Trip Diary

Software Engineering (14:332:452)

Report 1

Team 5

Samuel Zahner
Kinjal Patel
Vincent Chan
Nisha Bhat
Gaurav Sethi
Samuel Minkin
Yash Shah
Jonathan Banks

Submission Date

February 23, 2020

Table of Contents

Contribution Breakdown

		Sam Z	Kinjal	Vincent	Nisha	Gaurav	Sam M	Yash	Jon B
Project Manag (9 points)		10%	75%			5%	10%		
Section 1 (10 points)	Prob Statement (6 points)		33%	20%	14%		33%		
	Glossary (4 points)					34%			66%
Section 2 (6 points)	Functional Req (2 points)	30%			30%	20%		20%	
	Non-Funct Req (2 points)	45%		15%			40%		
	UI Req (2 points)		33%	14%		33%		20%	
Section 3 (30 points)	Stakeholders (2 points)						10%		90%
	Actor and Goal (3 points)				33%		34%		33%
	UC Casual (3 points)	33%			33%	34%			
	UC Diagram (5 points)	50%			20%	10%		20%	
	Trace Matrix (5 points)	34%			33%	33%			
	UC Fully (6 points)	33%			33%	5%	29%		
	Sequence Diag (6 points)			40%			30%		30%
Section 4 (15 points)	Prelim design (7.5 points)		33%	33%			12%	22%	
	User Effort (7.5 points)		33%	33%			12%	22%	
Section 5 (25 points)	Concepts (6 points)		80%	20%					
	Diagram (5 points)		75%			25%			

	Association (4 points)		50%	25%		25%			
	Attributes (3 points)	50%		50%					
	Trace Matrix (3 points)					100%			
	Contracts (4 points)	33%			33%			34%	
Section 6 (2.5 points)	Size Estimation			10%		10%	25%	25%	30%
Section 7 (2.5 points)	Plan of Work						50%	50%	

Responsibility Allocation

Sam Z: 2 * .3 + 2* .45 + 3*.33 + 5*.5 + 5*.34 + 6*.33 + 3*.5 + 4*.33 + 9*.10 = **12.39 points**

Kinjal: 9*.75 + 6*.33 + 2*.33 + 7.5*.33 + 7.5*.33 + 6*.8 + 5*.75 + 4*.5 = **24.89 points**

Vincent: 6*.2 + 2*.15 + 2*.14 + 6*.4 + 7.5*.33 + 7.5*.33 + 6*.2 + 4*.25 + 3*.5 + 2.5*.1 = **13.08 points**

Nisha: 6*.14 + 2*.3 + 3*.33 + 3*.33 + 5*.2 + 5*.33 + 6*.33 + 4*.33 = **9.37 points**

Gaurav: 9*.05 + 4*.34 + 2*.2 + 2*.33 + 3*.34 + 5*.1 + 5*.33 + 6*.05 + 5*.25 + 4*.25 + 3 + 2.5*.1 = **11.84 points**

Sam M: 9*.1 + 6*.38 + 2*.4 + 2*.1 + 3*.34 + 6*.24 + 6*.3 + 7.5 *.12 + 7.5*.12 + 2.5*.25 + 2.5* .5 = **12.12 points**

Yash: 2*.2 + 2*.2 + 5*.2 + 7.5*.22 + 7.5*.22 + 4*.34 + 2.5*.25 + 2.5*.5 = **8.335 points**

Jon B: 4*.66 + 2*.9 + 3*.33 + 6*.3 + 2.5* .33 = **8.055 points**

1. Customer Problem Statement

A. Problem Statement

1. Actor - General Traveller

As a traveller, I understand that there are many items that compromise going on a trip, these include planning in advance, doing what was planned, and then reflecting on the journey. It is easy to spread your travels on several platforms -- you are responsible for booking flights on an airline's website, you plan on a trip planning platform such as TripAdvisor or a local application, and you store your photos in an album.

A. Concern 1: Trip Summary

I have never had the opportunity to view everything in one place, it's increasingly frustrating understanding expenses from mismanaged credit card statements, knowing what plans you've made each day, and my reflections upon each day (since I often forget to journal, which is something essential when understanding how I experienced each day).

What I would find ideal is a platform where I was able to see everything together. Everything includes what I did each day (including the transportation I took), all of my journals (which reflect on what I did), and what I spent on different excursions. This would allow me to be able to reflect on my trip without having to scour different platforms, and essentially places everything related to my vacations together

Trip Diary Solution: Trip Diary allows for an environment which aggregates all information related to a vacation. It provides a calendar in which people are able to view the dates of their trip. On that same page, they are able to view a trip summary, which includes locations they've visited (through a map visual), their budget (with interactive charts to organize information), pictures, and journal entries. To further organize this, if the 'general traveller' wants to see a further breakdown, they are able to interact with the calendar. If they choose a specific date of their trip, they can view the day's summary, which includes the same items from the trip summary (but only for the entries on that date).

2. Actor - Journaller

Whenever I travel, I love to write about my reflections from each day. According to this travel blog which references the benefits of keeping a travel journal, I essentially identify with being able to

1. Gathering information for different needs and wants during the trip
2. Documenting thoughts provides a meditative therapy through the hassle of travelling
3. Provides a creative outlet to my texts, drawings and non-text scribbles
4. An alternative when passing time during a dead period on a trip (ex. Sitting at the airport, waiting for a train, etc.)

5. It is a personal souvenir of my trip, with several sentimental entries reflecting on my thoughts from my time away
6. Provides for richer memories, and an opportunity to look back at how I experienced each day

A. Concern 1: Digital Platform

Although journalling is a great pursuit in a physical book, it’s quite frustrating having to find a place for my diary, especially when I have limited room in my backpack while on excursions. Also, there have been several times where I have lost my book, which absolutely demotivates me to continue journaling for the remainder of my travels. Additionally, sometimes I am so tired, and physically writing is the last thing I want to do. I wish there was a means to type my thoughts out on something, instead of always having to physically write.

What I want to see is a digital platform, that is accessible off of my phone or tablet. As for several people, my life is held in my phone or digital device. It would be increasingly helpful if I could journal within an app that was related to my travels. Additionally, it would be great if I had several options -- doodling, typing, and adding other emoticons.

Trip Diary Solution: Trip Diary offers a journaling feature, where the ‘journaller’ is able to document his/her experiences on a digital platform. This feature is not solely a text sheet, but provides prompts of the user to answer (“What was your favorite memory from today?”, etc.), allows for typing and doodling, photo, and sticker features. This provides the user with an array of options when documenting their thoughts.

B. Concern 2: Adding Pictures to Entries

Another concern I have is wanting to put pictures in my journal. It’s great being able to print pictures and placing them in my journal, but that has to be done after the trip. Sometimes, I completely forget or do not have time to go and print pictures.

What I would like to see is a place where I can put pictures directly in my journal. Essentially, I would only be able to solve this concern by having a digital platform, where I can instantly take/place a picture and place it within my text.

Trip Diary Solution: Trip Diary provides an option to add photos to accompany text. Essentially, users will be able to either choose from their camera roll or physically take a picture while adding entries. This will allow them to connect their photos (which are additionally incorporated into the application and journal entries, providing for interaction between features.

C. Concern 3: Remembering to Journal

As any person has, I often forget to do non-essential tasks, especially journaling during my trip. It is frustrating having journal entries for only select days of the trip, and I always wish I had a full collection of entries when reflecting back on my trip.

