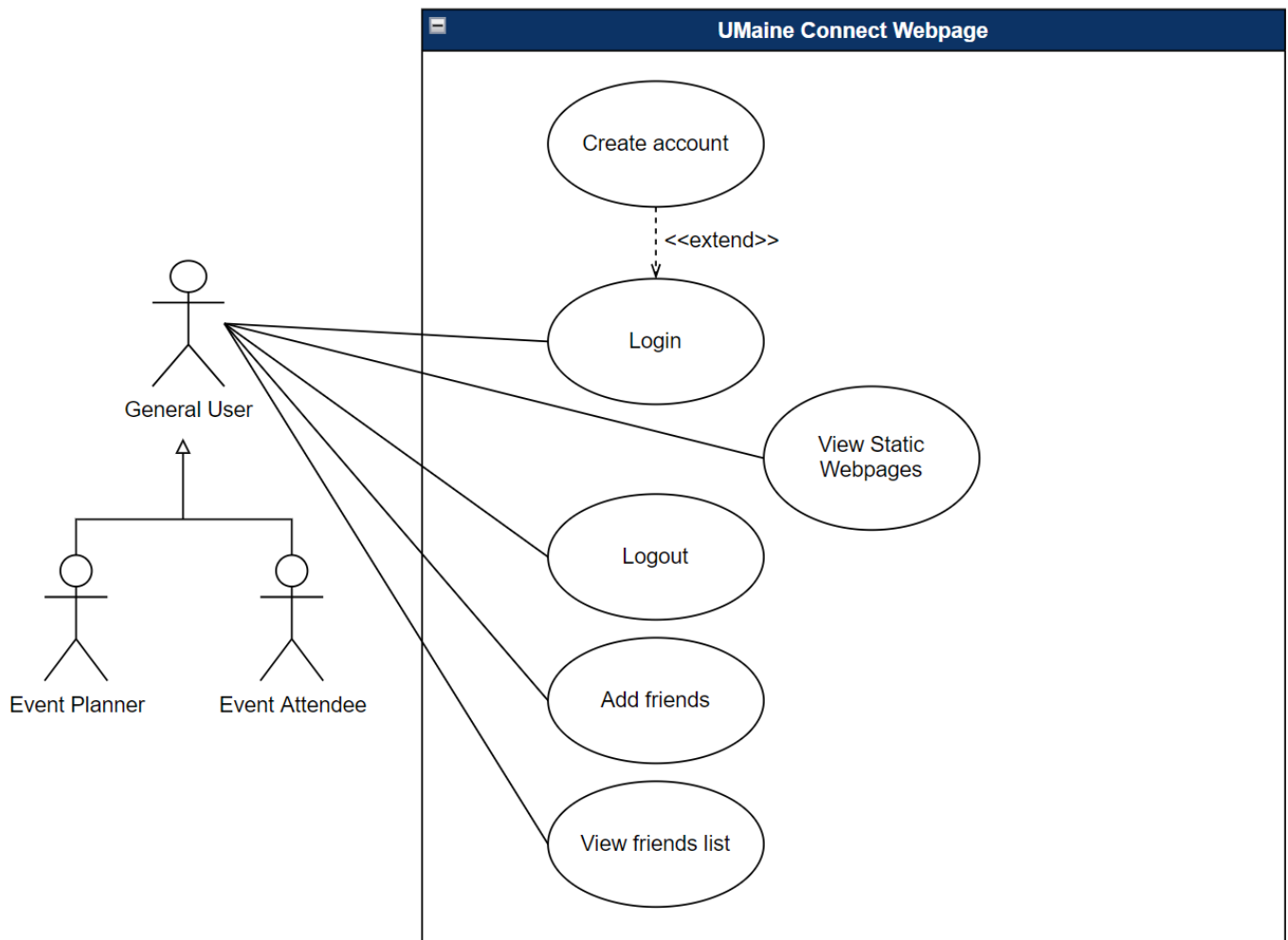
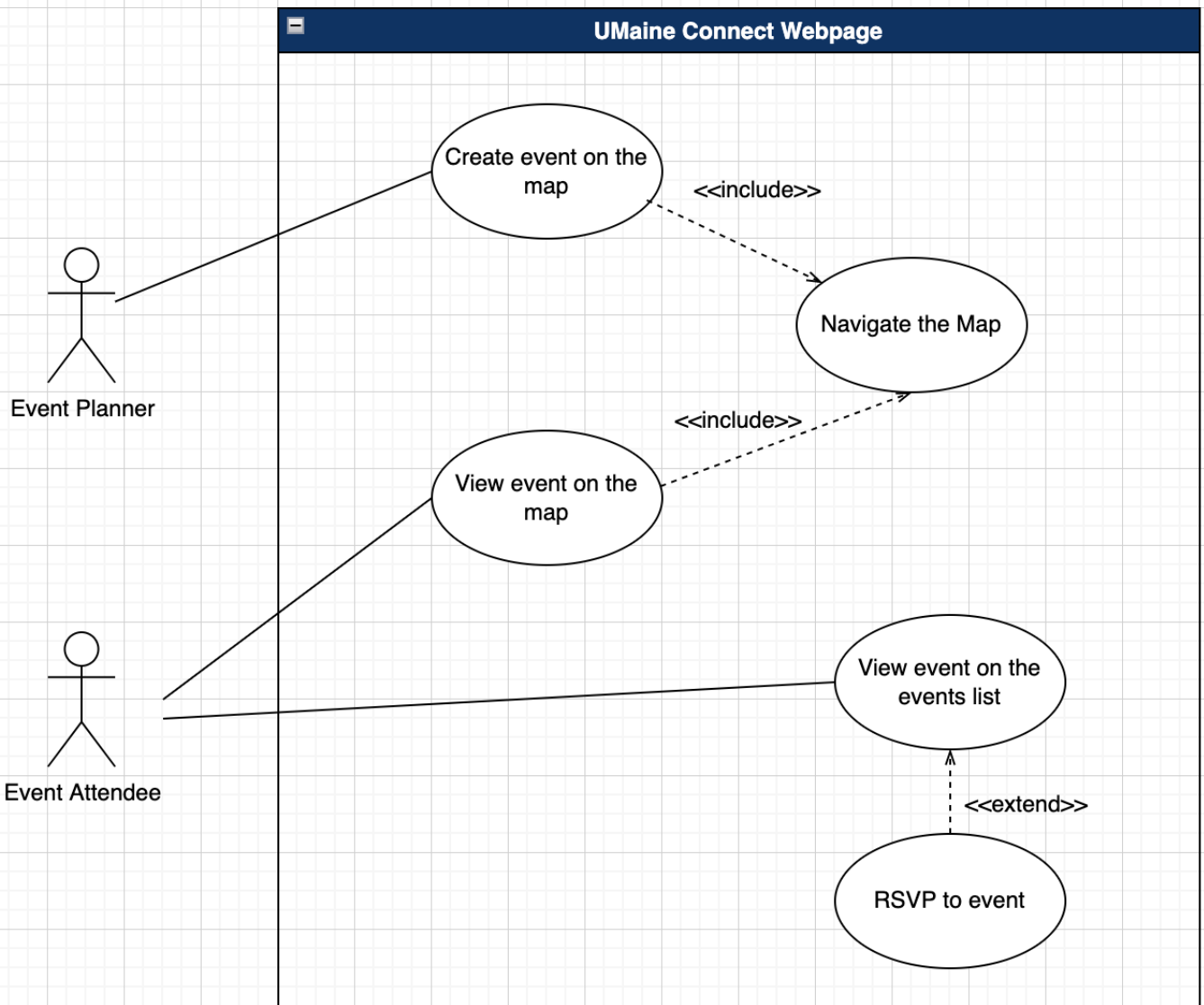

