TraViz Use Case Scenarios

Author (s): **Matthew Varona, Gopika Krishnan, Arshiya Khattak**
Date: **November 8, 2021** **Version: 1.0**

USE CASE NAME:	**Register**	**USE CASE TYPE** **Business Requirements:**
USE CASE ID:	001	
PRIORITY:	Medium	
PRIMARY BUSINESS ACTOR:	User	
SHORT DESCRIPTION:	Allows users to register new accounts.	
PRE-CONDITION:	None	
TRIGGER:	User navigates to register page	
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**
	Step 1: User enters desired username and password into the text fields.	
	Step 2: User clicks the “Register” button.	**Step 3**: System checks if the username is taken.
		Step 4: System checks if the password is a valid password (minimum 8 characters).
		Step 5: Once all checks have been passed, the account is saved to the database of the website..
ALTERNATE COURSES:	**Step 3a.**: If the username is already taken, the system returns to the register page and displays the message “That username is taken.”	
	Step 3b.: If the password is invalid, the system returns to the register page and displays the message “That password is invalid.”	
CONCLUSION:	The user has been logged into the system.	
POST-CONDITION:	Registration is complete.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None	

USE CASE NAME:	**Login**	**USE CASE TYPE** **Business Requirements:**
USE CASE ID:	002	
PRIORITY:	Medium	
PRIMARY BUSINESS ACTOR:	User	
SHORT DESCRIPTION:	Allows users to input username and password. If these match an account, the user is logged in.	
PRE-CONDITION:	None	
TRIGGER:	User navigates to login page	
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**
	Step 1: User enters username and password into the text fields.	
	Step 2: User clicks the “Login” button.	**Step 3**: System checks if the username and password combination is valid.

		Step 4: After verifying the username and password, the system navigates to the account page of the user.
ALTERNATE COURSES:	**Step 3a.**: If error checking fails (e.g. the login credentials do not exist or are invalid), then the site returns to the login page, displaying “Invalid username”/”Invalid password” depending on the situation.	
CONCLUSION:	The user has been logged into the system.	
POST-CONDITION:	Login is complete.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None	

USE CASE NAME:	**Add a Trip**	**USE CASE TYPE** **Business Requirements:**
USE CASE ID:	003	
PRIORITY:	High	
PRIMARY BUSINESS ACTOR:	User	
SHORT DESCRIPTION:	Allows users to input trips that they have previously taken.	
PRE-CONDITION:	User must be logged in	
TRIGGER:	User navigates to visualization page	
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**
	Step 1: User clicks the “Add a Trip” button.	
	Step 2: User starts typing the name of the starting location of the trip.	**Step 3**: System sends this text to the geocoding API, and in return receives a list of possible locations (including coordinates which are not shown to the user but are saved internally)
	Step 5: User selects the appropriate location from the menu	**Step 4**: System displays the suggested locations in a dropdown menu, similar to Google search suggestions
	Step 6: User starts typing the name of the destination of the trip.	**Step 7**: System sends this text to the geocoding API, and in return receives a list of possible locations (including coordinates which are not shown to the user but are saved internally)
	Step 9: User selects the appropriate location from the menu	**Step 8**: System displays the suggested locations in a dropdown menu, similar to Google search suggestions
	Step 10: User inputs the date of the trip	
	Step 11: User selects the mode of transport (flight, bus, etc.) from a dropdown menu	
	Step 12: User uploads any images of the trip by clicking the “Upload Image” button.	
	Step 13: User clicks the “Submit Trip” button	**Step 14:** The system checks when the date of the trip is/was.