What I would like to see is a reminder feature for my journals. This would remind me to journal at a (possibly preselected) time directly to my phone.

Trip Diary Solution: Trip Diary allows users to receive reminders to journal if they have not completed it already for that date. This would be directly on their home screen, and if they clicked on the notification, it would take them to the journal entry page on their Trip Diary application.

3. Actor - Photographer

As a photographer, I would like to store all of the wonderful memories that I capture on my trips. Photos allow me to not just see a moment of my trip but also serve as mental cues which instigate entire memories that I have associated with the photo. In most trips, I have experienced pieces of culture, history, and life which inspire, amaze, and transcend the life that i'm used to. When I look back at my trips I want to be able to feel those same wonders so that I can truly understand the importance of traveling and learning about the world.

A. Concern 1: Photo Organization

There is an issue of how to best organize photos so that the user can get the most out of viewing them. There exist social media sites such as Facebook which allow one to store photos in the order that they are uploaded in an album. These photos typically have metadata such as date, location, and descriptions. However, when one wants to view an entire trip, viewing an album may not be the optimal way of re-experiencing a trip. As a whole, it is hard to tell how the pictures fit in together and one may struggle to understand the meaning of some pictures whose purpose may be hard to discern or are not well described.

Trip Diary Solution: I believe that users have an unmet desire to be able to organize their trips into categories and be able to write descriptions of individual photos and groups of photos. One of the major purposes of the trip diary is to encourage users to better document and manage their trips, and by doing so to their photos they will have a better and fuller understanding of a trip.

4. Actor - Mapper

A map is a visual tool which can help trace the way one traveled. Looking at a map gives me a better idea of what regions I visited, differences between those regions, and also specific points within a region. I find it interesting to be able to see how places are organized within a particular location, it provides a story about what life is like in a city. For instance, if I were to visit a park, a broadway show, and a skyscraper I would like to know how these parts are located relative to each other so that I can get a better idea of the city's structure.

A. Concern 1: Trip Ordering

An important aspect which photos may fail to convey is a specific ordering of how a day was spent as photos are taken sporadically or may not be taken at all at particular locations. Hence, there should be a way for users to understand how they were able to get from one place to another.

Trip Diary Solution: So, I believe there should be a mapping feature which uses geolocation to trace location. Doing so, one can curate a summary of places visited, restaurants that were stopped at, and roads traveled. This feature aggregates a lot of information for me and is convenient in not having to be bogged down in writing the same detailed information. Additionally, one can fetch information about the places visited that I may have not known, enlightening me about every aspect of my trip. This feature is meant to provide as much convenience as possible so that the trip diary acts as an enhancement and not as a distraction.

B. Concern 2: Privacy

However, with this convenience comes a tradeoff in privacy. Many users, including myself, may be uncomfortable with having their location being tracked at all times. I especially would not want data about the trips I've taken to be shared with third-party sources. As this is a diary, I expect a high level of privacy. The world is a complicated place and I may choose to travel to an area which may be experiencing a conflict with other regions. I would feel distressed if that information was shared without my permission and ended up offending someone, and I would much rather it be kept private.

Trip Diary Solution: I believe that the application should have the geolocation tracking in the background be optional. This will not severely hinder the mapping feature as one can employ additional ways of getting locations. For instance, one can use APIs to extract the location of a picture taken and based on that form some kind of ordering. I hope that the application will seek to balance privacy and a detailed mapping as best as possible. It should be obvious for the user to see if location is currently being tracked and to turn it off.

5. Actor - Budgeter

As a single traveler, I am solely responsible for how I spend my money. A goal of mine when I travel is trying to maximize my experiences, but limit how much money I spend. For me, it is difficult to keep track of my expenses, since there are many expenses to keep track of. There are multiple forms of payment that I use (different currencies, card, online payments) as well as many different activities and expenses (transportation, food, traveling, souvenirs). Trying to keep track of receipts, bank statements, and payments is a frustrating element that takes away from the experience of traveling

A. Concern 1: Budgeting

What I would like to see from an app is an easy way to log and organize my expenses during a trip, so that I can forget about keeping track of payments, and enjoy my travel experience. I want a feature where I can categorize my payment plans, and send notifications when I'm nearing my spending limit.

Trip Diary Solution: The trip diary has a budget feature where a user will be sent notifications to log their expenses and warn them where they are nearing their budget. A user will be able to label different categories that they define or from a preset list from the application, and the user will be able to assign smaller budgets to each category. For each category the user will be able to see a running list of their expenses, and a total expenses list will also be available to the user. A visual graph will also be displayed, so that the user has a visual understanding of how well they are following their financial goals. Once a budget has been set, it can be edited as well.

6. Actor - Parent with Children (Finding Attractions)

As a parent, I need to be able to find attractions that will be fun for the whole family, and document the places I visited, so that we can remember the experiences we had together. Trying to parse through several travel guides to see what exciting activities we can do as a family is a frustrating experience, since each guide has different information.

A. Concern 1: Finding and Saving Different Attractions

I want to be able to view different attractions, both excursions and locations that I have visited so that I can share this data with other family members and friends. I want to be able to not only view these locations in a list format, but also in a visually appealing way so I can understand how much distance I have traveled to visit these locations. I want to be able to view all the destinations I have visited so that I can pick my favorite ones and I see them in an organized format. I also want to be able to see my attractions so that I can see I am making the most of our trip and visiting as many important locations as possible. I also want to be able to view attractions by category, whether that be food, excursions, etc. This would help me note all the different types of attractions I have visited while on my trip.

Trip Diary Solution: The trip diary has a map feature where a user can view all the attractions and locations they have visited within their trip. When a user uploads a photo or video, the diary can suggest nearby locations that the destination might be or the user can manually input the attraction. Then, the user can have that location pinned on the map so that they can view how much they have traveled on their trip and how many spots they have visited. This gives the user an idea of what they have done on their day-to-day in regards to visited attractions. These pins have tags, whether that be food, excursions, hidden gems, etc. This provides organization for all the attractions visited so that the user can view all the food spots they hit, all the tourist sites they’ve visited, all the hidden gems they have discovered, etc.

B. Glossary of Terms

Term	Definition
Trip Summary	Overview of a completed trip or a trip in progress. One may view the trip budget, the photos of the trip, the journal entries, the timeline maps, etc.
Journal Entries	A feature that allows users to describe their trip experience with a creative and easy to use and navigate questionnaire. This feature saves the responses for future reference.
Trip	A well-defined data model used in our application to store and analyse information about the user's trips. This feature stores data such as location, photos, journal entries, attractions visited, hotel stays, flights, etc.
Daily Summary	A feature of the application that allows the user to view a concise analysis of a specified day of the user's current trip. The daily summary breaks down the users spending, location, photos taken and more under one page. In addition, the daily summary will display other relevant information to the user such as the suggested funds left to spend during the current day on the trip. The daily summary is designed to be a convenient, easy to use and aesthetically pleasing overview of a single day.
Attractions	The places that the user visits during their trip that does not include transportation or shelter. An example of an attraction would be a restaurant or a concert. Attractions could also be referring to the predicted places that the user could visit in the future. - Note: Events fall under the category of attractions.
Calendar	A feature that allows users to view major components of a trip, such as hotel check-ins, flights and planned attractions on an easy-to-follow day-to-day calendar grid.
Trip Budget	A feature that enables the user to enter the amount of money they would like to spend on the trip in total as well as each component of the trip. The Trip Budget feature will store this data and will eventually analyse this information to build a financial plan for the user.
Toolbar	A part of the application that allows the user to quickly access a different feature of the application. This toolbar is an access bar that will lie at the bottom of the application on most pages.
Photos	Collection of photos taken by the user during the current trip.
Geotag	Location data associated with an image taken by a camera.

