendar 8

2.2.5 Use Case 5: Unassign currently assigned Task 9

2.2.6 Use Case 6: Modify Task Due Date 9

2.2.7 Use Case 7: Modify Task ETA 10

2.2.8 Use Case 8: Delete Task 10

2.2.9 Use Case 9: Remind user of upcoming Task Block 11

2.2.10 Use Case 10: Alert user when the end of the last Task Block is reached 11

# Team Description

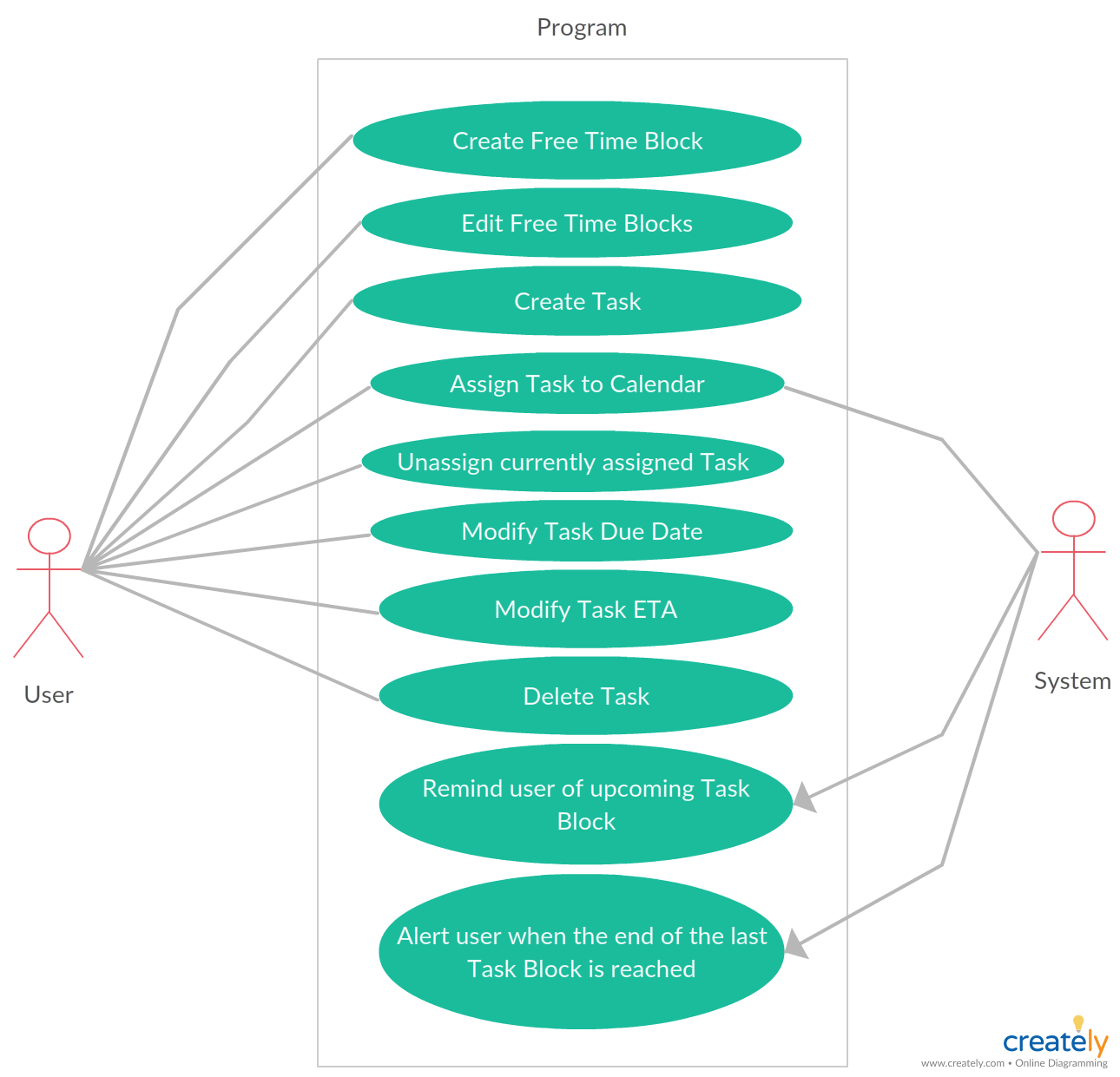
|  |  |
| --- | --- |
| **Team Member Name** | **Email Address** |
| Eric Guzman | eg70889@csu.fullerton.edu |
| Adam Weesner | klaviam@gmail.com |
| Jesus Rodriguez | jrjssrdrgz@csu.fullerton.edu |
| Michael Perez | mperez980089@yahoo.com |
| Henry Yang | hkyang@csu.fullerton.edu |
|  |  |

# Project Description

Statement of Purpose: To make it convenient for users to schedule and keep track of their progress on activities.

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.

## Use Case Diagram



## Use Case List

|  |  |  |
| --- | --- | --- |
| **Use Case** | | |
| **Sequence Number** | **Actor** | **Goal** |
| 1 | User | Create Free Time Block |
| 2 | User | Edit Free Time Blocks |
| 3 | User | Create Task |
| 4 | User | Assign Task to Calendar |
| 5 | User | Unassign currently assigned Task |
| 6 | User | Modify Task Due Date |
| 7 | User | Modify Task ETA |
| 8 | User | Delete Task |
| 9 | User | Remind user of upcoming Task Block |
| 10 | User | Alert user when the end of the last Task Block is reached |

### Use Case 1: Create Free Time Block

Primary Actor: User

Secondary Actor(s): None

Goal in Context: To define a segment of free time on the Calendar that Tasks can fill.

