



# **SC2006 Software Engineering Project**

## **Product Description**

### Purpose

---

\_\_\_\_ app/website name \_\_\_\_ is a website designed to create a seamless experience for basketball enthusiasts, enabling them to find and participate in games that fit their preferences while also fostering a community of like-minded sports enthusiasts. The platform is designed to be user-friendly, making it easy for users to connect, organise, and enjoy basketball games together.

Effectively, \_\_\_\_ app/website name \_\_ serves two main purposes. They are as follows:

1) Connecting Basketball Players by Location and Time:

The primary purpose is to allow users to find or create basketball events based on their preferred location and timing. By using our website, users can quickly discover nearby basketball games that fit their schedule, whether they want to join an existing game or start one of their own.

2) Providing Weather Forecasts for Better Planning:

The second purpose of \_\_\_\_\_ is to help users plan their basketball sessions more effectively by providing regional weather forecasts for their chosen time range. This feature ensures that users can avoid disruptions due to weather and make the most of their basketball experience.

---

## Scope

---

The scope of the app includes the following key functionalities:

1) Event Search:

- Users can search for basketball events in their area by selecting a location, date and a specific time range.
- The website will then display a list of basketball events created by other users which match the user's criteria, allowing them to view details such as the location of event, date, start and end times, and the number of participants required.
- Users can access further details about each event by clicking on it and it will show the picture of the location, additional details from the event organiser.
- Users can then click to join the event if it is suitable.

2) Event Creation:

- Users can create their own basketball events by specifying details such as the location, date, start time, end time, maximum number of participants needed , and any additional information.
- This feature allows users to organise basketball games that fit their schedule and location preferences, inviting others in the community to join.

3) Additional Features:

- The app provides regional weather forecasts for the selected time and location of the event, helping users plan their games effectively and avoid disruptions due to adverse weather conditions
-

## **Users and Stakeholders**

---

## **Assumptions and Constraints**

---

---

## **Constraints**

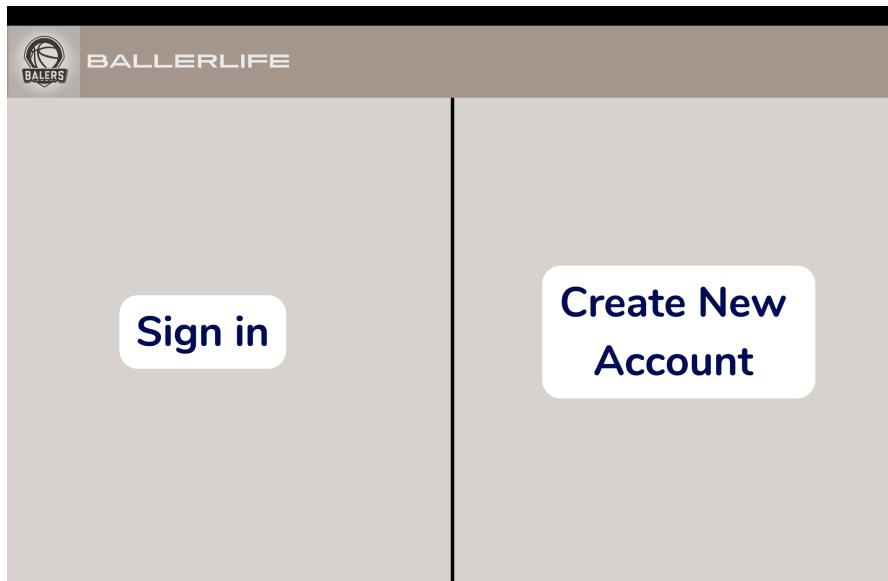
---

---

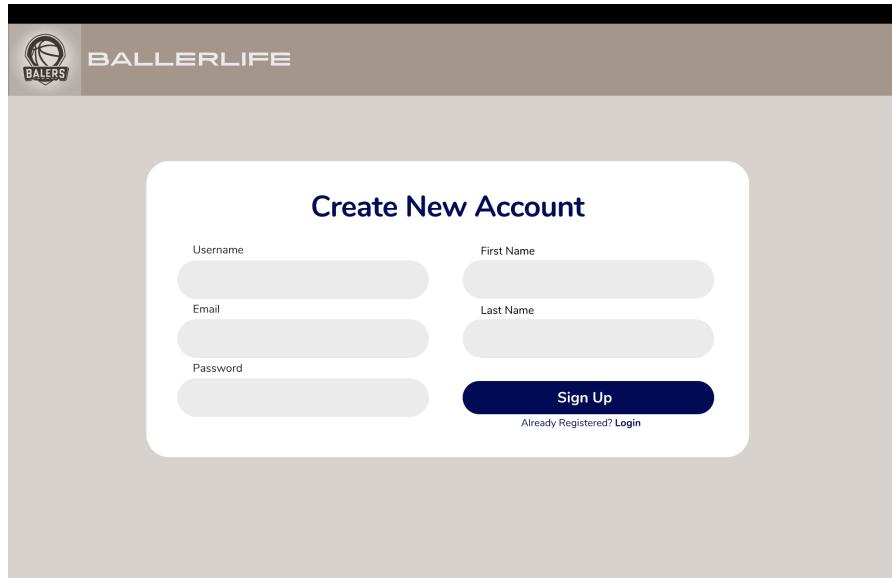
## Initial UI Mock-Up

---

**1. Initial Page (Login/SignUp):** Upon visiting the website, users must sign in to access any of the available features. If they do not have an existing account, they will need to create a new one.