Map	View within the application which allows the user to view their activity throughout their trip
Activity	User’s location history throughout the trip duration.
Pin	Point on map where the user visited an attraction or took a photo.
Transportation	Aggregated travel data for the current trip; includes varying modes of travel such as flights, ubers, trains, etc.
Save	Saves an attraction that is appealing to the user, and places it in a list, so that the user can explore attractions later for future consideration.
Commit	Adds an attraction to an itinerary and will send push notifications to remind the user to go to these events.
Budget	Allocating money to different categories (either defined by the user, or any number of the preset categories in the application) to set a spending cap for the user.

2. System Requirements

A. Functional Requirements

Identifier	User Story	Size
REQ - 1	As a user I will be allowed to create an account and login so I can use the application, so I can get access to my trip journal.	3
REQ - 2	As a user, I will be able to view all of the trips I created so I choose to view, edit, or create new trips accordingly.	3
REQ - 3	As a user, I will be able to click on any date of my trip from the calendar	2
REQ - 4	As a user, I will be able to view all my photos, journal entries, and events in the trip summary	3
REQ - 5	As a user, I will be able to view all my photos, journal entries, and events on a single day in the daily summary	3
REQ - 6	As a user, I will be able to view suggestions of specific attractions or events for the current or future day	3
REQ - 7	As a user, I will be able to choose to add photos, videos, journal entries, and events to the current day	4
REQ - 8	As a user, I will receive a notification if an event I have planned is happening soon	3
REQ - 9	As a user, when creating a journal entry, I will be able to add pictures, drawings, and videos to my writing	3
REQ - 10	As a user, when creating a journal entry, I will be presented with multiple prompts to write on	2
REQ - 11	As a user, I will be able to add photos to my trip so I can organize them into being part of my trip diary.	3
REQ - 12	As a user, I will be able to view the photos I took for the trip so I can get a closer look at the pictures I took.	2
REQ - 13	As a user I will be able to have my location tracked so I can view where exactly I went throughout the day during my trip.	2
REQ - 14	As a user, I will be able to sort the photos I took by the location so I can easily organize and view photos how I please.	2

REQ - 15	As a user, I will be able to view my location data on a map for the trip so I can look back at exactly where I went every day of the trip without writing it down.	5
REQ - 16	As a user, I will be able to view pins on a map where I took my photos and visited so photos can be viewed by locations on the map adding another dimension to sharing and organizing photos/ marking down attraction.	2
REQ - 17	As a user, I will be able to set pins on a map where I visited attractions so exact details of the trip can be stored to share with others.	2
REQ - 18	As a user, I will be able to view the mode of transportation I took to get from place to place so exact details of the trip can be stored to share with others.	2
REQ - 19	As a user, I will be able to sort my location data by date so looking into details of past trips is easy (can be exceptionally helpful for business expenses).	1
REQ - 20	As a user, I will be allowed to set a predetermined amount of money, so that I can establish a budget.	1
REQ - 21	As a user I will be able to input money spent on purchases, so that I can keep track of how much money I am spending relative to my budget.	4
REQ - 22	As a user, I will receive notifications to input money spent on a trip, so I can be reminded about my budget.	1
REQ - 23	As a user I will be allowed to categorize purchases, so that I can better organize my budget.	1
REQ - 24	As a user, I will be allowed to assign a budget for each category, so I can allocate funds to different activities.	3
REQ - 25	As a user, I will be able to view a running list of expenses for each category and a total expense list, so I can see what specifically I spent my money on.	2
REQ - 26	As a user I will be sent notifications when I’m nearing my spending limit, to warn me about my spending limitations.	2
REQ - 27	As a user, when I click on today’s date or future dates on the main calendar, I will see top attractions in the area, so that I can pick activities to do.	1
REQ - 28	As a user I will be able to filter activities by price, so that I can choose activities that fit my budget.	1
REQ - 29	As a user I will be allowed to filter activities by category, to choose which activities I want to pursue.	1
REQ - 30	As a user, I will be allowed to save future activities that I am interested in, so I can decide later whether I want to commit to the activities or not.	2

REQ - 31	As a user I will be allowed to commit to activities, and I will receive notifications before, so that I am reminded when my activities start	2
REQ - 32	As a user I will be allowed to delete saved activities that I am no longer interested in, so that my saved attractions view is not cluttered.	1
REQ - 33	As a user I will be allowed to delete budgets, so that my budget view is not cluttered.	1
REQ - 68	As a user, I will be able to delete any journal entry	1

B. Non-Functional Requirements

Identifier	User Story	Size
REQ - 32	As a user I will be able to access the application on my android device so that I can use the device while I'm traveling.	4
REQ - 33	As a user, my trips, daily summaries, journal entries, and photos will be kept private unless I choose to share them publicly	4
REQ - 34	As a user, I will wait no more than two seconds for the daily summary to render, no matter how many users may be active	3
REQ - 35	As a user, I will be able to utilize the calendar and journaling features on a mobile device	2
REQ - 36	The mobile application will be compatible with android phones	2
REQ - 37	The application will have an 80% reliability rating (the percent chance it doesn't crash) in a month	2
REQ - 38	As a user, I will be able to use the application without Internet, but cannot post anything or receive suggestions	4
REQ - 39	As a user, I will be able to view my photos, maps, and transportation on my phone and my computer so that I can have the convenience of using different devices.	4
REQ - 40	As a user, I will have the option to choose whether my phone is currently tracking my location so I can retain control over my privacy.	2
REQ - 41	As a user, I will be provided documentation that thoroughly describes how to use the photos, mapping, and transportation features.	1
REQ - 42	As a user, I will be assured that data being compiled by the application is	2

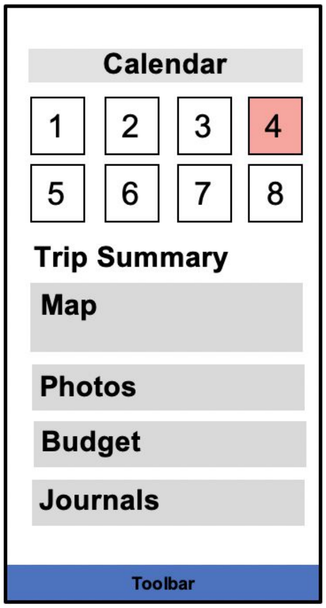
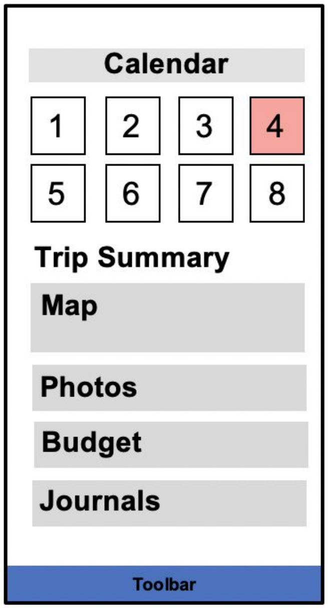
	accurate.	
REQ - 43	As a user, I will be able to upload hundreds of photos without running out of memory.	3
REQ - 44	As a user, I will be assured that my password to login is encrypted and my data is secure from other users as well as secure in the cloud.	2
REQ - 45	As a user, I will be able to easily navigate through all of my photos the first time I encounter the feature.	2
REQ - 46	As a user, I will be able to easily navigate through the photos feature because of its memorableness.	2
REQ - 47	As a user, I will be able to view my budget and saved attractions offline, so that the application is still usable without an internet connection.	1
REQ - 48	As a user my budget and chosen attractions will be private by default, unless I want to share them publicly with others.	1

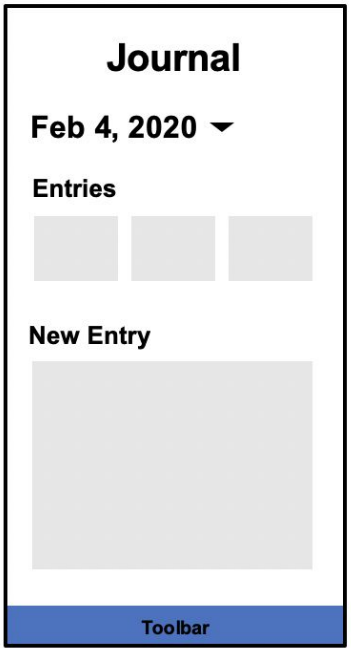
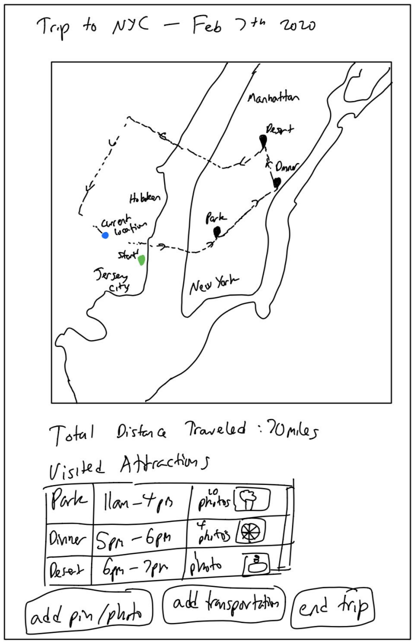
C. User Interface Requirements

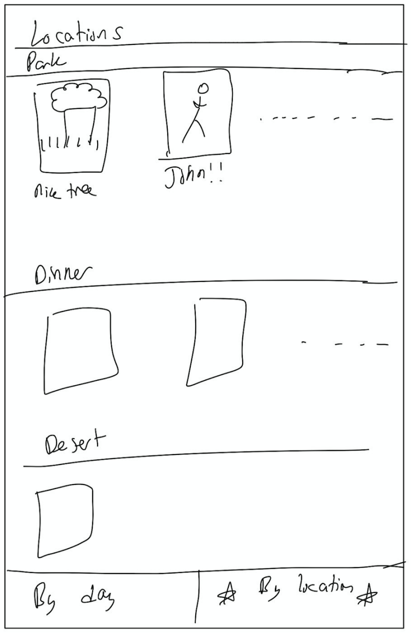
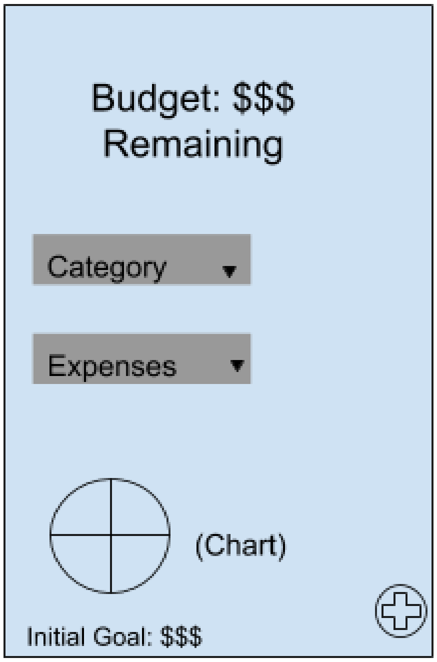
Identifier	**User Story**	**Size**
REQ - 49	As a user, I will have a log-in page to view my personal account	3
REQ - 50	As a user, I will have a page where I can select separate trips	2
REQ - 51	As a user, I will be able to view all of the dates in my trip from the calendar	2
REQ - 52	As a user, I will be able to view my trip summary (map, photos, journal, budget) in one location	3
REQ - 53	As a user, I will be able to click on a certain date from my trip, taking me to another page to access a summary of entries from that day (map, photos, journal, budget)	3
REQ - 54	As a user, I will be able to click on specific entries (map, photos, journal, budget) from the daily summary page to direct me to a full page of only that entry	4
REQ - 55	As a user, I will be able to go back to key locations within the app via a toolbar located at the bottom of the app	3
REQ - 56	As a user, I will be able to view my existing journal entries from a specific date	3

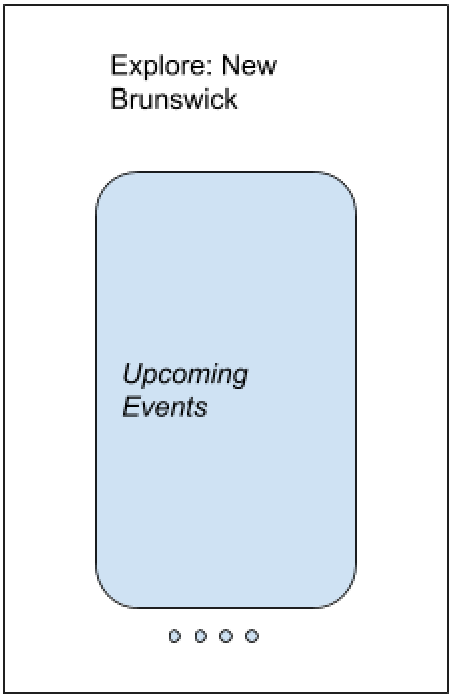
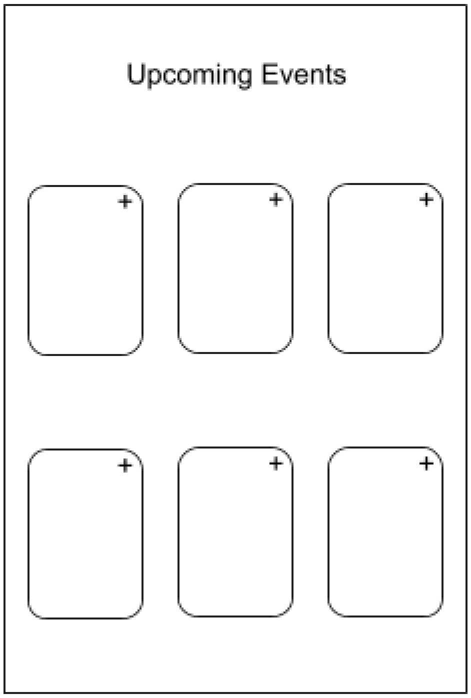
REQ - 57	As a user, I will be able to change the date from any page in the application	2
REQ - 58	As a user, I will be able to view the status of the current trip I am on which includes a map with all the pins I have set and the list of pins that can be clicked on to view/add relevant photos	3
REQ - 59	As a user, I will be able to view all the pictures of a trip organized by either the location they were taken or the day of the trip they were taken	2
REQ - 60	As a user, I will be able to view all information relevant to my budget, including, my pre-set budget, how much do I have remaining, and budgets that I allocated money to.	2
REQ - 61	As a user, I will be able to view a graph, depicting all of my budget information, so that I will be able to visualize my expenses.	4
REQ - 62	As a user, I will be able to click an edit button that will allow me to edit my budget information.	1
REQ - 63	As a user I will be able to click on future dates and see top attractions in an area separated by category.	1
REQ - 64	As a user, I will be presented with a list of several thumbnails, each representing a different attraction, so that the user interface will not be cluttered.	3
REQ - 65	As a user, I will be able to click on a button to save attractions for future consideration	2
REQ - 66	As a user, I will be able to click on a button to commit to attractions thus, sending push notifications as reminders.	3
REQ - 67	As a user, I will be able to toggle my list of expenses, so the interface will not be cluttered	1

(Cont'd) User Interface Pictures

Calendar	Daily Summary
 Calendar 1 2 3 4 5 6 7 8 Trip Summary Map Photos Budget Journals Toolbar	 Calendar 1 2 3 4 5 6 7 8 Trip Summary Map Photos Budget Journals Toolbar

Journal	Map
	 Trip to NYC — Feb 7th 2020 Manhattan Desert Dinner Hoboken current location Park Start Jersey City New York Total Distance Traveled : 70 miles Visited Attractions Park 11am – 4pm 10 photos Dinner 5pm – 6pm 4 photos Desert 6pm – 7pm photo add pin / photo add transportation end trip

Photos	Budget
 Locations Park Nice tree John!! Dinner Desert By day By location	 Budget: $$$ Remaining Category Expenses (Chart) Initial Goal: $$$

Suggestions (Pt.1)	Suggestions (Pt. 2)
	

3. Functional Requirement Specification

A. Stakeholders

1. Internal Stakeholders

a. **Developers**: This group of people works on developing the application in order to enhance traveling experiences

   i. Front-End

   ii. Back-End

   iii. Full-Stack

2. External Stakeholders

a. **Travellers:** These groups of people are interested in documenting and organizing a trip that they are currently on. Examples of these groups include (but are not limited to):

   i. **Group travellers:** Having this app will allow the members of a group to coordinate their decisions about their upcoming trip in one, easy-to-use web-app. Then, during the trip, recording trip information is beneficial for the future reference of any member in a group to recall what you and your friends did in a past trip.

   ii. **Institutional Travellers:** Whether on a business trip or a cool school field trip/vacation, this app is useful for the travellers themselves to view each and every trip detail so the traveller(s) can avoid confusion and miscommunication and, as a result, can remain on-schedule and on-task for the entire trip. As for the administrators/facilitators of the trip, the services that our app provides will be a great way to consolidate general records and financial records for the institution.

   iii. **Individual Travellers:** Documenting trips can boost popularity and fan-base for popular users and celebrities. But for the more common user, this app can be used to optimize the current trip (financially, quality of trip, etc.). In any case, it is also useful for anybody travelling solo to be able to reference older trips for any number of reasons.

   iv. **Couple or Family Travellers:** It is always a good feeling when memories and experiences are remembered from a family vacation. This app makes it easier to recall the experiences you had with your lover and children with our journal and photos feature.

b. **Visitors:** These groups of people are especially interested in looking through their own or other's previous trips for reference and inspiration for future plans. Examples of these groups include (but are not limited to):

   i. **Fans:** Fans would be interested in planning future trips based on their favorite celebrity's past plans. For instance, say a fan's favorite artist took a vacation in Paris, stood on top of the Eiffel Tower and took a picture of themself with a unique pose. That person may be interested in going to Paris and going to the Eiffel Tower and taking a picture of themselves with the same pose as their favorite celebrity. Our app makes it easy to do everything from budget this trip to uploading their own Eiffel Tower photo to our app and social media.

ii. **Family members:** Anyone that is part of a family might like to rediscover the trips that another family member took for any number of reasons. An example of this would be of a son looking to remember his lost father. Since the best memories are made on vacation, the son would find the most moving and sentimental memories while scrolling through his father's vacations in the diary and the photos feature on the father's profile.

B. Actors and Goals

Actors	Type	Role	Goals
Users	Initiating, Participating	The user creates, edits, or views features of the application which initiate requests to the system. The user participates in use cases where the system sends him or her a notification.	1. To log finances of attractions, food, excursions, etc. into the trip diary 2. To determine a budget for trip and view/update this budget 3. To locate attractions, food, excursions, etc. through search or suggestions 4. To save/add or delete locations of attractions, food, excursions, etc.
System	Initiating, Participating	The system initiates notifications to be sent to the users to ask about locations or remind them about documenting their day. The system participates in all kinds of requests sent to the service dealing with the application	1. To process requests from the client and send back responses. 2. To send notifications to the user to ask for participation or access to things like photos or locations
Database	Participating	The system receives requests from the service to retrieve, update, create, or destroy data.	1. Store user data 2. Analyze user data to provide interesting results such as mode, averages, ranges, etc.
Geolocation API	Participating	The system initiates a request to the geolocation API to retrieve data about the user's location	1. To determine the locations of nearby attractions and excursions so that the user can be suggested possible locations they have visited when they log in their diary

			2. To determine the location the user is currently at so that the user can use that information 3. To determine the exact location of any inputted GPS coordinates or names of locations so the user can view locations on a map 4. To save locations, their GPS coordinates and the name of that location on the map, so that users can save information on destinations 5. To use metadata from photographs and videos and geotag those GPS coordinates on a map so that the user can document locations of memories and trips
Bank	Participating	The bank provides details about the user’s bank account	1. To provide information on all spending so that user can select information to log into budgeting component of trip diary 2. To provide information on the total financial capacity of the user and by doing so, alerting the user if they are going to exceed the amount in their bank account

C. Use Cases

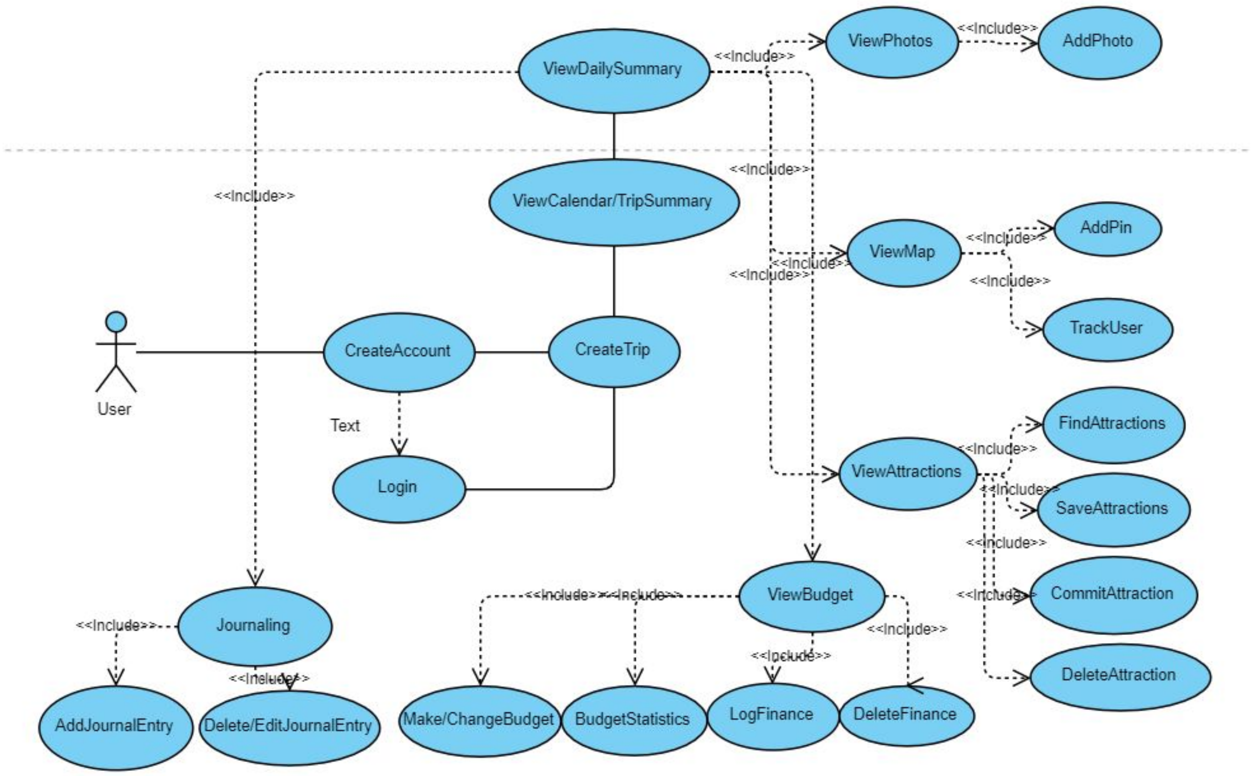
I. Casual Description

Use Case	Casual Description
UC-1: CreateAccount	REQ-1 Visitor can create a new account with a username and password
UC-2: Login	REQ-1 User can access their accounts and trip history by logging into the application
UC-3: CreateTrip	REQ-2 User can create new trip with starting date, ending date, and trip title
UC-4: ViewPhotos	REQ-4, REQ-12, REQ-14 User can view all photos taken on the trip either by day, or by the location they were taken
UC-5: AddPhoto	REQ-11 User can add a photo to a specific day/attraction for their trip
UC-6: ViewMap	REQ-4, REQ-15, REQ-16 User can view a map of the path they have taken with the attractions they have pinned
UC-7: AddPin	REQ-17 User can add an attraction as a pin with a title on the map to add specific photos to
UC-8: TrackUser	REQ-13, REQ-18 Geolocation API will be able to track the user's location and display the users path on a map in realtime and user can add mode of transportation as he pleases.
UC-9: ViewDailySummary	REQ-5, REQ-15, REQ-16, REQ-19 User will be able to view a page that has the daily summary of their trip from where they can edit details about their trip, add pins, add photos, etc
UC-10: FindAttractions	REQ-27, REQ-28, REQ-29, REQ-63 User can either search up specific attractions or can be suggested nearby attractions based on their current location, where this search can be filtered by category, price, and date.

UC-11: SaveAttraction	REQ-30, REQ-65 User can save a given attraction they have searched up or found
UC-12: CommitAttraction	REQ-31. REQ-66 User can commit to an attraction, allowing them to be reminded about that upcoming attraction on the date it is scheduled.
UC-13: DeleteAttraction	REQ-32 User can delete a saved attraction or delete a committed attraction
UC-14: ViewAttractions	REQ-47, REQ-48, REQ-64 User can view all the attractions they have saved and can search through them through organized categories for quick and easy access.
UC-15: LogFinance	REQ-21, REQ-67 User can log a finance by manual entry
UC-16: DeleteFinance	REQ-33, REQ-62, REQ-67 User can delete a logged finance
UC-17: BudgetStatistics	REQ-23, REQ-25, REQ-61 User can view budget statistics, through graphical and other visual forms to see where their money has be allocated and how much of their money is left in the budget
UC-18: ViewBudget	REQ-22, REQ-25, REQ-26, REQ-47, REQ-48, REQ-60 User can directly see how much money is left in their budget
UC-19: Make/ChangeBudget	REQ-20, REQ-24, REQ-62, REQ-67 User can either make a new budget or change the current budget they have
UC-20: ViewCalendar/TripSummary	REQ-51, REQ-52, REQ-53, REQ-55, REQ-56, REQ-57 Users will be able to set the dates of their trip and see them in a calendar as well as view their entire trip summary below their calendar within the app
UC-21: ViewDailySummary	REQ-3, REQ-5, REQ-6, REQ-7,REQ-15, REQ-16,REQ-19, REQ-33, REQ-34, REQ-35, REQ-38, REQ-53, REQ-54, REQ-56 Users click on a specific day of their trip to view their photos, journal entries, suggestions, or transportation used that day. They may also add any of these aspects to that day
UC-22: Journaling	REQ-4, REQ-5, REQ-53, REQ-54, REQ-56 Users can view their journal entries as well as the photos, drawings, videos that they added

UC-23: AddJournalEntry	REQ-7, REQ-9, REQ-10 Users will be able to add a new journal entry to any specific date. They can attach photos, drawings, or videos. Also, they will be presented with a selection prompt to write
UC-24: Delete/EditJournalEntry	REQ-7, REQ-9, REQ-68 Users will be able to delete or edit any journal entry that they have added

II. Use Case Diagram



Key:

Straight Line = Initiates

Dotted Line = Includes

III. Traceability Matrix

REQ	PW	UC 1	UC 2	UC 3	UC 4	UC 5	UC 6	UC 7	UC 8	UC 9
REQ1	5	X	X							
REQ2	4			X						
REQ3										
REQ4	3				X					
REQ5	3									X
REQ6										
REQ7										
REQ8										
REQ9										
REQ10										
REQ11	3					X				
REQ12	3				X					
REQ13	3								X	
REQ14	1				X					
REQ15	3						X			X
REQ16	2						X			X
REQ17	2							X		
REQ18	1								X	
REQ19	1									X
Max PW		5	5	4	3	3	3	2	3	3
Total PW		5	5	4	7	3	5	2	4	6

REQ	PW	UC10	UC11	UC12	UC13	UC14	UC15	UC16	UC17	UC18	UC19
REQ20	1										X
REQ21	4						X				
REQ22	1									X	
REQ23	1								X		
REQ24	3										X
REQ25	2								X	X	
REQ26	2									X	
REQ27	1	X									
REQ28	1	X									
REQ 29	1	X									
REQ30	2		X								
REQ31	2			X							
REQ32	1				X						
REQ33	1							X			
REQ47	1					X				X	
REQ48	1					X				X	
REQ60	2									X	
REQ61	4								X		
REQ62	1							X			X
REQ63	1	X									
REQ64	3					X					
REQ65	2		X								
REQ66	3			X							
REQ67	1						X	X			X
Max PW		1	2	3	1	3	4	1	4	2	3

Total PW		4	4	5	1	5	5	3	7	9	6

REQ	**PW**	**UC20**	**UC21**	**UC22**	**UC23**	**UC24**
REQ3	2		X			
REQ4	3			X		
REQ5	3		X	X		
REQ6	3		X			
REQ7	4		X		X	X
REQ9	3				X	X
REQ10	2				X	
REQ33	1		X			
REQ34	3		X			
REQ35	2		X			
REQ38	4		X			
REQ51	2	X				
REQ52	3	X				
REQ53	3	X	X	X		
REQ54	4		X	X		
REQ55	3	X				
REQ56	3	X	X	X		
REQ57	2	X				
REQ68	1					X
Max PW		3	4	4	4	4
Total PW		16	32	16	9	8

IV. Fully Dressed Description

UC-21: View Daily Summary

Related Requirements: REQ-3, REQ-5, REQ-6, REQ-7, REQ-33, REQ-34, REQ-35, REQ-38, REQ-53, REQ-54, REQ-56

Initiating Actor: User
Actor's Goal: View journal entries, photos, suggestions, and transportation taken on a specific day of their trip
Participating actors: System

Preconditions:

- The user can view the dates of the trip they specified in the calendar
- There is a database that can store the users information they decide to upload

Postconditions:

- The user can view all information about a specific day of their trip
- The user can add photos, suggestions, budget, transportation to that specific day

Flow of Events for Main Success Scenario:
→ 1. User clicks on a specific day of their trip, requesting to view all information stored on that day
← 2. The system processes the request and makes a database connection to retrieve the stored data
← 3. The database sends the requested information
← 4. The system generates a page displaying the users photos, suggestions, journal entries, and transportation from that day

Flow of Events for Extensions(Alternate Scenarios):
→ 1. The user clicks the "view" button on any of the aspects (photos, suggestions, journal entries, etc.)
← 2. The system directs the user to a separate page where only the specific item is shown
→ 3. The user clicks the "add" prompt in this new window
← 4. The system directs the user to a new page depending on what aspect they chose.

a. They are directed to adding a journal entry
b. They are linked to their trip photos
c. They are directed to the map
d. They can view their budget

→ 5. The user chooses to add one of these entities by clicking
← 6. The system saves this information by making a database connection

UC-5: Add Photo

Related Requirements: REQ-11

Initiating Actor: User
Actor's Goal: To add a photo from the camera roll to a specific day on a trip.
Participating Actors: System

Preconditions:

- The System has access to user's camera roll
- There exists a database in which the system can store the photo/path to photo

Postconditions:

- Successfully stored photo path in database
- User can access the photo in the diary that was selected

Flow of Events for Main Success Scenario:
→ 1. User requests to upload a photo from the camera roll into a specific diary date.
← 2. The system receives the request. It will make a database connection and add a new entry containing the path of the photo.
← 3. The system updates the album so that the user may see their new photo uploaded

Flow of Events for Extensions (Alternate Scenarios):
← 1. The system does not have permission to access the user's camera roll so it sends a notification requesting such access.

UC-6: View Map

Related Requirements: REQ-4, REQ-17

Initiating Actor: User
Actor's Goal: To view a map of pinpoints at locations the user has saved throughout the trip.
Participating Actors: System

Preconditions:

- Location data is safely stored within a database
- The location of pins placed by user are stored within database

Postconditions:

- Path of where exactly the user went that day is drawn out with a line on the map
- Pins the user placed are displayed on the map and can be clicked on for more info

Flow of Events for Main Success Scenario:
→ 1. User opens up a trip and can view trip summary page or clicks on specific day of a trip and can view the day's summary page, both of which include a map at the top
← 2. The system processes the request and makes a database connection to retrieve the of the stored route taken as well as the location pins
← 3. The system generates a page which has a map at the top with a dotted line to represent the route the person took, start and end pins, and location pins the user placed that can be clicked on for more details

Flow of Events for Extensions (Alternate Scenarios):
← 1. The system does not have permission to access the users location at all times so it sends a notification requesting such access
→ 2. The user uses a VPN to spoof their location and this results in a discontinuous map that looks glitchy when drawn out

UC-23: Add Journal Entry

Related Requirements: REQ-7, REQ-9, REQ-10

Initiating Actor: User
Actor's Goal: Add a new journal entry to a specific date that can accurately reflect their experience
Participating Actors: System

Preconditions:

- The user can view their daily summary within the calendar
- The system has access to the database
- The system has access to the user's photo album

Postconditions:

- The user is presented with a writing prompt
- The user can add photos, drawings, and videos
- The journal entry will be saved to that specific day

Flow of Events for Main Success Scenario:
→ 1. The user requests to add a journal entry
← 2. The system processes the requests and generates a new page to add a journal entry
← 3. The system presents the user with a randomly generated prompt
→ 4. The user can click to do the following to their journal entry:
   a. Add photos
   b. Add videos
   c. Add drawings
← 5. The system directs the user to
   a. A photo album
   b. A sketching window
→ 6. The user clicks to "save" their current journal entry
← 7. The system creates a database connection and saves the journal entry to that specific day

UC-17: Budget Statistics

Related Requirements: REQ-23, REQ-25, REQ-61
Initiating Actor: User
Actor's Goal: View statistics in the form of numbers, graphs, and charts to illustrate the different allocations of their budget and how they are doing managing their budget.
Participating Actors: System
Preconditions:

- The user has input budget information
- The system has saved purchase history/budget information in database

Postconditions:

- The user can view a graphical analysis of the distribution of budget purchases

Flow of Events for Main Success Scenario:

→ 1. The user clicks on “view budget statistics”
← 2. The system processes the request by generating all budget data for the user
← 3. The system develops a pie chart based on purchases that have allocated the most finances
← 4. The system develops a meter to describe how much of the current budget has been used
← 5. The system develops the top 3 purchase categories the user has spent most on
← 6. The system develops a statement based on the user’s budgeting (e.g. “You are getting close to your budget”, “Great job saving!”)
← 7. The system develops randomly generated saving tips to portray
← 8. The system displays all this information on the budget statistics page

UC-18: View Budget

Related Requirements: REQ-22, REQ-25, REQ-47, REQ-48, REQ-60

Initiating Actor: User
Actor’s Goal: View the current budget that they have
Participating Actors: System

Preconditions:

- The user has already input a budget

Postconditions:

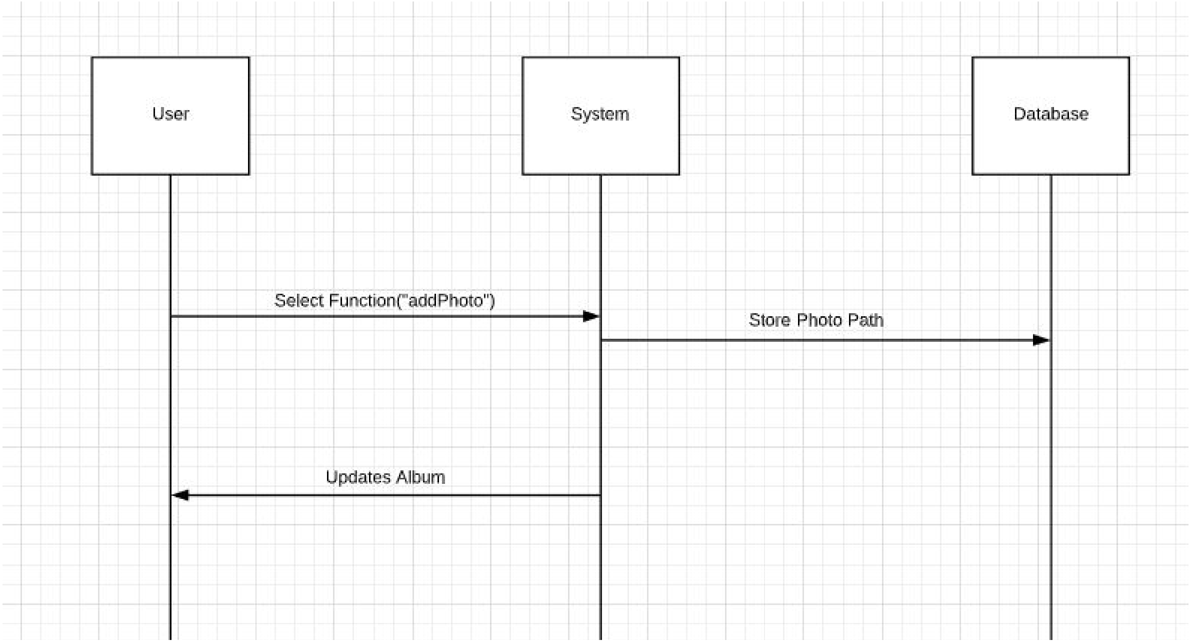
- The user can view the current budget they have
- The user can add a new finance or delete an old one
- The user can view budget statistics
- The user can change/make a new budget

