

Use Cases

Use-case: LoginUser

Primary actor: user

Goal in context: Allow the user to login with Facebook (and only with Facebook if they want sharing and seeing Facebook friends. They can also sign up the usual way and then login with site-specific credentials. Facebook will allow us to share the user trips safely assuming they made FB friends they know, and will also make it easy to see who else they know are using the application).

Preconditions: User has already registered.

Trigger: User clicks on the Login button after first filling in all fields and registering.

Scenario:

1. User does not have account. He does not have Facebook account. He types in his name, email, and password. If he already has a Facebook account, he will register his Facebook account with us through OAuth. Either way, after registering, he is automatically logged in.
2. User is redirected to his profile home page, which will contain the list of friends on the left hand bar, and a list of all the trips he has shared on the main content area. There is also a 'create trip' button at the top of the page.

Exceptions:

1. User fails to select the proper syntax required for registration (no non-alphanumerics). We will alert with a red message on the home page.

Priority: Essential, must be implemented.

When available: First increment.

Frequency of use: Necessary for every login

Channel to actor: Via web browser.

Open Issues:

1. Do we need to differentiate between accounts made on our site and the ones using Facebook OAuth?
2. What fields are required?

Use-case: CreateTrip

Primary actor: user, superuser

Goal in context: Allow the user to create trip from scratch - be it in the future (plan), now(take), or the past (share).

Preconditions: User has already registered and is authenticated

Trigger: User clicks on the 'Create trip' button in the user's profile page

Scenario:

1. User logs in and in his home page clicks on the 'create trip' button.
2. User enters the start/end time of the trip. Three possibilities, all changed dynamically through AJAX. If start time after or equal to the current time, then it is a future trip (plan); if it includes today, but started before today, then it is a current trip (take); otherwise, it is a past trip (share). For plan/take refer to points **3-8**; for share, points **9-11**. **Past trips are only different in that some options (like day-to-day planning) are optional.**

plan new/current trip:

3. User is at the trip home page. User enters the required details of the basic itinerary of the trip - starting and ending locations, start and end dates. The application will automatically update with the total mileage, total time to drive estimate. Suggestions of where to go for attractions base on user interests and food, lodging, gas on preferred expense level (frugal, standard, luxurious) can be seen when the user clicks on one of the icons on the right. Those icons are optional features.
4. User now can select any of the days and the map will automatically update the starting and ending point markers on the Google map, with the ending point draggable. Aforementioned vital statistics will be updated on the fly.
5. User can select different ways to travel the distance for any particular day - there are three main options: fastest(don't care for attractions), carefree(most attractions), or standard(default, trade off between getting there on time and visiting as many places as possible. These options are on the right bar.
6. User will also be given optional fields to select, like places to lodge, eat, etc. that are non-vital to the trip for each day (on the right bar).
7. User will be allowed to type a blog-like post about each day and upload pictures to the album for the day. The pictures will all be visible in one cumulative album at the trip home page and also available when viewing the particular day.
8. After the trip is finished, users have the option to share the trip on their profile page.

past trip:

9. User is at the share trip home page (different from the previous). User enters the details of the basic itinerary of the trip - starting/end locations, start/end dates. The application will automatically update the total mileage, time to drive, and other vital statistics.
10. User will then have the **option** to have day to day maps to show where they traveled for each day.
11. User will be able to upload photos, post about the entire trip or each day, and then share it with friends.
12. User clicks in the menu that is always visible and saves the trip.

Exceptions:

1. User does not specify the requisite details of the trip - a trip name, start/end times and locations, and daily basic itinerary - start/end location and time, and planned route. The lack of these fields will be detected on client side so no server time is wasted.

Priority: Essential, must be implemented.

When available: First increment.

Frequency of use: Should be used a lot application. Core feature

Channel to actor: Via web browser.

Open issues:

1. Where on the web interface will all the functionality and buttons for each day of the day be displayed for good UX?

Use-case: ShareTrip

Primary actor: user

Goal in context: Allow the user to share the trip with friends.

Preconditions: User has already registered and logged in, and has at least one created trip that was saved.

Trigger: User clicks on the 'save' button at the bottom of any selected trip.

Scenario:

1. User logs in and clicks on one of his trips in his profile page main content area.
2. User makes some changes to the trip, or maybe creates a new trip and starts filling information.
3. User scrolls to the bottom of the page, and then clicks on the 'save' button.
4. User then selects what content he wants to share - blog posts for each day/entire trip, location, pictures, etc. User also selects with whom he wants to share the data with: everyone who uses the site, only friends, or only himself.
5. User can change the settings of the shared trip at anytime by clicking on the trip and clicking 'share' button again.

Exceptions: None

Priority: Non-essential for core, but important feature

When available: Second increment.

Frequency of use: Often enough to be make it useful.

Channel to actor: Via web browser.

Open Issues:

1. Should we makes saves periodical or not to save work?
2. What aspects of trips should never be shared, even to friends?

Use-case: ViewTrip

Primary actor: user

Goal in context: Allow the user to view the trips of friends.

Preconditions: User has already registered and logged in, and is at the profile page of a friend.

Trigger: User clicks on the link to the friend's profile page, and then clicks on the link to the friend's specific trip.

Scenario:

1. User logs in and clicks on one of his friends on the left hand bar.
2. User clicks on one of the trip's link on the friend profile page.
3. User as a friend can see all the information about the trip that the friend decided to share.

Exceptions:

1. User attempts to view a trip that he has no permission to view - e.g., a trip of a non-friend, whose profile page he nonetheless can see. In this case, we just show a custom (non-JS) popup that will alert the user of invalid permissions.

Priority: essential

When available: First increment.

Frequency of use: Most often used feature

Channel to actor: Via web browser.

Open Issues:

1. What parts of a trip are never viewable by someone else, even if they are friends?

Use-case: SearchBar

Primary actor: user

Goal in context: Allow the user to search for trips, users, etc.

Preconditions: User has already registered and logged in.

Trigger: User types in the search bar at the top and clicks search.

Scenario:

1. User logs in and types query in the search bar.
2. AJAX filter for all relevant results.
3. Press search/enter produces results of trips and users that match the query.
4. Display the contents in the page. Each query will have a unique url so that they can be copy and pasted.

Exceptions:

1. User types a query that returns nothing. In that case, just output the literal 'There is nothing found'.

Priority: Non-essential

When available: Release

Frequency of use: Depends on how well we implement it.

Channel to actor: Via web browser.

Use-case: Destinations

Primary Actor: User, Non-User

Goal in Context: Allow the user to browse trips on the site by location.

Preconditions: User has signed in and can view all trips of friends(unless private), Non-user can preview trips but must register before being allowed to view trips in detail.

Trigger: User selects destinations tab at the top of the site.

Scenario:

1. User Logs in to website(if registered)
2. User Selects destinations tab at the top
3. Destinations tab pulls up(using database) all locations that include trips by country(Some countries will have "hot" cities, which are very popular worldwide for more direct access)
4. User browses to trip of desire and opens trip page.

Priority: Non-Essential

When available: Shortly after release, need to allow at least a small user base before implementing.

Frequency of Use: Frequently enough, those that are browsing and would like to see what we have to offer would make great use of this feature.

Channel to Actor: Via Web Browser.