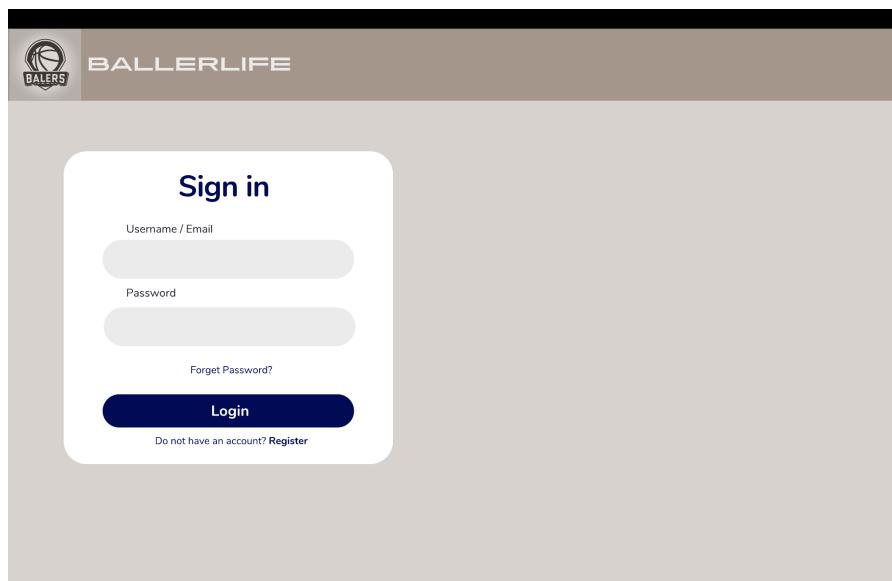


**2. Create New Account Page:** When the user clicks "Create Account," they will be directed to this page where they can enter all necessary credentials, including their username, email, name, and the password they wish to use for the website.



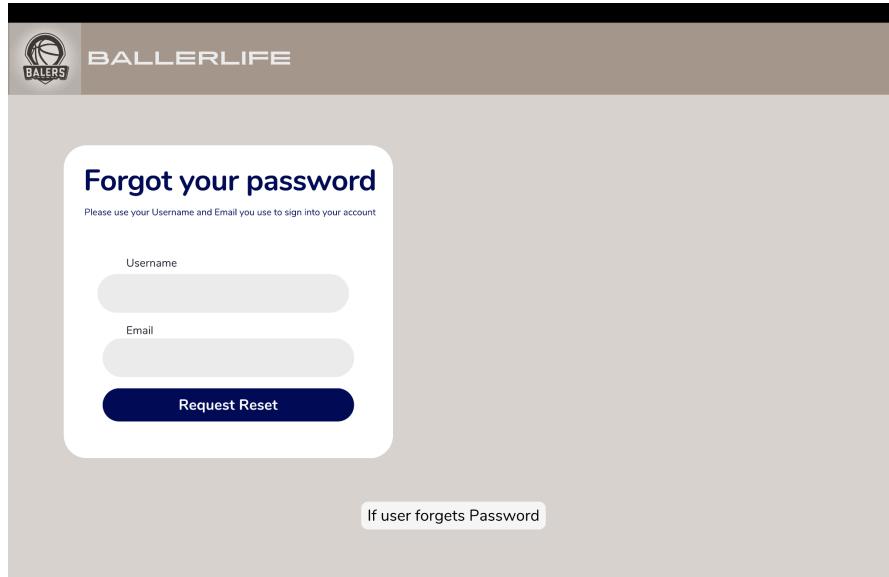
The screenshot shows the 'Create New Account' form. At the top left is the BALERS logo, followed by the word 'BALLERLIFE'. The main title 'Create New Account' is centered at the top of the form. Below the title are four input fields: 'Username' (left), 'First Name' (right), 'Email' (left), and 'Last Name' (right). Each input field has a placeholder text above it. Below these fields is a large blue 'Sign Up' button. Underneath the button, there is a small link 'Already Registered? Login'.

**3. Login Page:** After the user has finished creating a new account or if they already have an existing account, they will be directed to the login page.



The screenshot shows the 'Sign in' form. At the top left is the BALERS logo, followed by the word 'BALLERLIFE'. The main title 'Sign in' is centered at the top of the form. Below the title are two input fields: 'Username / Email' (top) and 'Password' (bottom). Each input field has a placeholder text above it. Between the two input fields is a small link 'Forgot Password?'. Below the password field is a large blue 'Login' button. Underneath the button, there is a small link 'Do not have an account? Register'.

**4. Forget Password:** If existing users forget their passwords, they can click the "Forgot Password" shown on the previous page. They will then be redirected to a page where they can follow the instructions to reset their passwords.



**5. Display Events:** The main page where all events are listed in ascending order by date and time.

The screenshot shows the BALLERLIFE app's event listing screen. At the top, there is a search bar with fields for Location (e.g. Toa Payoh Lorong 7), Date (DD/MM/YY), and Time (Start Time). There are also 'My Activities' and 'My Profile' buttons. A red button at the top center says 'Sorted By: Date/Start Time (Ascending)'. Below this is a section titled 'Events' containing three event cards:

- Siyuan's Basketball Party** (NTU SRC) - Start: 04/09/2024 at 4:00 PM, End: 7:30 PM. Pax: 2/8. Buttons: 'More Information' and 'Join Now'.
- Michael's Casual Invite** (Woodlands Ave 13 Blk 45) - Start: 05/09/2024 at 4:00 PM, End: 7:30 PM. Pax: 2/5. Buttons: 'More Information' and 'Join Now'.
- Just balling** (Bishan Park) - Start: 05/09/2024 at 7:00 PM, End: 9:30 PM. Pax: 5/8. Buttons: 'More Information' and 'Join Now'.

At the bottom, there is a page navigation bar with 'Pages: 1 2 3 4 5 6 7 8 9 10 .... 20 <>'.

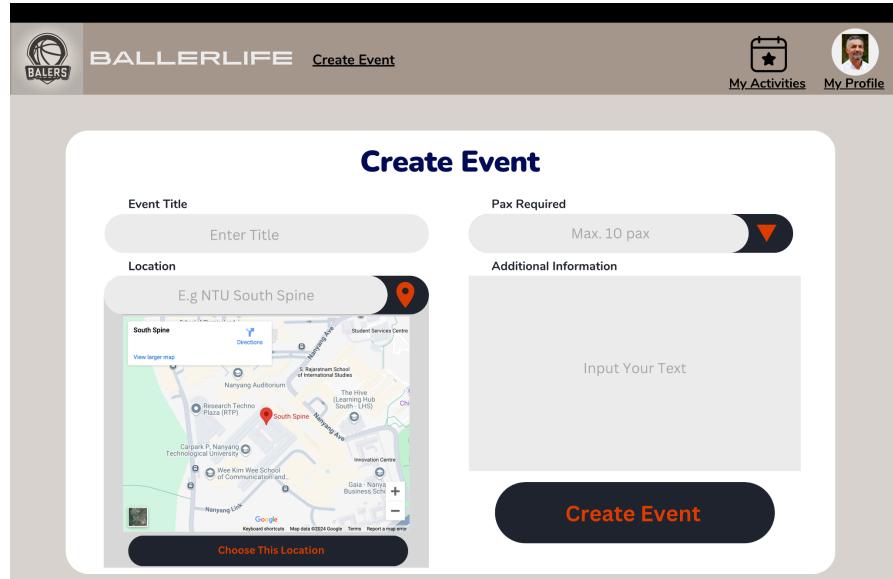
**6. Create Event:** The user can create their own event on this page by entering the necessary event details.