Preconditions: None

Additional Description: Users can press a button on the Calendar to create a new Free Time Block. They’re prompted to enter in a Start Time and an End Time, which is then graphically placed on the Calendar once they press OK.

2.2.1.1 Scenario 1: Setting the created Free Time Block’s Start and End Time to an invalid time value

Upon pressing OK, system displays an error and erases their entries. Invalid time values include the following:

* Start Time or End Time occupies the same space as an existing Task Block
* Start Time and End Time encapsulate a Task Block
* Start Time and End Time are encapsulated by an existing Free Time Block

2.2.1.2 Scenario 2: The created Free Time Block encapsulates an existing Free Time Block

System adds the Free Time Block, and automatically deletes the encapsulated Free Time Block.

2.2.1.3 Scenario 3: The Start Time is within an existing Free Time Block, and the End Time is outside of it (or vice versa)

System merges the existing Free Time Block and the new Free Time Block.

### Use Case 2: Edit Free Time Blocks

Primary Actor: User

Secondary Actor(s): None

Goal in Context: To edit the Free Time Blocks currently on the Calendar

Preconditions: Free Time Blocks exist on the Calendar

Additional Description: Users can select Free Time Blocks on the calendar and edit their Start/End Times.

#### 2.2.2.1 Scenario 1: Editing the Free Time Block’s Start/End Time to an invalid time value

Upon pressing OK, system displays an error and erases their entries. Invalid time values include the following:

* Start Time or End Time occupies the same space as an existing Task Block
* Start Time and End Time encapsulate a Task Block
* Start Time and End Time are encapsulated by an existing Free Time Block

2.2.2.2 Scenario 2: The modified Free Time Block encapsulates an existing Free Time Block

System modifies the Free Time Block, and automatically deletes the encapsulated Free Time Block.

2.2.1.3 Scenario 3: The modified Start Time is within another Free Time Block, and the modified End Time is outside of it (or vice versa)

System merges the other Free Time Block and the modified Free Time Block.

**2.2.3 Use Case 3: Create Task**

Primary Actor: User

Secondary Actor(s): None

Goal in Context: To create a new Task

Preconditions: None

Additional Description: The user can press the “New Task” button on the calendar, and will then be prompted to enter a Name, Description, Due Date, ETA, Scheduling Algorithm, and Color. The created Task will then be placed into the “Unassigned Tasks” list.

* Name is a text field that accepts any string of 256 characters
* Description is a text field that accepts any string of 256 characters
* Due Date consists of
* ETA is a text field that accepts an integer
* Scheduling Algorithm is a drop down menu with the options “Slow and Steady” and “Ripping the Bandaid”
* Color is a drop down menu with various curated colors to pick from

**2.2.4 Use Case 4: Assign Task to Calendar**

Primary Actor: User

Secondary Actor(s): System

Goal in Context: To assign a Task from the “Unassigned Tasks” list to the Free Time Blocks on the Calendar depending on which algorithm the Task uses

Preconditions: Task exists and is currently unassigned

Additional Description: Users can select a specific Unassigned Task and press a button to have the system place the Task somewhere within their free time. The “Slow and Steady” algorithm splits up the Task into smaller Blocks throughout the week, while the “Rip the Bandaid” algorithm tries to keep the Task in large Blocks.

2.2.4.1 Scenario 1: Task’s ETA is longer than the amount of available free time on the Calendar leading up to the Due Date

Upon pressing ASSIGN, system displays an error and prevents the action, requesting the user either extend the Due Date of the Task, or shorten the ETA.

**2.2.5 Use Case 5: Unassign currently assigned Task**

Primary Actor: User

Secondary Actor(s): System

Goal in Context: To unassign a Task that is currently assigned on the Calendar

Preconditions: Task exists and is currently assigned

Additional Description: Assigned Tasks on the Calendar can be selected and unassigned. They retain all of their information, including the ETA, as well as their Due Date.

**2.2.6 Use Case 6: Modify Task Due Date**

Primary Actor: User

Secondary Actor(s): None

Goal in Context: To change the Due Date of a specific Task

Preconditions: Task exists and is currently unassigned

Additional Description: Users can select Tasks from the “Unassigned Tasks” list and modify their Due Date.

**2.2.7 Use Case 7: Modify Task ETA**

Primary Actor: User

Secondary Actor(s): None

Goal in Context: To modify a specific Task’s ETA value

Preconditions: Task exists and is currently unassigned

Additional Description: Users can select Tasks from the “Unassigned Tasks” list and modify their ETA.

**2.2.8 Use Case 8: Delete Task**

Primary Actor: User

Secondary Actor(s): None

Goal in Context: To delete a Task

Preconditions: Task has been created

Additional Description: Users can select Tasks from the “Unassigned Tasks” list, or from the Calendar itself, and request to delete them. The system will ask the user if they’re sure they want to delete it, and upon clicking yes, the Task will be deleted.

**2.2.9 Use Case 9: Remind user of upcoming Task Block**

Primary Actor: System

Secondary Actor: None

Goal in Context: To remind the user one hour before a Task Block on the Calendar that they need to start on it in an hour

Preconditions: The Calendar is populated with at least one Task

Additional Description: Reminders are simple pop-up boxes that are dismissed by pressing “OK” on the box.

**2.2.10 Use Case 10: Alert user when the end of the last Task Block is reached**

Primary Actor: System

Secondary Actor: None

Goal in Context: To alert the user when the end of the last Task Block is reached

Preconditions: Task exists in Calendar

Additional Description: Upon reaching the end of the last Task Block, the system asks the user if they have finished or are very close to finishing the Task. If the user selects “Yes”, the Task is deleted. If the user selects “No”, the Task is placed into the “Unassigned Tasks” list, where the system then requests the user go and modify the ETA and Due Date values for rescheduling.