		Step 15: If the trip has already occurred, the trip information is saved straight into the user's account as a previous trip.
ALTERNATE COURSES:	**Step 13a.** If the file uploaded is not a valid image (.jpg or .png), an error message pops up saying "The image is invalid", and the site remains on the same page.	
	Step 14a. If the date of the trip is in the future, this is a planned trip.	
	Step 14b. The system sends the destination of the trip to the Sygic Travel API, receiving in return a list of 5 possible tourist spots within the destination.	
	Step 14c. If the mode of transportation is "flight", the system sends the origin, destination, and date of the trip to the Travelpayouts API, receiving in return a list of 5 possible flight prices.	
	Step 14d. The trip is saved in the user's account as a planned trip.	
CONCLUSION:	A new trip has been added to the user's account	
POST-CONDITION:	"Add new trip" is complete.	
NOTES:	This use case scenario encompasses multiple bubbles from the diagram: "Input Previous Trip", "Auto-suggest Locations", "Plan Trip", "Provide Tourist Attractions", "Provide Travel Cost".	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None	

USE CASE NAME:	**Visualize Trips**	**USE CASE TYPE** **Business Requirements:**
USE CASE ID:	004	
PRIORITY:	High	
PRIMARY BUSINESS ACTOR:	User	
SHORT DESCRIPTION:	Visualizes the trips that the user has inputted.	
PRE-CONDITION:	User must be logged in, and must have inputted at least 1 trip.	
TRIGGER:	User navigates to visualization page	
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**
		Step 1: The system accesses the list of trips currently associated with this session (the user's saved trips).
		Step 2: The system accesses the list of trips currently associated with this session (the user's saved trips).
		Step 3: For each trip: Based on the starting location coordinates of the trip, the system shows a dot on the world map indicating the start location.
		Step 4: For each trip: Based on the destination coordinates of the trip, the system shows a dot on the world map indicating the destination.
		Step 5: For each trip: The system draws an arc between the origin and destination.
		Step 6: For each trip: The color of all of the above visuals are dependent on the mode of transport (yellow for planes, blue for buses, purple for boats).

	Step 7: The user hovers their mouse over the dot or arc of one of the trips.	**Step 8:** The system displays a popup card for that trip, listing the start and end location, the date, the mode of transport, and any images that the user has uploaded.
ALTERNATE COURSES:	**Step 13a.** If the date of the trip is in the future, this is a planned trip.	
	Step 13b. The system sends the destination of the trip to the Sygic Travel API, receiving in return a list of 5 possible tourist spots within the destination.	
	Step 13c. If the mode of transportation is “flight”, the system sends the origin, destination, and date of the trip to the Travelpayouts API, receiving in return a list of 5 possible flight prices.	
	Step 13d. The trip is saved in the user’s account as a planned trip.	
CONCLUSION:	The travel data is visualized.	
POST-CONDITION:	“Visualize travel map” is complete.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None	

USE CASE NAME:	**Export Map**	**USE CASE TYPE** **Business Requirements:**
USE CASE ID:	005	
PRIORITY:	Low	
PRIMARY BUSINESS ACTOR:	User	
SHORT DESCRIPTION:	Exports an image of the user’s travel map	
PRE-CONDITION:	User must be logged in, and must have inputted at least 1 trip. The travel visualization (use case 004) must already have loaded.	
TRIGGER:	User clicks the “Export” button	
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**
	Step 1: User clicks the “Export” button	**Step 2**: Using D3.js built-in functionality, the system generates a .jpg image of the travel map, containing the user’s trips.
		Step 3: The site sends the image to the user in the form of a browser download.
ALTERNATE COURSES:		
CONCLUSION:	The travel map is exported as an image.	
POST-CONDITION:	“Export travel map” is complete.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None	

USE CASE NAME:	**Recommend locations**	
USE CASE ID:	006	
PRIORITY:	Low	

PRIMARY BUSINESS ACTOR:	User	
SHORT DESCRIPTION:	Recommends possible destinations to the user	
PRE-CONDITION:	User must be logged in, and must have inputted at least 3 trips.	
TRIGGER:	Recommendations appear constantly on the webpage.	
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**
	Step 1: More than 3 trips exist in the user's profile.	**Step 2**: The system sends the user's previously visited locations to the Sygic Travel API. It receives in response the tourism categories of the locations.
		Step 3: The system feeds these categories/tags to a machine learning recommender system.
		Step 4: The machine learning recommender system suggests 3 destinations with similar tags to the user's previous destinations.
		Step 5: The system displays the 3 similar destinations as 3 cards underneath the map visualization.
ALTERNATE COURSES:	None	
CONCLUSION:	The system has recommended locations to the user.	
POST-CONDITION:	"Recommend locations" is complete.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None	