The screenshot shows the 'Create Event' form. It has several input fields:

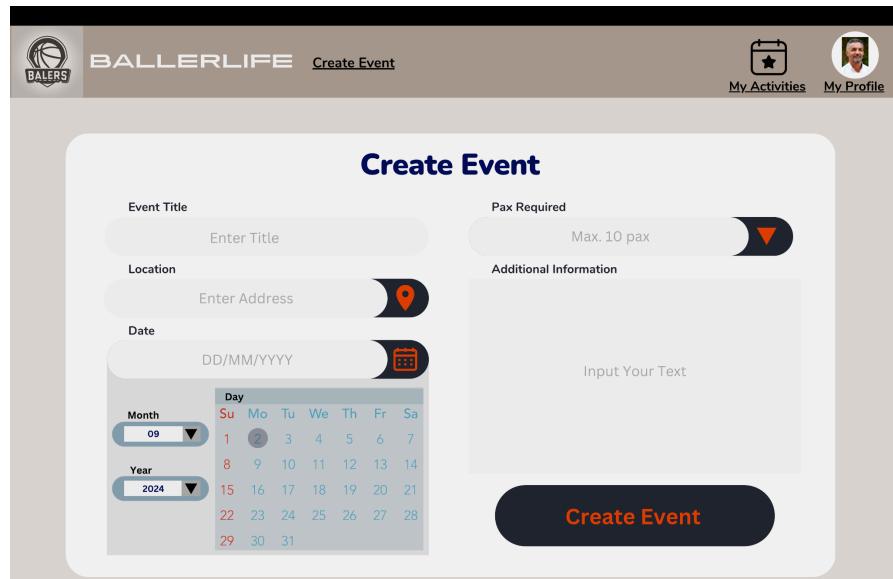
- Event Title:** A text input field with placeholder 'Enter Title'.
- Pax Required:** A dropdown menu set to 'Max. 10 pax'.
- Location:** A text input field with placeholder 'Enter Address'.
- Date:** A date picker input field with placeholder 'DD/MM/YY'.
- Start Time:** A time picker input field with placeholder 'Click Icon to Select Time'.
- End Time:** A time picker input field with placeholder 'Click Icon to Select Time'.
- Additional Information:** A large text area for additional details.

A large orange button at the bottom right says 'Create Event'.

**6.1. Input Location:** The user can either use their GPS location or manually enter the basketball court location.



**6.2. Input Date:** The user can input the date of event



**6.3. Input Start and End Time:** The user can input the Start and End time of the event

The screenshot shows the first step of creating an event. The title "Create Event" is at the top. The form includes fields for "Event Title" (placeholder "Enter Title"), "Pax Required" (set to "Max. 10 pax"), "Location" (placeholder "Enter Address"), "Date" (placeholder "DD/MM/YY"), "Start Time" (placeholder "Click Icon to Select Time" with dropdowns for Hour, Minute, and AM/PM set to 06, 00, and PM respectively), and an "Additional Information" text area ("Input Your Text"). A large orange "Create Event" button is at the bottom.

The screenshot shows the second step of creating an event, identical to the first one. It includes fields for "Event Title" (placeholder "Enter Title"), "Pax Required" (set to "Max. 10 pax"), "Location" (placeholder "Enter Address"), "Date" (placeholder "DD/MM/YY"), "Start Time" (placeholder "Click Icon to Select Time" with dropdowns for Hour, Minute, and AM/PM set to 09, 54, and AM respectively), and an "Additional Information" text area ("Input Your Text"). A large orange "Create Event" button is at the bottom.

**7. Filter Event:** The user can apply filters to find events that match their preferences, such as location, date, and start time.

The screenshot shows the filter interface. It features three input fields: "Location" (placeholder "E.g. Toa Payoh Lorong 7" with a location pin icon), "Date" (placeholder "DD/MM/YY" with a calendar icon), and "Time" (placeholder "Start Time" with a clock icon). Below these is an orange "Filter Event" button.

**7.1. Input Location:** The user can either use their GPS location or manually enter a location of their choice.

The screenshot shows the BALLERLIFE app interface. At the top, there is a search bar with placeholder text "E.g. Toa Payoh Lorong 7" and a location pin icon. To the right of the search bar are buttons for "Date DD/MM/YY", "Time Start Time", and "Filter Event". In the top right corner, there are links for "My Activities" and "My Profile". Below the search bar is a map showing the area around Toa Payoh Lorong 7, with a red marker indicating the location. A button labeled "Choose This Location" is present. The main content area is titled "Events" and displays three event cards. Each card includes a thumbnail, the event name, start time, end time, and participant count (Pax). Buttons for "More Information" and "Join Now" are also shown. The first event is "Just balling" at Bishan Park, starting at 7:00 PM and ending at 9:30 PM, Pax: 5/8. The second event is "Siyuan's Bas" at NTU SRC, starting at 4:00 PM on 04/09/2024, Pax: 2/8. The third event is "Michael's Casual Invite" at Woodlands Ave 13 Blk 45, starting at 4:00 PM on 05/09/2024, Pax: 2/5. At the bottom, there is a pagination control with buttons for pages 1 through 20 and a double arrow icon.

**7.2. Input Date:** The user can input their preferred date

This screenshot is similar to the previous one but includes a date picker overlay. The overlay has fields for "Month" (set to 09) and "Year" (set to 2024), with a dropdown arrow icon. The calendar grid shows days from Sunday to Saturday, with the current day highlighted in blue. The main content area below the overlay shows the same three events as the previous screenshot, but the first event's details are partially obscured by the date picker. The events are: "Siyuan's Bas" at NTU SRC, "Michael's Casual Invite" at Woodlands Ave 13 Blk 45, and "Just balling" at Bishan Park. The bottom of the screen features a pagination control with pages 1 through 20 and a double arrow icon.

