Account Management Interface

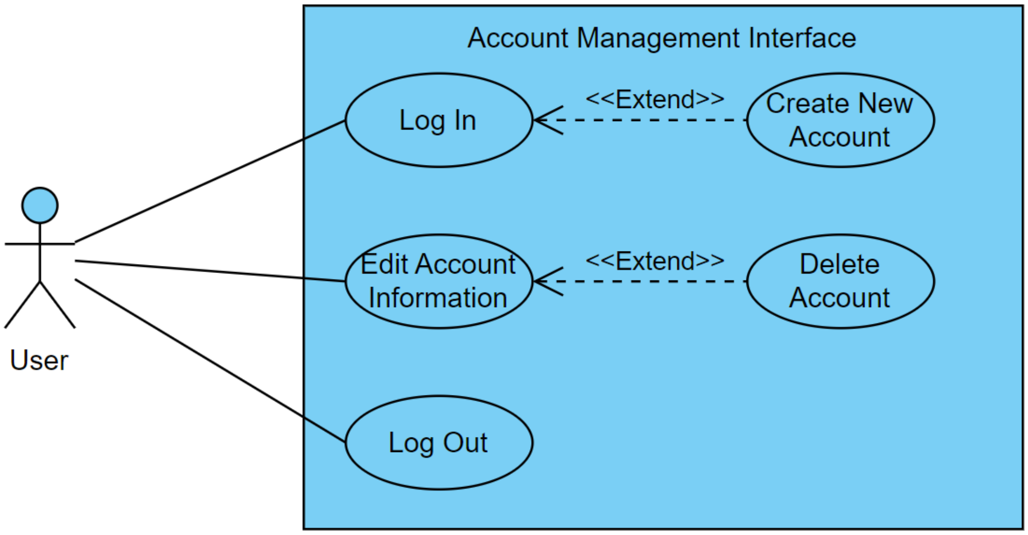
Title:	**CreateNew Account**
Description:	User registers new account with Login Repository to use for repeated login
System Under Design:	
Primary Actor:	User
Participants:	
Goal:	New account created for user
Related Use Cases:	Specialization of: “Login”
Invariant:	
Precondition:	UI displaying “Login” interface with “Create A New Account” button available; Database with available space to store new account
Success Postcondition:	Database updated with new account information; UI displaying “Login” interface
Steps:	
Actor Action:	**System Response:**
1. Choose the “Login” button.	2. Transfer to the “Login” interface; display “Username” and “Password” text input fields, and “Create New Account” button.
3. Choose the “Create New Account” button.	4. Transfer to the “Create New Account” interface; display “First Name”, “Last Name”, “Username”, “Create Password”, and “Verify Password” text input fields, and “Create New Account” button.
5. Inputs information into necessary text fields.	6. Verifies inputted passwords match.
7. Choose the “Create New Account” button.	8. Inputted information saved to “Login” repository in database
	9. Transfer to the “Travel Logs” interface.

Title:	**Login**
Description:	User provides stored account information to access saved information
System Under Design:	
Primary Actor:	User

Participants:	
Goal:	User successfully “signs in”; User accesses stored information linked to account
Related Use Cases:	
Invariant:	
Precondition:	UI displaying “Login” button; Database contains account login credentials to authenticate successful login
Success Postcondition:	Database verifies account information provided matches stored information; UI displays data connected to account user logged in to
Steps:	
Actor Action:	**System Response:**
1. Choose the “Login” button.	2. Transfer to the “Login” interface.
	3. Display “username” and “password” text input fields, “Login” button.
4. Inputs information into text input fields.	
5. Choose the “Login” button.	6. Verify inputted information matches stored information from the database.
	7. Transfer to “Travel Logs” interface.

Title:	**Edit Account Information**
Description:	User registers new account with Login Repository to use for repeated login
System Under Design:	
Primary Actor:	User
Participants:	
Goal:	New account created for user
Related Use Cases:	Specialization of: “Login”
Invariant:	
Precondition:	UI displaying “Login” interface with “Create A New Account” button available; Database with available space to store new account
Success Postcondition:	Database updated with new account information; UI displaying “Login” interface
Steps:	

Actor Actions:	**System Response:**
1. Choose the “Edit Account Information” button.	2. Transfer to the “Edit Account Information” interface
	3. Display text input fields containing information saved to account.
4. Choose input fields to make edits to.	
5. Choose the “Save Changes” button	6. Database updates information saved to account based on new information provided to text input fields.
	7. Transfer to the “Travel Logs” interface.