Use Case Models

for UMaine Connect by Maineframe (Group 2)



Use case model 1. Account actions and navigation



Use case model 2. Event actions

Title:	Login
ID:	01
Description:	The user will log in to their account using their credentials on the UMaine Connect webpage.
System Under Design:	UMaine Connect
Primary Actor:	Event Planner or Event Attendee (General User)
Participants:	none
Goal:	Allow a user to access their account using their credentials on the UMaine Connect webpage.
Following Use Cases:	<i>Extended by:</i> Create Account
Invariant:	User must be connected to the internet
Precondition:	User must be connected to the webpage
Success Postcondition:	Success message is displayed by a dialogue box

STEPS: 1. User clicks the login button on the landing page. 3. User inputs email and password. 4. User clicks the login button on the login window.	SYSTEM RESPONSE: 2. System opens the login window. 5. System closes the login window. 6a. System tells the user the login was successful. 6b. System displays an error for invalid email or password and prompts the user to enter their login info again (continue from step 3). 7. System displays the landing page.
---	--

Title:	Create Event on the Map
ID:	02
Description:	A user opens up the map and plots a point where they plan to host an event. The user gives a brief description of the activity and uploads it to the server.
System Under Design:	UMaine Connect
Primary Actor:	Event Planner
Participants:	Event Attendee
Goal:	The goal is to allow Event Planners the ability to quickly and effectively create events on the map for possible Event Attendees to view.
Following Use Cases:	Includes: Navigate the Map
Invariant:	User is connected to the webpage, Google Maps API and Firebase are functioning as intended.
Precondition:	User has their screen over the map, in either full screen or regular view.
Success Postcondition:	Red pointer on the map that holds information related to their event.

