Project Backlog - Team 7

7th, September, 2018

Team members: Harika Lingareddy, Simona Virga, Sean Becker, Alexis Williams, Shengqi Wang, Pooja Tewari
Project Coordinator: Yi Sun

Problem Statement:

Planning vacations can be difficult; booking hotels, finding transportation, and discovering landmarks can be challenging since it usually involves jumping from one specialized travel website to another looking for deals. Planning gets even more complicated when scheduling a trip with multiple destinations with multiple people. Our team plans to develop a web app that simplifies the process of planning and booking a vacation.

Background Information:

Audience:

People across the globe plan vacations throughout the year using various online platforms and services. It's surprising that despite having multiple platforms to book hotels, reserve cars, or find attractions, there are not very many options that allow all three related aspects of a vacation to be organized at one place.

Similar Platforms:

There are several travel management services such as Kayak.com, Hotwire.com, Trivago..com. There services allow users to plan different aspects of vacations such as flights, hotels, and cars, but they all work differently. Kayak.com and hotwire.com allow flight, hotel, and car booking. Trivago.com only allows hotel price comparisons.

Limitations:

Even though these services are all useful, there is no one place to really plan out hotels, transportation, and nearby tourist attractions. Our team wants to create a platform where users can have the convenience to plan everything all together. We also plan to make a user friendly UI which will help cut down the time for planning. We would like to have a platform that is simple and to the point in terms of what can be done, but offers everything a vacation planner would need to plan out their dream vacation. Based off of this, we would like to implement a solution where a user can plan nearly everything related to a vacationing in one place, whether is is booking a hotel, reserving transportation, or finding places to see.

Environment:

As part of our development environment, our web app will be made with HTML/CSS/Bootstrap and Javascript. Our backend will be made of Node.js, Express, etc. For our database we plan to either MongoDB, Firebase, or MySQL and a database management system to store our user information. Our team will use Jenkins to automate testing frameworks.

Functional Requirements:

Backlog ID	Functional Requirements	Hours	Status
1	As a user, I would like to be able to create an account using my Google credentials.	5	Completed in Sprint 1
2	As a user, I would like to be able to login and manage my account using my Google credentials.	4	Completed in Sprint 1
3	As a user, I would like to view my trip history.	14	Sprint 2
4	As a user, I would like to be able to filter through my past trip history.	6	Sprint 2
5	As a user, I would like to favorite previous trips.	3	Sprint 2
6	As a user, I would like to pin locations on a map in order to specify each location I would like to go.	14	Completed in Sprint 1
7	As a user, I would like to specify coordinates of each location I would like to go.	8	Completed in Sprint 1
8	As a user, I would like to search by name to specify each location I would like to go.	5	Completed in Sprint 1
9	As a user, I would like to have the ability to choose how long to stay at each location.	4	Completed in Sprint 1
10	As a user, I would like to choose my preferred mode of transportation.	2	Completed in Sprint 1
11	As a user, I would like to see nearby tourist attractions.	5	Sprint 2
12	As a user, I would like to specify how many people I am traveling with.	2	Sprint 2
13	As a user, I would like to choose to have a time efficient or cost efficient trip.	20	Completed in Sprint 1
14	As a user, I would like to see the price of my trip categorized into their individual components.	10	Completed in Sprint 1
15	As a user, I would like to be able to share my trips with others.	6	Sprint 2
16	As a user, I would like to be able to name my trips.	2	Completed in Sprint 1

17	As a user, I would like to be able to reuse past trip information.	5	Sprint 2
18	As a user, I would like to be able to filter my searches.	6	Sprint 2
19	As a user, I would like to see trip ideas.	5	Sprint 2
20	As a user, I would like to be able to reroute my trip.	6	Sprint 2

Non-Functional Requirements:

- Security - Users should not be able to delete other user's travel plans, user identity should remain hidden, secure authentication
- Usability - Responsive to user input and make sure during calculations it does not look like the application has frozen, ability to run on most popular browsers (firefox and chrome), make user interface easy to understand and navigate, have user interface be visually appealing
- Scalability - Multiple users will be able to login to the website simultaneously and, access the features extended by the website without compromising security and performance.

