**Section I: Goals**

In this sprint, we want to implement a calendar style view for displaying daily tasks and create a page to display completed tasks with updated duration of each task.

**Section II: Tasks for Each User Story**

1. **As an employee, I need to keep plan hourly schedules so that I know what to do in every hour of a day.**

| **Task Number** | **Time Estimate**  **(hours)** | **Story Point Estimates** | **Description** |
| --- | --- | --- | --- |
| 1 | 4 | 3 | Add End dates to task object |
| 2 | 3 | 2 | Research API for Calendar use |
| 3 | 3 | 2 | Pull tasks from database to populate calendar |
| 4 | 4 | 3 | Make “database-utils.js” function that READS tasks from db to server |
|  | 5 | 3 | Dynamically update new calendar view when creating new task |
|  | 6 | 5 | Integrate Calendar API onto homepage |
| 6 | —-  (See below) | **5** | **Total additions to “object.js” file to mock database for tasks.js and “verify.js” files.** |
|  | 3 | 2 | * Add member class (DB) |
|  | 3 | 2 | * Add taskpreset class (DB) |
|  | 3 | 2 | * Add taskscheduled class (DB)   + LOOK AT THE schema.sql in the SQL folder in the backend. |
|  | 3 | 2 | * Create tasklist (DB) |
|  | 2 | 1 | DELETING task dbUtils.js button |
|  | 1 | 1 | Change date format sending from frontend to backend |
|  | 4 | 3 | Get task id system in place btn front end and back end |
|  | 3 | 2 | Research React&Node Testing Frameworks |
|  | 8 | 8 | Set up frameworks for testing |
|  | 4 | 3 | Connect updateTask() to backend&DB |
| Total: | 57 |  |  |

1. **As a student, I need to record how I use my time by the hour so that I can keep track of what I've done at the end of the day.**

| **Task Number** | **Time Estimate**  **(hours)** | **Story Point Estimates** | **Description** |
| --- | --- | --- | --- |
| 1 | 4 | 3 | Create new Dashboard Page |
| 2 | 6 | 5 | Add create task function that creates new task in front end and back end |
| 3 | 4 | 3 | Add “database.js” function that CREATES new task in database |
|  | 4 | 3 | Add “database.js” function that UPDATES task in database |
| 4 | 2 | 1 | Add routes to dashboard page |
| 5 | 3 | 2 | Pass user & task list data to dashboard page from homepage |
| 6 | 2 | 1 | Add button to home page for navigation to dashboard |
| 7 | 2 | 1 | Add completed field to task object in server |
| 8 | 4 | 3 | Add completed field to task in DB |
| 9 | 6 | 5 | Create cards to display each task including start/end times and completed bool |
| 10 | 4 | 3 | Allow cards to edit each tasks times and if it's completed |
| 11 | 4 | 3 | Updated new information to server |
| 12 | 6 | 5 | Save updated task info from server to DB |
|  | 2 | 1 | Standardize button styles |
|  | 6 | 5 | Add Error Handling (field verification) to Create Account Page |
|  | 4 | 3 | Add a home button on the login & register page & dashboard page |
|  | 4 | 3 | Clear form after submitting tasks |
| Total: | 48 |  |  |

**Section III: Team Member Roles**

| **Team Member** | **Roles** |
| --- | --- |
| **Kalani Richard** | Scrum Master |
| **Vlad Lekhtsikau** | Team Member |
| **Collin McColl** | Team Member |
| **Gene Park** | Team Member |
| **Simeon Tran** | Product Owner |

**Section IV: Initial Task Assignment**

| **Team Member** | **User Story** | **Initial Tasks** |
| --- | --- | --- |
| **Kalani Richard** | 1 | Design and Research Calendar API |
| **Vlad Lekhtsikau** | 1 | Add End dates to task object |
| **Collin McColl** | 1 | Make “database-utils.js” function that READS tasks from db to server |
| **Gene Park** | 1 | Add member class (DB) |
| **Simeon Tran** | 1 | Update task list view |

**Section V: Initial Burnup Chart**

The Burnup Chart is linked [here](https://docs.google.com/spreadsheets/d/1eT2X4jkW8zJl4sqHub6wWgTYjLR5e6xgqt6CDqwLX3g/edit?usp=sharing).

**Section VI: Initial Scrum Board**

The Scrum Board is linked [here](https://app.asana.com/0/1203076727968591/1203076727968591).

**Section VII: Scrum Times**

* Mondays 9:10 am to 9:35 am
* Wednesday 9:10 am to 9:35 am
* Fridays 9:10 am to 9:35 am
* **TA Meeting Time:** Wednesdays 2:30 pm - 3:30 pm