STEPS: 1. The user clicks on the Google Map, where they would like to plot their point. 4. The user writes in their event details 5. The user clicks "post".	System Responses: 2. The system creates a new marker on the map where the user clicked 3. The system allows the user to type details about their event. 6. The event is posted on the system.
--	---

Title:	Create Account
ID:	03
Description:	The user will use the “sign up” buttons from the login window in order to create an account.
System Under Design:	UMaine Connect
Primary Actor:	Event Planner or Event Attendee (General User)
Participants:	none
Goal:	Allow a user to create their account using their email and a password or Google credentials on the UMaine Connect webpage.
Following Use Cases:	<i>Extends:</i> Login
Invariant:	User must be connected to the internet
Precondition:	User must be connected to the webpage and be on the login page. If the user is signing up with Google then they must have a Google Account.
Success Postcondition:	Success message is displayed by a dialogue box

STEPS: 1a. User clicks the “sign up” button on the login page. 1b. User clicks the “sign up with Google” button on the login page. 3. User inputs username, email, password, and password confirmation. 4. User clicks the sign up button in the signup window.	SYSTEM RESPONSE: 2a. System opens the signup window. 2b. System takes the user to the Google-specific account creation page. 5a. System tells the user signup was successful. 5b. System displays an error for an invalid email or password and prompts the user to enter their information again (continue from step 3). 6. System closes the signup window.
--	---

Title:	View Event on The Map
ID:	04
Description:	The user will go to the webpage and go to the Google Maps API plugin. The user will click on a red point and view the details of the event.
System Under Design:	UMaine Connect
Primary Actor:	Event Attendee
Participants:	Event Planner
Goal:	The goal is to allow Event Attendees the ability to quickly and effectively view events on the map.
Following Use Cases:	Includes: Navigate the Map
Invariant:	User is connected to the webpage, Google Maps API and Firebase are functioning as intended.
Precondition:	User has their screen over the map, in either full screen or regular view. Other users have posted events to the system.
Success Postcondition:	Pop-up with event information.

STEPS: 1. The user views the Google Map and sees the points. 2. The user clicks on a point that they may be interested in. 4. The user reads the information	System Responses: 3. The system displays information based on that event.
--	---

Title:	RSVP to Event
ID:	05
Description:	After the Event Attendee (user) has read the details to an event, the user can RSVP to the event.
System Under Design:	UMaine Connect
Primary Actor:	Event Attendee
Participants:	Event Planner
Goal:	The goal is to allow Event Attendees the ability to quickly and effectively view events on the map.
Following Use Cases:	Extended by: View Event on Events List
Invariant:	User is connected to the webpage, Google Maps API and Firebase are functioning as intended.
Precondition:	The user is viewing an event information pop-up.
Success Postcondition:	RSVP tally increases after the user confirms they are attending.

STEPS: <ol style="list-style-type: none"> 1. The user reads the event information 2. IF the user decides they would like to attend the event. THEN the user clicks "RSVP" 	System Responses: <ol style="list-style-type: none"> 3. The system adds +1 to the RSVP number. 4. The system updates the view to show the new number of attendees.
--	---

Title:	View Event on Event List
ID:	06
Description:	The user will go to the webpage and go to the right of the Google Maps API. Here the user will see a list of activities that are plotted on the map.
System Under Design:	UMaine Connect
Primary Actor:	Event Attendee
Participants:	Event Planner
Goal:	The goal is to allow Event Attendees the ability to quickly and effectively view events on the list. This is an alternative to viewing events on the map.
Following Use Cases:	Extends: RSVP to event
Invariant:	User is connected to the webpage, Firebase is functioning as intended.
Precondition:	User has their screen over the map, in either full screen or regular view. Other users have posted events to the system. User's mouse pointer is within the boundary of the list.
Success Postcondition:	Pop-up with event information.

<p>STEPS:</p> <ol style="list-style-type: none"> 1. User scrolls through the list view of events. 3. User clicks on an event they are interested in. 5. User views the information relating to the event. The user can now RSVP for the event. 	<p>SYSTEM RESPONSE:</p> <ol style="list-style-type: none"> 2. As the user uses their scroll wheel: <ol style="list-style-type: none"> a) List scrolls up b) List scroll down 4. System updates the view to show the information.
--	--

Title:	Add Friends
ID:	07
Description:	The user will use the add friend button to pull up a window where they can search for friends by their username.
System Under Design:	UMaine Connect
Primary Actor:	Event Planner or Event Attendee (General User)
Participants:	Other General User to be added
Goal:	Allow users to add members to their friends list. This is important for later features to be implemented such as event audience curation.
Following Use Cases:	N/A
Invariant:	User must be connected to the internet
Precondition:	User must be connected to the webpage and be logged into their account.
Success Postcondition:	A user is added to their friends list.