Use Cases:

Case 1: Create an account using Google credentials	
Action: 1. Choose 'create account' option 3. Fill out form 5. Submit form	System Response: 2. Account creation form appears 4. Process form 6. Indicate success or failure to user 7. Form disappears
Case 2: Login and manage account using Google credentials	
Action: 1. Click 'login' button 3. Fill out form 4. Submit form	System Response: 2. Initiate login process 5. Redirect to root
Case 3: View trip history	
Action: 1. Choose 'view trip history' option	System Response: 2. Query user's trip history 3. Return trip history 4. Show history

Case 4: Search past trip history	
Action: 1. Choose ‘search past trip history’ option 3. Select search criteria	System Response: 2. Show search popup 4. Query by search criteria 5. Return results 6. Show results
Case 5: Favorite previous trips	
Action: 1. Select ‘Favorite option’	System Response: 2. Add flag to trip in database 3. Indicate that trip has been favorited
Case 6: Start planning a trip	
Action: 1. Click on begin 3. Select starting location on map or input coordinates 5. Submit starting location	System Response: 2. Redirect user to trip planner page 4. Find coordinates on map 6. Indicate that said location has been selected
Case 7: Save a trip	
Action: 1. Click on save	System Response: 2. Save all trip information at current state to the user
Case 8: Cancel a trip	
Action: 1. Click on cancel 3. Confirm to cancel	System Response: 2. Prompt user for confirmation 4. Clear all trip information and redirect user to the homepage
Case 9: Pin locations on a map in order to specify each location I would like to go	
Action: 1. Click on location on a map	System Response: 2. Record location 3. Indicate that said location has been selected
Case 10: Specify coordinates of each location I would like to go	
Action: 1. Select option to specify coordinates	System Response: 4. Record coordinates

2. Select coordinates 3. Submit coordinates	5. Find coordinates on map 6. Indicate that coordinates have been selected
Case 11: Search by name to specify each location I would like to go.	
Action: 1. Click on search bar 2. Enter location name	System Response: 3. Search for places using Google API 4. Return results using Google API
Case 12: Choose how long to stay at each location	
Action: 1. Enter days staying at a location	System Response: 2. Record number of days staying 3. Update UI to show number of days staying
Case 13: Choose my preferred mode of transportation	
Action: 1. Select option to select preferred method of transportation 3. Select preferred method of transportation 4. Submit	System Response: 2. Dialog appears 5. Record response 6. Indicate that response has been recorded 7. Dialog disappears
Case 14: See nearby tourist attractions	
Action: 1. Goes to new tab	System Response: 2. New tab with map and tourist attractions appears
Case 15: Specify how many people I am traveling with	
Action: 1. Select option to specify number of people 3. Select the number of people 4. Submit	System Response: 2. Selection dialog appears 5. Record number 6. Dialog disappears
Case 16: Choose between cost efficient or shortest time for transportation	
Action: 1. Choose transportation type 3. Select listing by price or time	System Response: 2. Dialog appears 5. Record response

4. Submit	6. Dialog disappears
Case 17: Choose between cost efficient or nearest distance for hotels	
Action: 1. Select listing by price or distance from destination 3. Select hotel 5. Submit	System Response: 2. Show list of hotels based on user's choice 4. Record response 6. Proceed to nearby attractions
Case 18: Choose between cost efficient or nearest distance for nearby attractions	
Action: 1. Select listing by price within a region or distance from destination location 3. Select attraction(s) 5. Submit	System Response: 2. Show list of nearby attraction based on user's choice 4. Record response 5. Close dialog
Case 19: See the price of my trip categorized into their individual components	
Action: 1. Select option to categorize trip result 2. Pick option	System Response: 3. Categorize trip 4. Return results
Case 20: Share my trips with others	
Action: 1. Select option to share trip	System Response: 2. Find other users 3. Show trip option to other users 4. Indicate that trip has been shared
Case 21: Name my trips	
Action: 1. Click on add name field 3. Write name of trip	System Response: 2. Name field is now typeable 4. Record trip name 5. Display trip name
Case 22: Reuse past trip information	
Action: 1. Select past trip	System Response: 2. Query information from user's selected past trip 3. Return information

	4. Show dialog with auto-filled information
Case 23: Filter my searches	
Action: 1. Select option to filter search 3. Select options to filter search	System Response: 2. Dialog appears 4. Record filtered options 5. Search based on filtered options 6. Display search based on filters
Case 24: See trip ideas	
Action: 1. Select option to see trip ideas	System Response: 2. Query ideas database 3. Return results 4. Show dialog of results
Case 25: Reroute my trip	
Action: 1. Select option to ‘reroute trip’ 3. Reroute trip start/end as desired 4. Confirm trip route	System Response: 2. Dialog appears 5. Record information 6. Return result