Exploratory Interface:

Title:	**Search Based On Criteria**
Description:	User searches for particular features with specific criteria
System Under Design:	
Primary Actor:	Logged-In
Participants:	
Goal:	Find a specific feature that fits the provided criteria
Related Use Cases:	Necessary for Find Available Flights, Find Hotels, Find Restaurants, Find Car Rentals, Find Trains, Find Buses, and Find Local Attractions
Invariant:	
Precondition:	User is logged-in
Success Postcondition:	The user is able to log-in
Steps:	
Actor Action:	**System Response:**
1. User selects the "Search" button. 3. User inputs item they're searching for 5. User checks additional criteria options	2. Transfer to the Exploratory Interface and display the search field and search criteria options. 4. System displays options that fit the type of item user entered 6. System removes items from display that don't meet new criteria

Title:	**Find Available Flights**
Description:	User searches for available flights
System Under Design:	
Primary Actor:	Logged-In
Participants:	
Goal:	Find available flights to user’s destination from their specified location
Related Use Cases:	Includes Search Based on Criteria and base case for Find Alternative Flights
Invariant:	
Precondition:	User is logged-in
Success Postcondition:	The user is able to log-in
Steps:	
Actor Action:	**System Response:**
1. User selects the “Search” button. 3. User inputs flight they’re searching for 5. User adds flight date/time, destination, or departure location	2. Transfer to the Exploratory Interface and display the search field and search criteria options. 4. System displays available flights that fit the type of item user entered 6. System removes items from display that don’t meet new criteria

Title:	**Find Alternative Flights**
Description:	User searches for alternative flights to a specified flight
System Under Design:	
Primary Actor:	Logged-In
Participants:	
Goal:	Find specific alternative flights to user’s destination from their specified location and time
Related Use Cases:	Extension of Find Available Flights

Invariant:	
Precondition:	User is logged-in
Success Postcondition:	The user is able to search for available flights
Steps:	
Actor Action:	**System Response:**
1. User selects the “Search” button. 3. User inputs flight they’re searching for 5. User adds original flight date/time, destination, or departure location	2. Transfer to the Exploratory Interface and display the search field and search criteria options. 4. System displays available flights that fit the type of item user entered 6. System displays flights from specified departure location either to specified destination or to connecting flight to specified destination

Title:	**Find Hotels**
Description:	User searches for hotels at their destination
System Under Design:	
Primary Actor:	Logged-In
Participants:	
Goal:	Find available hotels at a user’s specified destination
Related Use Cases:	Includes Search Based on Criteria
Invariant:	
Precondition:	User is logged-in
Success Postcondition:	The user is able to log-in
Steps:	
Actor Action:	**System Response:**
1. User selects the “Search” button. 3. User inputs search for hotels and destination	2. Transfer to the Exploratory Interface and display the search field and search criteria options. 4. System displays hotels in the user’s specified destination or within 30 miles of specified destination

Title:	**Find Restaurants**
Description:	User searches for restaurants at their destination
System Under Design:	
Primary Actor:	Logged-In
Participants:	
Goal:	Find available restaurants at a user's specified destination
Related Use Cases:	Includes Search Based on Criteria
Invariant:	
Precondition:	User is logged-in
Success Postcondition:	The user is able to log-in
Steps:	
Actor Action:	**System Response:**
1. User selects the "Search" button. 3. User inputs search for restaurants and destination	2. Transfer to the Exploratory Interface and display the search field and search criteria options. 4. System displays restaurants in the user's specified destination or within 30 miles of specified destination