<p>STEPS:</p> <ol style="list-style-type: none"> 1. User clicks the “friends” button on the home page. 3. User clicks the “add friends” option from the dropdown. 5. User enters the username of the friend they would like to add. 6. User clicks the search button. 8. User clicks the desired account from the search results. 	<p>SYSTEM RESPONSE:</p> <ol style="list-style-type: none"> 2. System displays a dropdown menu from the friends button. 4. System opens the add friends window. 7a. The system shows the user a list of accounts that match their search. 7b. The system displays an error message indicating there are no users that match the search. 9. The system adds the friend to the user’s friends list and tells the user the addition was successful.
---	---

Title:	Navigate the Map
ID:	08
Description:	The user will be able to navigate the Google Map plugin using their mouse. The user will be able to zoom in on different points and places.
System Under Design:	UMaine Connect
Primary Actor:	Event Attendee and Event Planner
Participants:	N/A
Goal:	The goal is to allow Event Attendees the ability to quickly and effectively view the map and any event points.
Following Use Cases:	Included By: View Event on the Map, Create Event on the Map
Invariant:	User is connected to the webpage, Google Maps API is functioning as intended.
Precondition:	User has their screen over the map, in either full screen or regular view.
Success Postcondition:	User moves around the map, zooms in and out.

STEPS: 1. User clicks on the Map. 2. User can either navigate in full screen or normal view. 3. User clicks and drags or zooms in using the scroll wheel or '+' and '-' buttons.	SYSTEM RESPONSE: 4. Map position is changed: <ul style="list-style-type: none"> a) map is dragged north bound b) map is dragged south bound c) map is dragged east bound d) map is dragged west bound e) map is dragged some diagonal f) map zooms in g) map zooms out 5. View is updated.
--	---

Title:	View Friends List
ID:	09
Description:	The user will use the friends list button to pull up a window where they can view their friends.
System Under Design:	UMaine Connect
Primary Actor:	Event Planner or Event Attendee (General User)
Participants:	none
Goal:	Allow users to see their friends in a list view.
Following Use Cases:	N/A
Invariant:	User must be connected to the internet
Precondition:	User must be connected to the webpage and be logged into their account.
Success Postcondition:	A user views their friends list on the webpage

<p>STEPS:</p> <ol style="list-style-type: none"> 1. User clicks the “friends” button on the home page. 3. User clicks the “friends list” option from the dropdown. 	<p>SYSTEM RESPONSE:</p> <ol style="list-style-type: none"> 2. System displays a dropdown menu from the friends button. 4. System opens the friends list window.
---	--

Title:	View Static Pages
ID:	10
Description:	Users who access the UMaine Connect website will be able to view different pages within the website. The landing page will have buttons that have a link reference to other pages. Some other pages may be: Privacy Policy, Terms and Conditions, About, Login, logout, Contact, Account, Friends.
System Under Design:	UMaine Connect
Primary Actor:	Event Attendee, Event Planner
Participants:	N/A
Goal:	Create an efficient and intuitive way for users to navigate the website.
Following Use Cases:	N/A
Invariant:	Internet connection
Precondition:	User is using a standard web browser and has searched for the IP of our webserver with the specified port 443.
Success Postcondition:	User is on the webpage, and can view other pages besides the landing page.

STEPS: <ol style="list-style-type: none"> 1. User connects to the UMaine Connect Website 3. User clicks on any of the buttons on the landing page. 	SYSTEM RESPONSE: <ol style="list-style-type: none"> 2. Website Displays landing page 4. When the user clicks on a button, it links to another page: <ol style="list-style-type: none"> a) Privacy Policy b) Terms and Conditions c) About d) Login e) Logout f) Contact g) Account h) Friends.
---	--

Title:	Logout
ID:	11
Description:	The user will log out of their account by clicking the logout button.
System Under Design:	UMaine Connect
Primary Actor:	Event Planner or Event Attendee (General User)
Participants:	none
Goal:	Allow a user to log out of their account on the UMaine Connect webpage.
Following Use Cases:	N/A
Invariant:	User must be connected to the internet
Precondition:	User must be connected to the webpage and signed into their account
Success Postcondition:	Success message is displayed by a dialogue box

STEPS: 1. User clicks the logout button on the landing page.	SYSTEM RESPONSE: 2. System logs the user out of their account. 3. System tells the user that they have been successfully logged out.
--	---