### 7.3. Input Start Time: The user can input preferred start time

**BALLERLIFE** Create Event

Location: E.g. Toa Payoh Lorong 7 Date: DD/MM/YY Time: Start Time: 06:00 PM Filter Event

Sorted By: Date

Event	Start	End	Pax	Action
Siyuan's Basketball Party NTU SRC	4:00 PM	7:30 PM	2/8	<a href="#">More Information</a> <a href="#">Join Now</a>
Michael's Casual Invite Woodlands Ave 13 Blk 45	4:00 PM	7:30 PM	2/5	<a href="#">More Information</a> <a href="#">Join Now</a>
Just balling Bishan Park	7:00 PM	9:30 PM	5/8	<a href="#">More Information</a> <a href="#">Join Now</a>

Pages: 1 2 3 4 5 6 7 8 9 10 .... 20 <>

### 7.4. Filtered Outcome: A list of events matching the user's criteria will be displayed, including all events occurring on the selected date, within a 5km radius of the specified location, and with a time range that overlaps the preferred start time.

**BALLERLIFE** Create Event

Location: Toa Payoh Lorong 7 Date: 02/09/2024 Time: 6:00 PM Filter Event

Your Filters Returned 10 Available Events  
Sorted By: Ascending

**Events**

Event	Start	End	Pax	Action
HQ's Basketball Party Toa Payoh Lorong 7 Blk 9	6:30 PM	7:30 PM	7/10	<a href="#">More Information</a> <a href="#">Join Now</a>
Urgent!! Require Players!!! Toa Payoh South CC	6:30 PM	10:30 PM	2/8	<a href="#">More Information</a> <a href="#">Join Now</a>
Casual Player Only Toa Payoh Lorong 8 Blk 222	7:00 PM	9:30 PM	4/8	<a href="#">More Information</a> <a href="#">Join Now</a>

Pages: 1 2 3 4 <>

**8. Event Information Page:** A page displaying comprehensive information about the selected event, including additional information, a map highlighting the event location, and a link to Google Maps for directions to the venue.

The screenshot shows the BallerLife platform interface for an event titled "HQ's Basketball Party".

**Header:** Features the BallerLife logo, a "Create Event" button, and links for "My Activities" and "My Profile".

**Event Title:** "HQ's Basketball Party"

**Event Information:** Includes details such as Location (Toa Payoh Lorong 7 Blk 9), Start Time (6:30 PM), End Time (7:30 PM), Date (14/09/2024), and Current Pax/Required Pax (7/10).

**Additional Information:** A list of instructions for attendees:

- Please bring your own water bottle (Drinks are not provided)
- Wear proper basketball shoes
- In case you cannot find your way, please contact via phone 98765432 or tele: iamMING1234

**Map:** An embedded map showing the location of Block 9, Toa Payoh, Singapore 310009, with nearby landmarks like Wei Chen Public School and Kim Kee Kopitiam.

**Directions:** A "Directions" button and a thumbnail image of the basketball court at the event location.

**Weather Forecast:** (Empty box)

**Ultra Violet Index:** (Empty box)

**Footer:** Includes "Event Created By:" (with a profile picture), "View Profile", "Like Event" (with a heart icon), and "JOIN NOW" buttons.

**9. My Activities Page:** Displays all events in which the user is involved.

You Have Joined 3 events  
Sorted By: Ascending

**My Activities**

User Profile	Event Details	Action
	<b>Casual Players !!!</b> Start: 02/09/2024 at 5:30 PM   End: 02/09/2024 at 7:30 PM 619 Jurong West Street 65 Pax: 8/10	<b>More Information</b>
	<b>Come Play And Chill</b> Start: 03/09/2024 at 8:30 PM   End: 03/09/2024 at 10:30 PM 810 Bukit Batok West Ave 5 Pax: 8/8	<b>More Information</b>
	<b>Just For Fun</b> Start: 05/09/2024 at 6:00 PM   End: 05/09/2024 at 8:00 PM 428 Choa Chu Kang Ave 4 Pax: 8/8	<b>More Information</b>

Pages: 1 2 3 4

**10. My Profile Page:** Displays the user's personal information, including their username, name, and email.

**My Profile**

	<input type="button" value="Camera icon"/>
Username	Johnlee123
First Name	John
Last Name	Lee
Email	Johnlee@gmail.com

## **Functional Requirements**

### **1. Main Menu**

- 1.1. The system must allow users to create an account.
- 1.2. The system must allow users to login to their account.
- 1.3. Each user must have a profile page
- 1.4. The system must allow users to use the Create Event function.
- 1.5. The system must allow users to use the Search Event function.
- 1.6. The user must be able to select and retrieve more details of a particular event including:
  - 1.6.1. Location
  - 1.6.2. Date
  - 1.6.3. Start Time
  - 1.6.4. End Time
  - 1.6.5. Pax required
  - 1.6.6. Picture of the location
  - 1.6.7. Location on embedded map
  - 1.6.8. Additional information from the event creator

### **2. Registration**

- 2.1. The user must be able to register for a new account with information which includes:
  - 2.2.1.1. First + Last Name
  - 2.2.1.2. Username

- 2.2.1.3. Email Address
- 2.2.1.4. Password
- 2.2. The system must validate user inputs with the following criterias:
  - 2.2.2.1. All input fields are filled up
  - 2.2.2.2. Username does not exist in database
  - 2.2.2.3. Email does not exist in database
  - 2.2.2.4. Password meets the security requirements in Section 3.1.1
- 2.3. If any of the above input validation fails, the system must display an error message for each validation failure.

### 3. Login

- 3.1. The user must input their username and password.
- 3.2. The system must validate inputs with the following criterias:
  - 3.2.1. All input fields are filled up.
  - 3.2.2. The username and password matches the database record.
- 3.3. If validation fails, the system must display an error message for each validation failure listed in Section 2.3.2.
- 3.4. Upon successful login, the system must redirect the user to the application's main menu.
- 3.5. The user must be able to request a change of password should he forget his password.
- 3.6. The system must send email verification code upon the user's request for change of password

### 4. Profile

- 4.1. Each user must have a profile page that displays the following:
  - 2.4.1.1. Profile Picture

- 2.4.1.2. Name
  - 2.4.1.3. Username
  - 2.4.1.4. Short description about the user.
  - 2.4.1.5. Events that the user has signed up for
- 4.2. Each user must be able to edit the following components of their profile page:
- 4.2.1. Profile Picture
  - 4.2.2. Short Description
- 4.3. Each user must be able to view other users' profile pages.

## 5. Event Creation

- 5.1. The system must prompt the event creator to enter the following information:
- 5.1.1. Title
  - 5.1.2. Current Location or User-defined Location
  - 5.1.3. Date
    - 5.1.3.1. Limited to range from current date to two weeks ahead
  - 5.1.4. Start Time
    - 5.1.4.1. Minimally >2 hours from current time
  - 5.1.5. End Time
  - 5.1.6. Maximum number of participants
  - 5.1.7. Additional text information and instructions (Optional)
- 5.2. The system must display a list or a map with pin-pointed locations of the basketball courts within a 5km radius of the chosen location
- 5.2.1. The user must be able to select a basketball court as the event location
- 5.3. The event creator must be an event participant

## 6. Event View

- 6.1. The event view displays the following information about the event:
  - 6.1.1. Title
  - 6.1.2. Basketball Court Location
  - 6.1.3. Date
  - 6.1.4. Start Time
  - 6.1.5. End Time
  - 6.1.6. Current Number of Participants
  - 6.1.7. Maximum Number of Participants
  - 6.1.8. Additional text information and instructions (if provided by creator)
  - 6.1.9. Link to Profile Page of Event Creator
- 6.1.10. Weather information from NEA Weather API:
  - 6.1.10.1. Weather Forecast Data
  - 6.1.10.2. Ultraviolet Index
- 6.2. The user must be able to join an event if the event is not full
- 6.3. The user must be able to withdraw participation in an event
- 6.4. The system should update the Current Number of Participants when a user joins or withdraws from an event

## 7. Event List

- 7.1. The event list is a page that displays all events the user has joined
- 7.2. The user must be able to select an event to view the event information in the event view

## 8. Event Search

- 8.1. The system must display events sorted by newest to oldest by default

- 8.2. The system must allow the user to enter the following filter inputs:
  - 8.2.1. Current Location or User-defined Location
  - 8.2.2. Date of event
  - 8.2.3. Time
- 8.3. The system must filter and return a list of events created by other users based on the following criteria:
  - 8.3.1. Location (Within a 5km radius of the chosen location)
  - 8.3.2. Date
  - 8.3.3. Time (As long as the start time that user selected is within this range)
  - 8.3.4. Participation count (If event is full, then it will not appear in the list)
- 8.4. The user must be able to select an event to view the event information in the event view

## 9. Interface with other systems

- 9.1. The following APIs are used:
  - 9.1.1. NEA Weather API from Data.gov.sg for weather forecast and ultraviolet index data
  - 9.1.2. Google Maps API for map data
  - 9.1.3. Google Geolocation API to retrieve user's current location

## 10. Format for information to be processed

- 10.1. Location
  - 10.1.1. Distance should be stored with "m" as the base unit

- 10.1.2. Distance will be displayed in “km” and two decimal places of accuracy (e.g. 2.05km) if the distance is more than or equal to 1000m.
  - 10.1.3. Distance will be displayed in “m” if the distance is less than 1000m
- 10.2. Date and Time
- 10.2.1. Date and Time must be jointly stored as a javascript Date object according to the ISO 8601 Datetime format

- Could add in a special admin account that has authorization to edit and delete any events, delete users
- Limit each user to only have 5 events created by them at any time to prevent spam
- List of user-registered events

## **Non-Functional Requirements**

1. Security
  - 1.1. User account passwords must satisfy the following criteria:
    - 1.1.1. Contains 12 characters or more
    - 1.1.2. Contains at least one uppercase alphabet character, one lowercase alphabet character, one special character and one numeric character
  - 1.2. Passwords should be hashed with the Bcrypt hash function before being stored in the database

- 1.3. Authentication should be done on both server-side and client-side
  - 1.4. Log-in inputs should be sanitised to prevent Cross-Site Scripting Attacks
2. Maintainability
  - 2.1. An MVC design pattern should be implemented for clear division of responsibility, keeping code organised and easier to maintain
  - 2.2. PostgreSQL (or any other decent SQL based database) will be used to ensure easy data migration to other SQL based databases in the future if needed
3. Usability
  - 3.1. Complies with WCAG 2 A Accessibility standard
  - 3.2. Responsive Web Design should be implemented such that the website scales to these classes of devices:
    - 3.2.1. Computer screens and monitors
    - 3.2.2. Tablets
    - 3.2.3. Mobile Phones
  - 3.3. All website functions should work on these browsers:
    - 3.3.1. Chrome
    - 3.3.2. Edge
    - 3.3.3. Safari
    - 3.3.4. Firefox

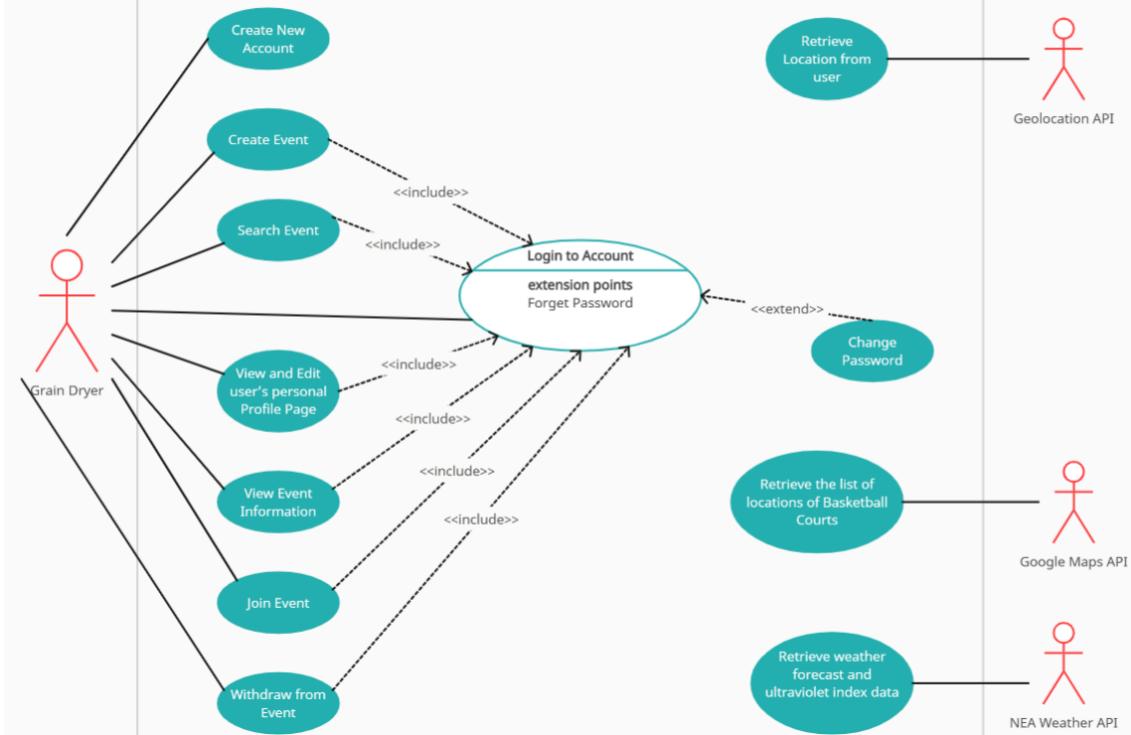
## Data Dictionary

Term	Definition
API	Short for Application Programming Interface. Allow applications to exchange data based on a common protocol. Commonly provided by companies like Google and OpenAI to allow developers to access their data
Main Menu	The default page of the website
Ultraviolet Index	An international standard measurement of the strength of the sunburn-producing ultraviolet (UV) radiation at a particular place and time.
Radius	The distance from the centre of a circle to its edge
Responsive Web Design	A property of websites that allow them to scale properly to various screen sizes and when zoomed in and out
WCAG	Short for Web Content Accessibility Guidelines, a web accessibility standard provided by the World Wide Web Consortium to ensure websites are accessible for people with various disabilities as well as usable for all people in general
Participation Count (Pax)	The total number of people who have joined an event

User	
Account	

# Use Case Model

## 3.1 Use Case Diagram



## 3.2 Use Case Description

---

Use Case ID:	1		
User Case Name:	<b>Create Event</b>		
Created By:	Bryan	Last Updated By:	Siyuan
Date Created:	2/9/2024	Date Last Updated:	4/9/2024
Actor:	User		
Description:	User creates a new event with specific details, such as title, location, date, time, and maximum participants		
Pre-Conditions:	<ul style="list-style-type: none"><li>User is authenticated</li></ul>		
Post-Conditions:	<ul style="list-style-type: none"><li>New event is created</li><li>New event can be seen by other users on main page</li></ul>		
Priority:	High		
Frequency of Use:	Frequently		
Flow of Events:	<ol style="list-style-type: none"><li>User chooses to create new event</li><li>System prompts user to enter event details<ol style="list-style-type: none"><li>Title</li><li>Location</li><li>Date</li><li>Start and End Time</li><li>Preferred number of participants</li><li>Short description (optional)</li></ol></li><li>System displays list of basketball courts near the user</li><li>User reviews whether information is correct and confirms if it is, creating a new event</li></ol>		

Alternative Flows:	
Exception:	
Includes:	<b>Login to Account</b>
Special Requirements:	<ul style="list-style-type: none"> <li>• Real time location tracking</li> <li>• List of basketball courts in Singapore</li> <li>• Location can be live or user-inputted</li> <li>• Date restricted from current to 2 weeks ahead</li> <li>• Start time must be at least 2 hours from current time</li> </ul>
Assumptions:	<ul style="list-style-type: none"> <li>• All basketball courts can be found on Maps API</li> </ul>
Notes and Issues:	<ul style="list-style-type: none"> <li>• Must be responsive enough for the user, i.e. creating an event should take a minute at most.</li> <li>• Can be scaled to allow users to recreate old events they made</li> <li>• Can be made so user cannot create event they created already</li> </ul>

Use Case ID:	2		
User Case Name:	<b>Create New Account</b>		
Created By:	Bryan	Last Updated By:	Michael
Date Created:	2/9/2024	Date Last Updated:	4/9/2024
Actor:	User		
Description:	Registration System for User		
Pre-Conditions:	<ul style="list-style-type: none"> <li>• User does not have existing account</li> </ul>		
Post-Conditions:	<ul style="list-style-type: none"> <li>• New user account is created</li> </ul>		
Priority:	High		

Frequency of Use:	Low, one time-use per user
Flow of Events:	<ol style="list-style-type: none"> <li>1. User clicks on register new account on websites registration page</li> <li>2. System prompts user to input the following fields             <ol style="list-style-type: none"> <li>a. First Name</li> <li>b. Last Name</li> <li>c. Username</li> <li>d. Password</li> <li>e. E-mail Address</li> </ol> </li> <li>3. System validates the inputs             <ol style="list-style-type: none"> <li>a. Ensure username does not exist in database</li> <li>b. Ensure email does not exist in database</li> <li>c. All fields are filled</li> <li>d. Password satisfies security criteria</li> </ol> </li> <li>4. If validations passed, new account is created</li> </ol>
Alternative Flows:	<ol style="list-style-type: none"> <li>1. Input validation failed due to not satisfying any of the criterions</li> <li>2. System prompts user to fix wrong field             <ol style="list-style-type: none"> <li>a. If password does not meet requirements, prompt user to change</li> <li>b. If username exists, prompt user to change</li> <li>c. If email exist, prompt user to change</li> <li>d. If field empty, prompts user to fill up</li> </ol> </li> </ol>
Exception:	<b>Change Password</b>
Includes:	
Special Requirements:	<ul style="list-style-type: none"> <li>● Database containing list of current users information exists</li> <li>● Password meets following requirements             <ol style="list-style-type: none"> <li>a. Contains 12 characters or more</li> <li>b. Contains at least one uppercase alphabet character</li> <li>c. Contains at least one lowercase alphabet character</li> <li>d. Contains at least one special character</li> <li>e. Contains at least one numeric character</li> </ol> </li> </ul>
Assumptions:	<ul style="list-style-type: none"> <li>● Users must have a valid email address</li> </ul>