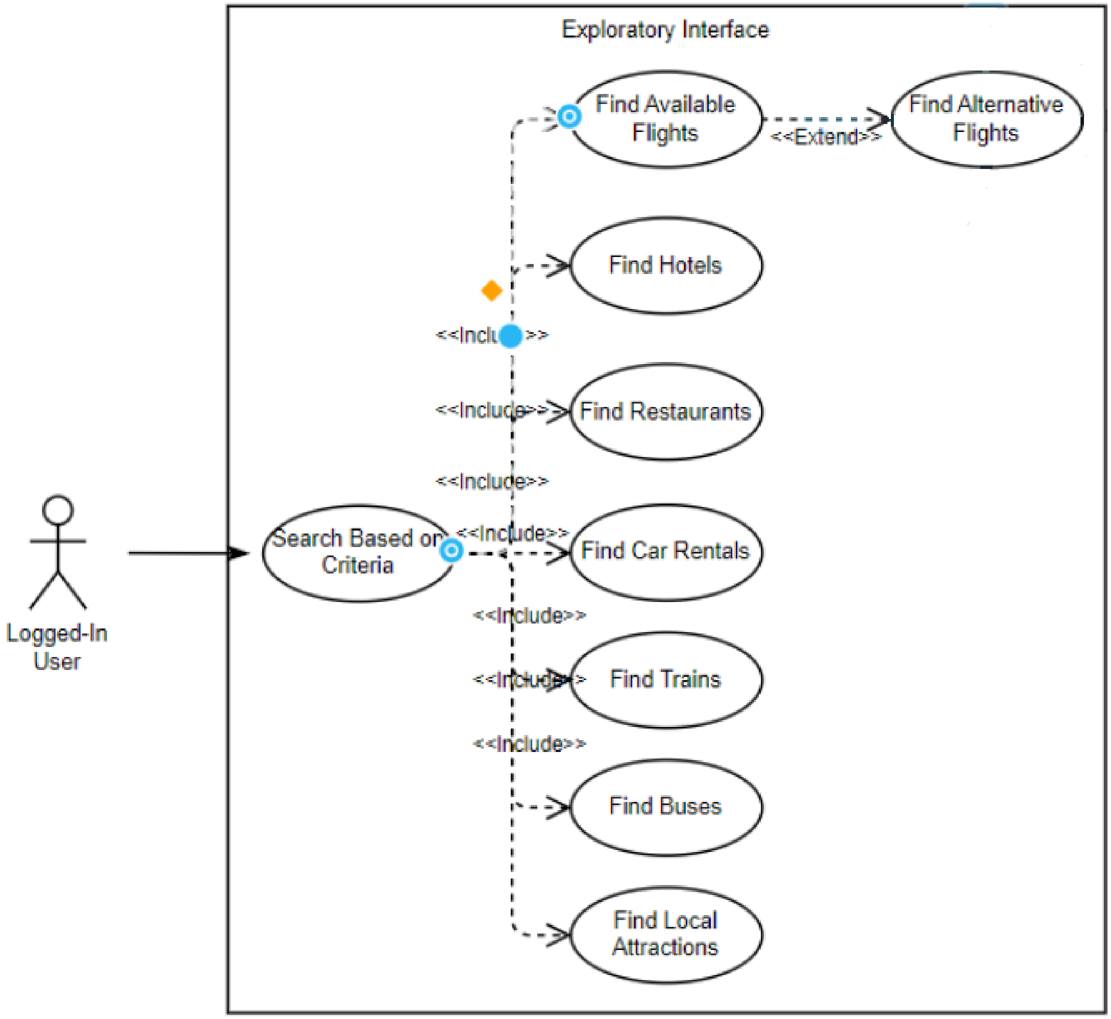
Title:	**Find Car Rentals**
Description:	User searches for cars available for rent at specified location
System Under Design:	
Primary Actor:	Logged-In
Participants:	
Goal:	Find cars available for rent based on a user’s specified area/destination
Related Use Cases:	Includes Search Based on Criteria
Invariant:	
Precondition:	User is logged-in
Success Postcondition:	The user is able to log-in
Steps:	
Actor Action:	**System Response:**
1. User selects the “Search” button. 3. User inputs search for car rentals	2. Transfer to the Exploratory Interface and display the search field and search criteria options. 4. System displays available car rentals in the user’s specified are or within 30 miles of specified area

Title:	**Find Trains**
Description:	User searches for available trains from one specified location to another
System Under Design:	
Primary Actor:	Logged-In
Participants:	
Goal:	Find available trains from a specified departure location to a specified destination
Related Use Cases:	Includes Search Based on Criteria
Invariant:	

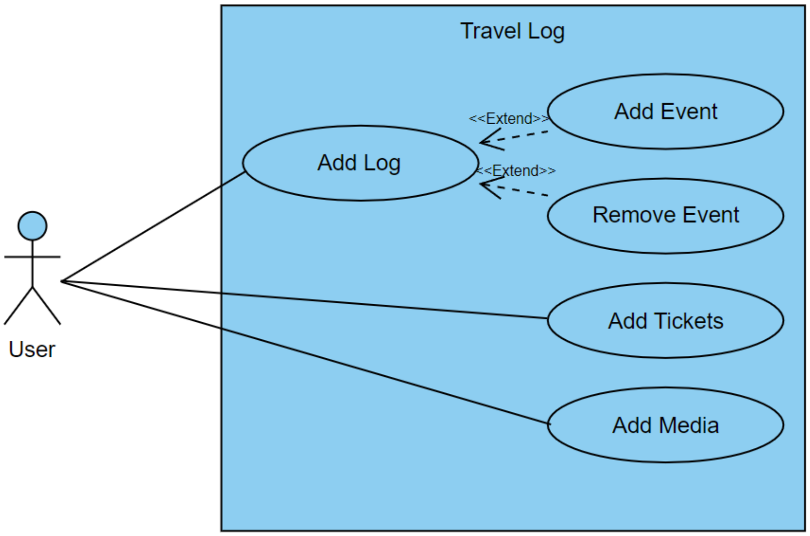
Precondition:	User is logged-in
Success Postcondition:	The user is able to log-in
Steps:	
Actor Action:	**System Response:**
1. User selects the “Search” button. 3. User inputs search for trains align with intended destination and/or departure location	2. Transfer to the Exploratory Interface and display the search field and search criteria options. 4. System displays available trains that fit the specified criteria

