# Team

1. James Taylor; Github: SolarAnomaly

2. Vinh Nguyen; Github: davinhster

3. Steven Luu; Github: BayAreaSjsu

4. Angelo Vino Ibarrola; Github: aibarrola

# Product Name: "Schedule-Master"

## Problem Statement

Digital calendars are incredibly useful for setting up reminders for one-time events, as well as annual events, but digital calendars fall behind when it comes to a platform that can allow multiple users to add, delete, and modify non-overlapping events.

## Product Objective:

To design and develop a reliable and easy-to-use scheduling application that allows users with accounts to publish and edit their availability times, and users without accounts to book meetings or plan events based on those availability times.

## Functional Requirements

- The Main page should allow the Creator to log in or create an account.

- The "*Create Account*" option for Creators should have a register page that prompts them to create a username and password while also entering a valid email.

- The "*Log In*" option should have a login page that takes a valid username and password.

- The "*View Events*" option should open a page that shows the Creator's past and future events.

- The "*Settings*" option should open a page that allows the Creator to delete their account (that purges any information from the database), add availability times, and reset their password.

- The Creators "*Availability*" setting should open a page that allows the Creator to specify when the Guests can book time with them. This includes the date or dates as well as start and end times.

## Non-functional Requirements

- The System has two possible users: A Creator and a Guest

- The Creator should see three options once logged in: View Events, Settings, and Log Out.

- The guest should see an option to book available times without logging in.

- The main page should have a description of the applications function and must look aesthetically pleasing (use images).

## Use cases

### UML Use Case Diagram

Use Case Diagram

# Use Case Descriptions

## USE CASE 1

**Use Case Name: Log in**

**Summary:** A user on the main page of our site can log in with a username and password and has the option to have the system remember them on different devices.

### Actors

- Creator

### Preconditions:

- The user must be logged out

### Triggers:

- When accessing the website, a login prompt will pop up when the user tries to schedule an event.
- When the user clicks the "login" button.

### Primary Sequence

1. User prompted to log in
2. The User enters in a username
3. The User enters in password
4. System checks if username and password are valid
5. System logs user in

### Primary Postconditions

- The user is logged in.
- The user's account is valid.

### Alternate Sequences

- If the username or password is invalid the user will be prompted again to enter a valid username or password or create an account.
- If the user clicks "remember me" the system will remember their login information on this device.

### Alternate Triggers

- User enters an incorrect password for their account
- User selects the "remember me" box under the login form

### Alternate Postconditions

- User's login credentials are stored so they don't have to resubmit login form.

### Glossary

- User = the actor
- Guest = any user that does not have an account or has not logged in.
- Creator = The user that publishes their availability times

# USE CASE 2

Use Case Name: Delete Account

Summary: A creator on the main page can click on settings and click "delete account" to purge any information from the database related to the account.

### Actors

- Creator

### Preconditions:

- The user must be logged in an account

### Triggers:

- When the creator clicks on delete account from the settings page

### Primary Sequence

1. User clicks on Settings
2. User clicks on 'Delete Account'
3. System prompts confirmation of account deletion
4. Creator confirms account deletion
5. All events created by creator will be deleted

6. Username and password information will be deleted

## Primary Postconditions

- User will be seen as a guest
- User will be back at the main page

## Alternate Sequences

User clicks on Settings User clicks on 'Delete Account' System prompts confirmation of account deletion Creator declines account deletion Creator is back on settings page

## Alternate Triggers

- When the creator declines the system prompt to delete account

## Alternate Postconditions

- User will stay on Creator account
- User will be back at the main page

## Glossary

- User = the actor
- Guest = any user that does not have an account or has not logged in.
- Creator = The user that publishes their availability times

# USE CASE 3

Use Case Name: Book time slot

Summary: A guest will be able to book available times

## Actors

- Guest

## Preconditions:

- Is there an available time
- Is the date and time input correct?

## Triggers:

- User clicks on 'Book time slot'

## Primary Sequence

1. User clicks on 'Book time slot'
2. System displays list of available dates
3. User selects date
4. System displays list of available times
5. User selects time
6. System prompts name of booking/reason for booking
7. User enters name of booking
8. System prompts name of guest
9. User enters name
10. User clicks on 'Book'

## Primary Postconditions

- Calendar will show the new booking
- User will go back to main page

## Alternate Postconditions

- User stays on Book time slot page

## Glossary

- User = the actor
- Guest = any user that does not have an account or has not logged in.
- Creator = The user that publishes their availability times

# USE CASE 4

Use Case Name: Create Account

Summary: A guest on the website can create an account that publishes availability times.

## Actors

- Creator

## Preconditions:

- User must be logged out

## Triggers:

- The user clicks create account on the main page

## Primary Sequence

1. User clicks Create New Account

2. System prompts new username, password, and email
3. System validates username, password, and email
4. System creates new account

## Alternative Sequences

- If the username is already taken the system will report an error and ask for a different username.

## Primary Postconditions

- Users are able to log in with their newly created account.

## Glossary

- User = the actor
- Guest = any user that does not have an account or has not logged in.
- Creator = The user that publishes their availability times

# USE CASE 5

Use Case Name: Edit Upcoming Event

Summary: A creator can edit events/bookings made by guests.

## Actors

- Creator

## Preconditions:

- The user is logged in
- Events exist to be edited

## Triggers:

- When viewing events, there will be an edit button that allows a user to make changes to their events.

## Primary Sequence

1. User clicks on view events.
2. User locates the desired event to change.
3. User clicks on the edit button under the event.
4. The user makes changes to the time, location, or name of the event.
5. User clicks confirm.
6. System updates the event.

## Primary Postconditions

- The user is able to view the event with new changes.

## Glossary

- User = the actor
- Guest = any user that does not have an account or has not logged in.
- Creator = The user that publishes their availability times