Notes and Issues:	<ul style="list-style-type: none"> <li>Sanitised inputs to prevent database injections</li> <li>System prompts to users following failed inputs are clear</li> </ul>
-------------------	--

Use Case ID:	3		
User Case Name:	<b>Login to Account</b>		
Created By:	Bryan	Last Updated By:	Siyuan
Date Created:	2/9/2024	Date Last Updated:	4/9/2024
Actor:	User		
Description:	Login to System		
Pre-Conditions:	<ul style="list-style-type: none"> <li>User is currently logged out</li> </ul>		
Post-Conditions:	<ul style="list-style-type: none"> <li>User successfully logged in and is redirected to the main menu</li> </ul>		
Priority:	High		
Frequency of Use:	Frequent		
Flow of Events:	<ol style="list-style-type: none"> <li>User navigates to login page and clicks button to login</li> <li>System prompts user to fill in fields             <ol style="list-style-type: none"> <li>Username/email</li> <li>Password</li> </ol> </li> <li>Input is authenticated             <ol style="list-style-type: none"> <li>Check if Username/Email exists in database</li> <li>Check if password matches Username/Email</li> <li>Check all fields filled</li> </ol> </li> <li>If all authenticated, user is logged in and redirected to main menu</li> </ol>		
Alternative Flows:	<ul style="list-style-type: none"> <li>Authentication failed             <ol style="list-style-type: none"> <li>System prompts user to fill in wrong/required fields</li> </ol> </li> </ul>		

	<ul style="list-style-type: none"> <li>b. User can choose to change inputs and try to login again</li> <li>● Password Forgotten           <ul style="list-style-type: none"> <li>a. User can click on a button to reset password</li> <li>b. Leads to <b>Change Password</b> use case</li> </ul> </li> </ul>
Exception:	<b>Change Password</b>
Includes:	
Special Requirements:	<ul style="list-style-type: none"> <li>● Passwords in database are hashed</li> <li>● Login limit to prevent brute force</li> </ul>
Assumptions:	<ul style="list-style-type: none"> <li>● User has access to either username or email still</li> </ul>
Notes and Issues:	<ul style="list-style-type: none"> <li>● Can use session cookies or JWT to keep user signed in to reduce frequency of this use case</li> </ul>

Use Case ID:	4		
User Case Name:	<b>View and Edit User's Personal Profile Page</b>		
Created By:	Bryan	Last Updated By:	Siyuan
Date Created:	2/9/2024	Date Last Updated:	4/9/2024
Actor:	User		
Description:	Profile Management		
Pre-Conditions:	<ul style="list-style-type: none"> <li>● User is authenticated</li> </ul>		
Post-Conditions:	<ul style="list-style-type: none"> <li>● User profile is changed to adapt to their new inputs</li> </ul>		
Priority:	Low		
Frequency of Use:	Low		

Flow of Events:	<ol style="list-style-type: none"> <li>1. User can view following information in their profile page             <ol style="list-style-type: none"> <li>a. Profile picture</li> <li>b. Name</li> <li>c. Username</li> <li>d. Email</li> </ol> </li> <li>2. User has the option to change the following             <ol style="list-style-type: none"> <li>a. Profile picture</li> </ol> </li> <li>3. System prompts user on what to edit in profile             <ol style="list-style-type: none"> <li>a. Prompt user to upload new photo</li> </ol> </li> <li>4. New profile reflects changes made by user</li> </ol>
Alternative Flows:	
Exception:	
Includes:	<b>Login to Account</b>
Special Requirements:	Photo must be less than 50kb in size
Assumptions:	
Notes and Issues:	

Use Case ID:	5		
User Case Name:	<b>Change Password</b>		
Created By:	Bryan	Last Updated By:	Siyuan
Date Created:	2/9/2024	Date Last Updated:	4/9/2024
Actor:	User		
Description:	When user forgot his password, he needs to change/update the password of the account		

Pre-Conditions:	
Post-Conditions:	<ul style="list-style-type: none"> <li>User password has changed to new password as indicated</li> </ul>
Priority:	Medium
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> <li>User can click on forgot password on login page</li> <li>Email is sent to their email as indicated in the database to validate whether they actually want to change password</li> <li>If email is validated, user is able to change password as liked</li> <li>Password is changed to the new password</li> </ol>
Alternative Flows:	<ol style="list-style-type: none"> <li>If new password is invalid, system prompts user to enter a valid password</li> <li>If account does not exist, alerts user</li> </ol>
Exception:	
Includes:	
Special Requirements:	<ul style="list-style-type: none"> <li>New password cannot be same as old password</li> <li>Passwords must follow same requirements as when creating account</li> </ul>
Assumptions:	<ol style="list-style-type: none"> <li>User still has access to their email</li> </ol>
Notes and Issues:	<ul style="list-style-type: none"> <li>Need send email using external API</li> </ul>

Use Case ID:	6		
User Case Name:	<b>View Event Information</b>		
Created By:	Bryan	Last Updated By:	Siyuan
Date Created:	2/9/2024	Date Last Updated:	4/9/2024

Actor:	User
Description:	Allow user to view event information
Pre-Conditions:	<ul style="list-style-type: none"> <li>• User is authenticated</li> </ul>
Post-Conditions:	<ul style="list-style-type: none"> <li>• List of events is displayed</li> </ul>
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. User clicks event</li> <li>2. Following information is displayed for each event             <ol style="list-style-type: none"> <li>a. Title</li> <li>b. Location</li> <li>c. Date</li> <li>d. Time</li> <li>e. Participants</li> <li>f. Text information (if exists)</li> <li>g. Profile picture of event creator</li> <li>h. Predicted weather (from weather API)</li> </ol> </li> </ol>
Alternative Flows:	<ol style="list-style-type: none"> <li>1. Weather API cannot retrieve weather</li> <li>2. Error showing unable to retrieve is shown in place of weather</li> </ol>
Exception:	
Includes:	<b>Login System</b>
Special Requirements:	<ul style="list-style-type: none"> <li>• Exclude events with full participation on the list</li> <li>• Access to weather API</li> </ul>
Assumptions:	<ul style="list-style-type: none"> <li>• Weather API is accurate and updates in real time</li> <li>• System can accommodate number of users</li> </ul>
Notes and Issues:	

Use Case ID:	7		
User Case Name:	<b>Filter Event</b>		
Created By:	Bryan	Last Updated By:	Siyuan
Date Created:	2/9/2024	Date Last Updated:	4/9/2024
Actor:	User		
Description:	Allows users to search for events		
Pre-Conditions:	<ul style="list-style-type: none"> <li>• User is logged in</li> </ul>		
Post-Conditions:	<ul style="list-style-type: none"> <li>• Filtered list of events is showed to the user based on their inputs</li> </ul>		
Priority:	High		
Frequency of Use:	High		
Flow of Events:	<ol style="list-style-type: none"> <li>1. User can click on a button to filter for specific events</li> <li>2. System prompts for <ul style="list-style-type: none"> <li>a. Location</li> <li>b. Date</li> <li>c. Time</li> </ul> </li> <li>3. Location filters based on inputs <ul style="list-style-type: none"> <li>a. Within a radius from selected location</li> <li>b. Within selected date</li> <li>c. Within selected timeframe</li> </ul> </li> <li>4. List of filtered events is displayed to the user</li> </ol>		
Alternative Flows:	<ul style="list-style-type: none"> <li>• User location cannot be found automatically <ul style="list-style-type: none"> <li>a. Prompts user to enter manual location</li> </ul> </li> <li>• No courts found <ul style="list-style-type: none"> <li>a. Informs user of lack of events and give suggestions on other events possible</li> </ul> </li> </ul>		
Exception:			

Includes:	<b>Login to Account</b>
Special Requirements:	<ul style="list-style-type: none"> <li>System contains a list of events sorted from newest to oldest automatically</li> <li>System needs to return filtered list quickly within 30 seconds</li> <li>Location can be live or user-inputted</li> <li>Prompted date has to be from current to 2 weeks after</li> <li>Within 5 km of selected location</li> </ul>
Assumptions:	1. Maps API contains all possible basketball courts
Notes and Issues:	

Use Case ID:	8		
User Case Name:	<b>Withdraw from event</b>		
Created By:	Bryan	Last Updated By:	Siyuan
Date Created:	2/9/2024	Date Last Updated:	4/9/2024
Actor:	User		
Description:	User can withdraw from events they joined		
Pre-Conditions:	<ul style="list-style-type: none"> <li>User has joined a pre-existing event</li> <li>User is logged in</li> </ul>		
Post-Conditions:	<ul style="list-style-type: none"> <li>User is withdrawn from the event</li> <li>Event details is updated to show change</li> </ul>		
Priority:	High		
Frequency of Use:	Medium		
Flow of Events:	<ol style="list-style-type: none"> <li>If enrolled in event, user can choose to withdraw by clicking on withdraw on the event they joined</li> </ol>		

	2. Event pax is updated to show withdrawal
Alternative Flows:	
Exception:	
Includes:	<b>Login to Account</b>
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	9		
User Case Name:	<b>Join Event</b>		
Created By:	Bryan	Last Updated By:	Siyuan
Date Created:	2/9/2024	Date Last Updated:	4/9/2024
Actor:	User		
Description:	User join existing event		
Pre-Conditions:	<ul style="list-style-type: none"> <li>● User is authenticated</li> <li>● Event is not full</li> </ul>		
Post-Conditions:	<ul style="list-style-type: none"> <li>● User joins event</li> <li>● Event description updates to reflect participation count</li> </ul>		
Priority:	High		
Frequency of Use:	High		
Flow of Events:	1. User navigates to the event view page		

	<ol style="list-style-type: none"> <li>2. User can click to join an event</li> <li>3. Event pax is updated to show participation</li> </ol>
Alternative Flows:	<ol style="list-style-type: none"> <li>1. If the user was the admin, admin is passed to next earliest user to join the event</li> </ol>
Exception:	
Includes:	<ul style="list-style-type: none"> <li>• Login System</li> </ul>
Special Requirements:	
Assumptions:	
Notes and Issues:	<ul style="list-style-type: none"> <li>• Can prevent users from joining events in same period</li> </ul>

Use Case ID:	10		
User Case Name:			
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	
Actor:	NEA Weather API		
Description:	Returns weather conditions of specific time and location		
Pre-Conditions:	<ul style="list-style-type: none"> <li>• Court location is given</li> <li>• Time is given</li> </ul>		
Post-Conditions:	<ul style="list-style-type: none"> <li>• Weather condition is returned</li> </ul>		
Priority:	Medium		
Frequency of Use:	High		

Flow of Events:	<ol style="list-style-type: none"> <li>1. User views list of available events</li> <li>2. Information about event such as location and time is returned by system to Weather API</li> <li>3. Weather API returns predicted weather during specified time and location</li> </ol>
Alternative Flows:	<ol style="list-style-type: none"> <li>1. Weather cannot be found even when information is given</li> <li>2. Error is returned</li> </ol>
Exception:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	11		
User Case Name:			
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	
Actor:	Geolocation API		
Description:	Retrieves the users location		
Pre-Conditions:	<ul style="list-style-type: none"> <li>Users location is accessible by API</li> </ul>		
Post-Conditions:	<ul style="list-style-type: none"> <li>Returns users location</li> </ul>		
Priority:	High		

Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. User selects to use live location when choosing location</li> <li>2. Geolocation API returns users location</li> </ol>
Alternative Flows:	<ol style="list-style-type: none"> <li>1. User location cannot be retrieved</li> <li>2. Error is returned</li> </ol>
Exception:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	12		
User Case Name:			
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	
Actor:	Google Maps API		
Description:	Returns list of basketball courts		
Pre-Conditions:	<ul style="list-style-type: none"> <li>Location is entered when creating event</li> </ul>		
Post-Conditions:	<ul style="list-style-type: none"> <li>List of basketball courts is returned</li> </ul>		
Priority:	High		
Frequency of Use:	High		

Flow of Events:	<ol style="list-style-type: none"> <li>1. Users live location or manual location is entered when creating new event</li> <li>2. List of basketball courts within 5km of selected location is returned by Maps API</li> </ol>
Alternative Flows:	
Exception:	
Includes:	
Special Requirements:	
Assumptions:	<ol style="list-style-type: none"> <li>1. Google maps API contains list of all basketball courts</li> </ol>
Notes and Issues:	

---