Title:	**Find Buses**
Description:	User searches for available buses from one specified location to another
System Under Design:	
Primary Actor:	Logged-In
Participants:	
Goal:	Find available buses from a specified departure location to a specified destination
Related Use Cases:	Includes Search Based on Criteria
Invariant:	
Precondition:	User is logged-in
Success Postcondition:	The user is able to log-in
Steps:	
Actor Action:	**System Response:**
1. User selects the “Search” button. 3. User inputs search for buses align with intended destination and/or departure location	2. Transfer to the Exploratory Interface and display the search field and search criteria options. 4. System displays available buses that fit specified criteria

Title:	**Find Local Attractions**
Description:	User searches for attractions near their specified location
System Under Design:	
Primary Actor:	Logged-In
Participants:	
Goal:	Find attractions within 30 miles of their specified location
Related Use Cases:	Includes Search Based on Criteria
Invariant:	
Precondition:	User is logged-in
Success Postcondition:	The user is able to log-in
Steps:	
Actor Action:	**System Response:**
1. User selects the “Search” button. 3. User inputs search for local attractions with their specified location	2. Transfer to the Exploratory Interface and display the search field and search criteria options. 4. System displays known attractions within 30 miles of the user’s specified location



Travel Log/Itinerary use case model and descriptions
Created by: Jeffery Parent



Title:	**Add Log**
Description:	**Successfully add a log to the travel log page**
System Under Design:	**Travel Log system**
Primary Actor:	**User**
Participants:	
Goal:	**Add a log to the travel log page**
Following Use Cases:	**Add Event , Remove Event**
Invariant:	
Precondition:	**User is on travel log homepage**
Success Postcondition:	
STEPS: 1. SYSTEM: System shows a button on the page with a “+” symbol 2. ACTOR: User presses the button 3. SYSTEM: Opens popup with fields to input various data 4. ACTOR: User inserts at least a name and presses continue button 5. SYSTEM: shows the newly crated travel log with another “+” button to add events	**ALTERNATIVES**:

Title:	**Add Event**
Description:	**User successfully adds an event to their travel log**
System Under Design:	**Travel Log system**
Primary Actor:	**User**
Participants:	
Goal:	**Add an event to a travel log**
Following Use Cases:	
Invariant:	
Precondition:	**User is on created travel log page(must have already created a log)**
Success Postcondition:	
STEPS: 1. ACTOR: User selects "+" button 2. SYSTEM: Shows search bar and list of any booked places 3. ACTOR: user either selects a booked place, or searches and adds their own 4. SYSTEM: Adds events to the travel log in chronological order	**ALTERNATIVES**:

Title:	**Remove Event**
Description:	**User successfully removes an event to their travel log**
System Under Design:	**Travel Log system**
Primary Actor:	**User**
Participants:	
Goal:	**Remove an event on a travel log**
Following Use Cases:	
Invariant:	
Precondition:	**User is viewing a travel log with events already added**
Success Postcondition:	
STEPS: 1. ACTOR: User selects an event to edit 2. SYSTEM: Shows the event on screen with info 3. ACTOR: User selects delete button at bottom of event 4. SYSTEM: Removes event from travel log and places it in "trash"	**ALTERNATIVES**:

Title:	**Add Tickets**
Description:	**User successfully adds tickets to an event in their travel log**
System Under Design:	**Travel Log system**
Primary Actor:	**User**
Participants:	
Goal:	**Add tickets to events**
Following Use Cases:	
Invariant:	
Precondition:	**User has added events to a travel log**
Success Postcondition:	
STEPS: 1. ACTOR: User selects event in log 2. SYSTEM: shows the event on screen with info 3. ACTOR: selects edit event button 4. SYSTEM: shows user fields to add more info and a add tickets/booking details field 5. ACTOR: users attaches tickets to field	**ALTERNATIVES**: 2. SYSTEM: Shows button to purchase tickets 3. ACTOR: User selects purchase tickets button and is transferred to third party site 4. SYSTEM: After purchase tickets are shown in the event details

Title:	**Add Media**
Description:	**User successfully adds media to an event in their travel log**
System Under Design:	**Travel Log system**
Primary Actor:	**User**
Participants:	
Goal:	**Add pictures and videos to events**
Following Use Cases:	
Invariant:	
Precondition:	**User has events in travel log**
Success Postcondition:	

STEPS: 1. ACTOR: User selects event in log 2. SYSTEM: Shows the event on screen with info 3. ACTOR: Selects add photos/videos button at the top of the event 4. SYSTEM: Asks for media/file permission from the user and opens a gallery of permission granted.	**ALTERNATIVES**: