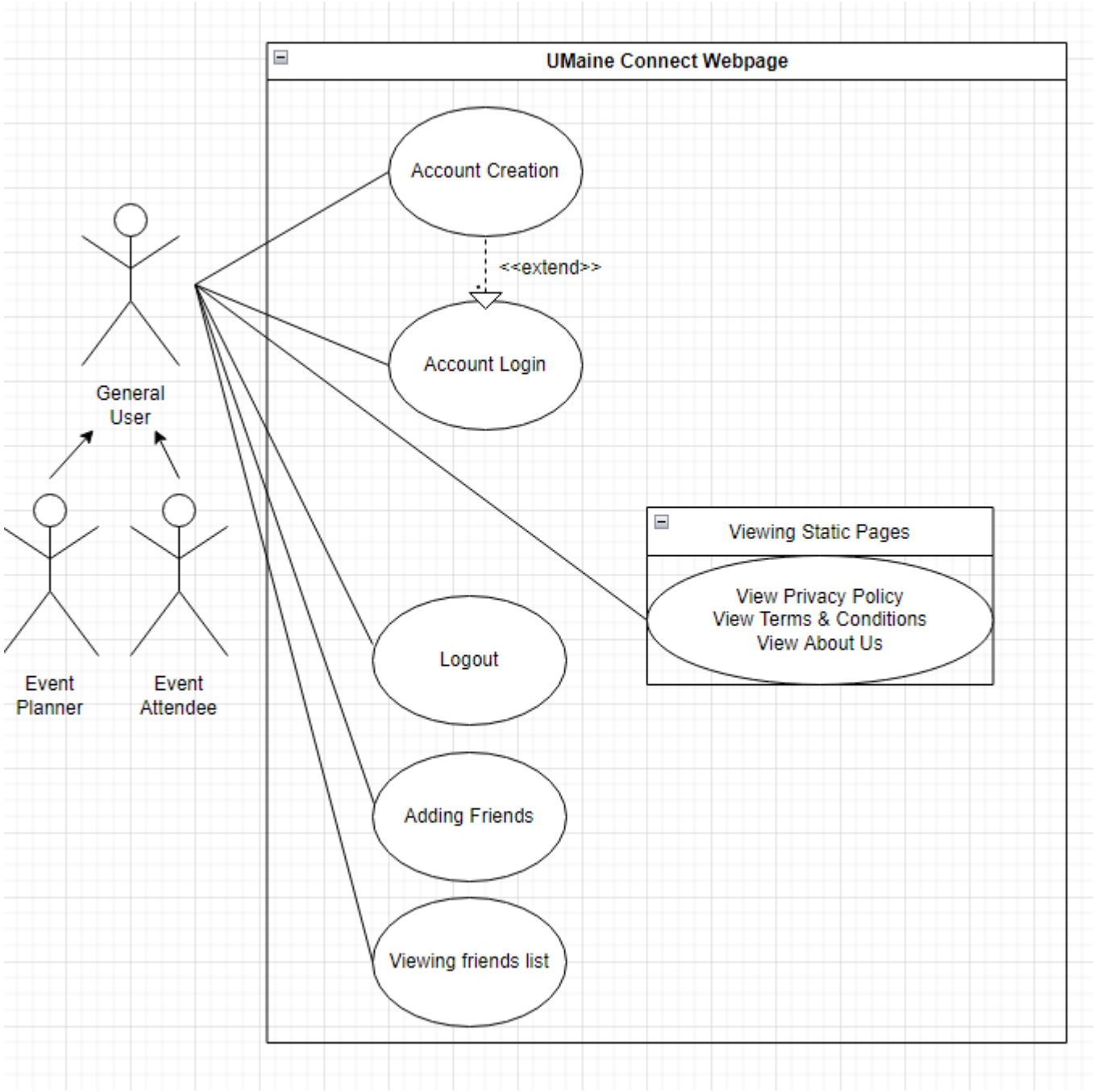
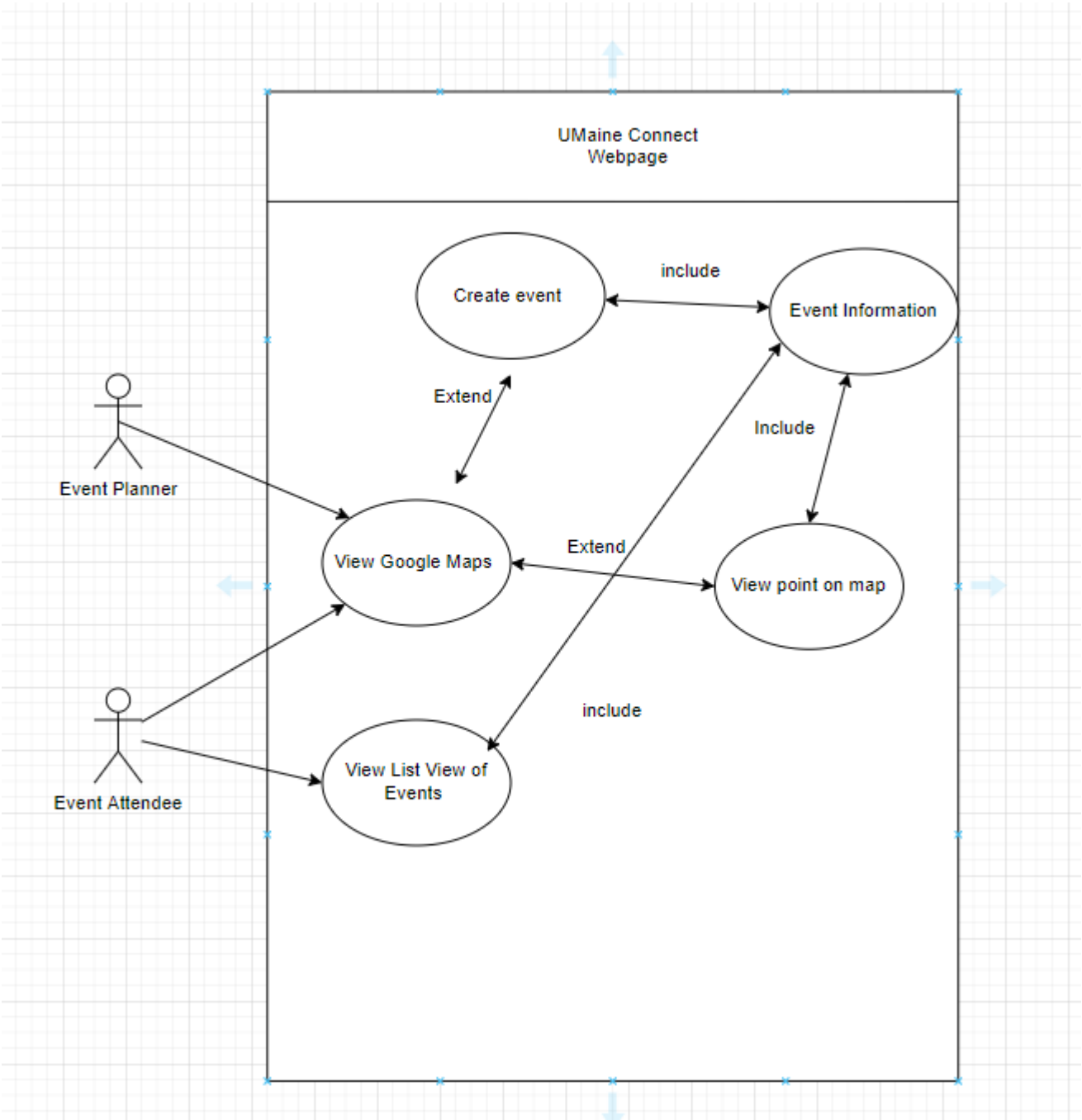
Use Case Models





Title:	**Account Login**
ID:	**01**
Description:	The user will log in to their account using their credentials on the Umaine connect webpage.
System Under Design:	Firebase
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:	Create Account, Logout
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:	**SYSTEM RESPONSE:**
	0. The system displays the login button on landing page
1. User clicks the login button	
	2. Login window is opened by the system
3a. User clicks on email field 3b. User clicks the “sign up” button 3c. User clicks the “sign up with Google” button	
	4a. The system allows for text box submission under the email field 4b. The system redirects the user to the account creation page 4c. The system redirects the user to the Google-specific account creation page.
5. User inputs email 6.User clicks on password field	
	7. The system allows for text box submission under the password field
8.User inputs password 9.User clicks login button on login window	
	10.System closes login window
	11a. System returns success message 11b. System Displays Error for invalid username or password and the use case continues from step 5.
	12. System displays 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:	This action utilizes the Google Maps API as well as Firebase for storing activity data.
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:	View Event on 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:	**System Responses**:
1. The Event Planner (user) uses their search engine to go to the IP the system is hosted on. 3. The user clicks on the Google Map, where they would like to plot their point. 6. The user inputs the information related to their events.. 7. The user clicks “post”.	2. The system shows the page “index.html” 4. The system creates a point at the longitude and latitude where the user clicked. 5. The system displays boxes to input information related to their event. 8. The system uploads the new data to the database, and will now show this point to all users accessing the system.

Title:	**Account Creation**
ID:	**03**
Description:	The user will use the “sign up” buttons from the login window in order to create an account.
System Under Design:	Firebase
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:	Login, Logout
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:	**SYSTEM RESPONSE:**
	0. The system displays the “sign up” and “sign up with Google” buttons on the login page.
1a. User clicks the “sign up” button 1b. User clicks the “sign up with Google” button	
	2a. Sign up window is opened by the system 2b. The system redirects the user to the Google-specific account creation page.
3. User clicks on username field	
	4. The system allows for text box submission under the username field
5. User inputs username 6. User clicks on email field	
	7. The system allows for text box submission under the email field
8. User inputs email 9. User clicks on password field	
	10. The system allows for text box submission under the password field
11. User inputs password 12. User clicks on confirm password field	
	13. The system allows for text box submission under the confirm password field
14. User inputs password again	
	15a. System returns success message 15b. System Displays Error for invalid email or password and the use case continues from step 4.
	16. System closes 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:	This action utilizes the Google Maps API as well as Firebase for storing activity data.
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:	RSVP to an Event
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:	**System Responses**:
1. The Event Attendee (user) uses their search engine to go to the IP the system is hosted on. 3. The user views the Google Map and sees the points. 4. The user clicks on a point that they may be interested in. 7. The user views the pop-up with the	2. The system shows the page “index.html” 5. The system pulls the data relating to the event from the database. 6. The system shows a pop-up with the information that the Event Planner posted.

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:	This action utilizes the Google Maps API as well as Firebase for storing activity data.
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:	N/A
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:	**System Responses**:
1. The user reads the event information 2. IF the user decides they would like to attend the event. THEN the user clicks “RSVP”	3. The system adds +1 to the RSVP number. 4. The system updates the view to show the new number of attendees.

Title:	**View Events 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:	This action uses Firebase to store all activities that will be shown on the list.
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:	RSVP to an 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.

STEPS:	**SYSTEM RESPONSE**:
1.User scrolls through the list view of events. 3. User clicks on an event they are interested in. 5. User views the pop-up with information relating to the event. The user can now RSVP or close out of the pop-up.	2. As the user uses their scroll wheel: a) List scrolls up b) List scross down 4. System pulls event data from the database and then updates the view to show the information.

Title:	**Friend Addition**
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:	Firebase
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:	View friends list
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.

STEPS:	**SYSTEM RESPONSE:**
	0. The system displays the “friends” tab on the login page.
1. User clicks the “friends” button	
	2. System drops down a menu from the friends button
3. User clicks the “add friends” button	
	4. System opens up the add friends window
5. User clicks on the textbox for their friend’s username.	
	6. The system allows for text box submission under the username field
7. User inputs their friend's username 8. User clicks on the search button	
	9a. The system displays search results 9b. The system responds with an error message indicating there is no user with that username.
10. User clicks desire account from search results	
	11. The system adds the friend and returns a success message

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:	This action utilizes the Google Maps API.
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:	View 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 Google Maps plugin. 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: 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:	**Viewing Friend’s 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:	Firebase
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:	Friend Addition
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

STEPS:	**SYSTEM RESPONSE:**
	0. The system displays the “sign up” and “sign up with Google” on the login page.
1. User clicks the “friends” button	
	2. System drops down a menu from the friends button
3. User clicks the “friends list” button	
	4. System opens up the friends list window
5. User views their friends list	

Title:	**View Different Webpages**
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:	Website
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:	Login, logout , view account, view friends
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:	**SYSTEM RESPONSE**:
1. User searches the web for the UMaine Connect IP Address. 3. User clicks on any of the buttons on the landing page.	2. System accepts traffic from the user and sends back website data 4. When the user clicks on a button, it links to another page: a) Privacy Policy b) Terms and Conditions c) About d) Login e) Logout f) Contact g) Account h) Friends.

Title:	**Account Logout**
ID:	**11**
Description:	The user will logout of their account by clicking the logout button.
System Under Design:	Firebase
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:	Create Account, Log in

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:	**SYSTEM RESPONSE:**
	0. The system displays the logout button on landing page
1. User clicks the logout button	
	2. System logs user out of their account 3. System returns successful logout message