Flow of Events for Main Success Scenario:
→ 1. The user clicks on “view budget”
← 2. The system processes the requests by generating the user’s budget in the database
← 3. The system displays the budget on the budget page
← 4. The system displays all the finances in a list
← 5. The system displays options to log a new finance, delete any of the current finances, change/make a new budget, and view budget statistics

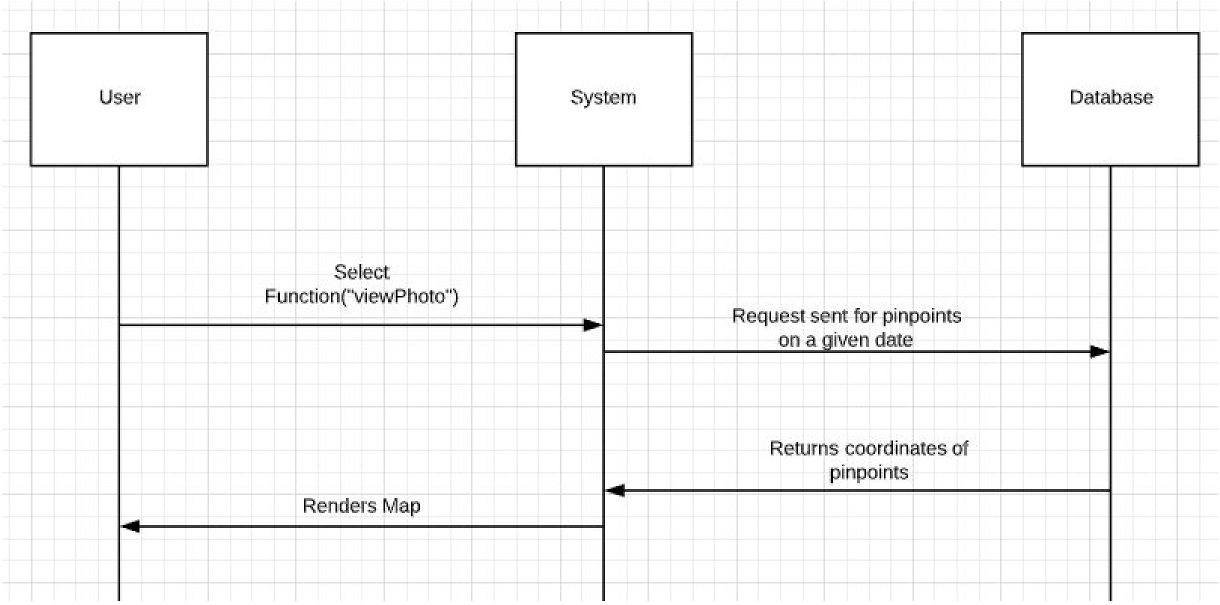
Flow of Events for Extensions (Alternate Scenarios):
→ 1. The user clicks on an attraction they wish to commit to
← 2. The system generates the user’s budget from the database and displays it
→ 3. The user commits to the attraction
← 4. The system subtracts from the total budget available and inputs it to the database

D. System Sequence Diagram

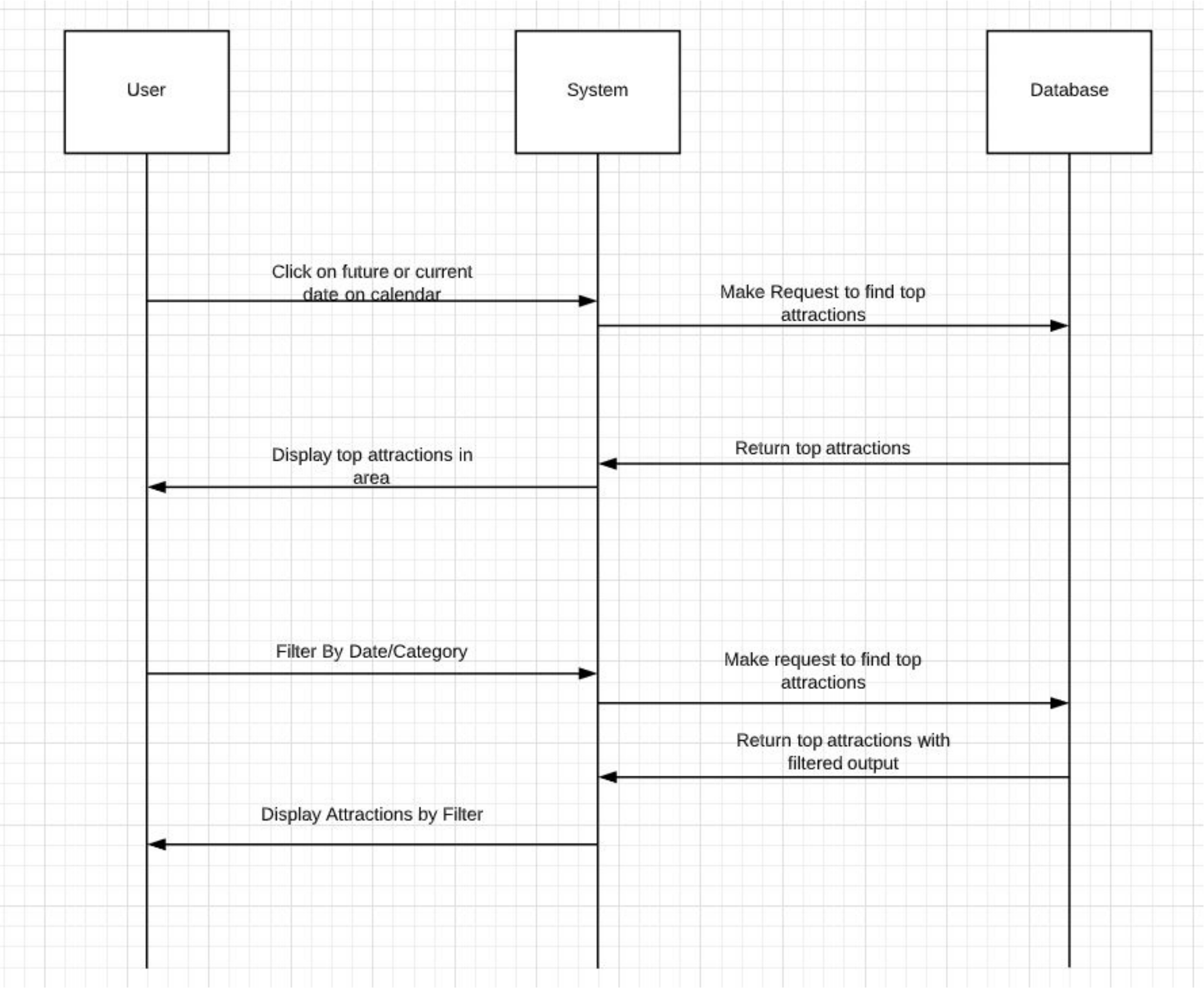
UC-5: AddPhoto



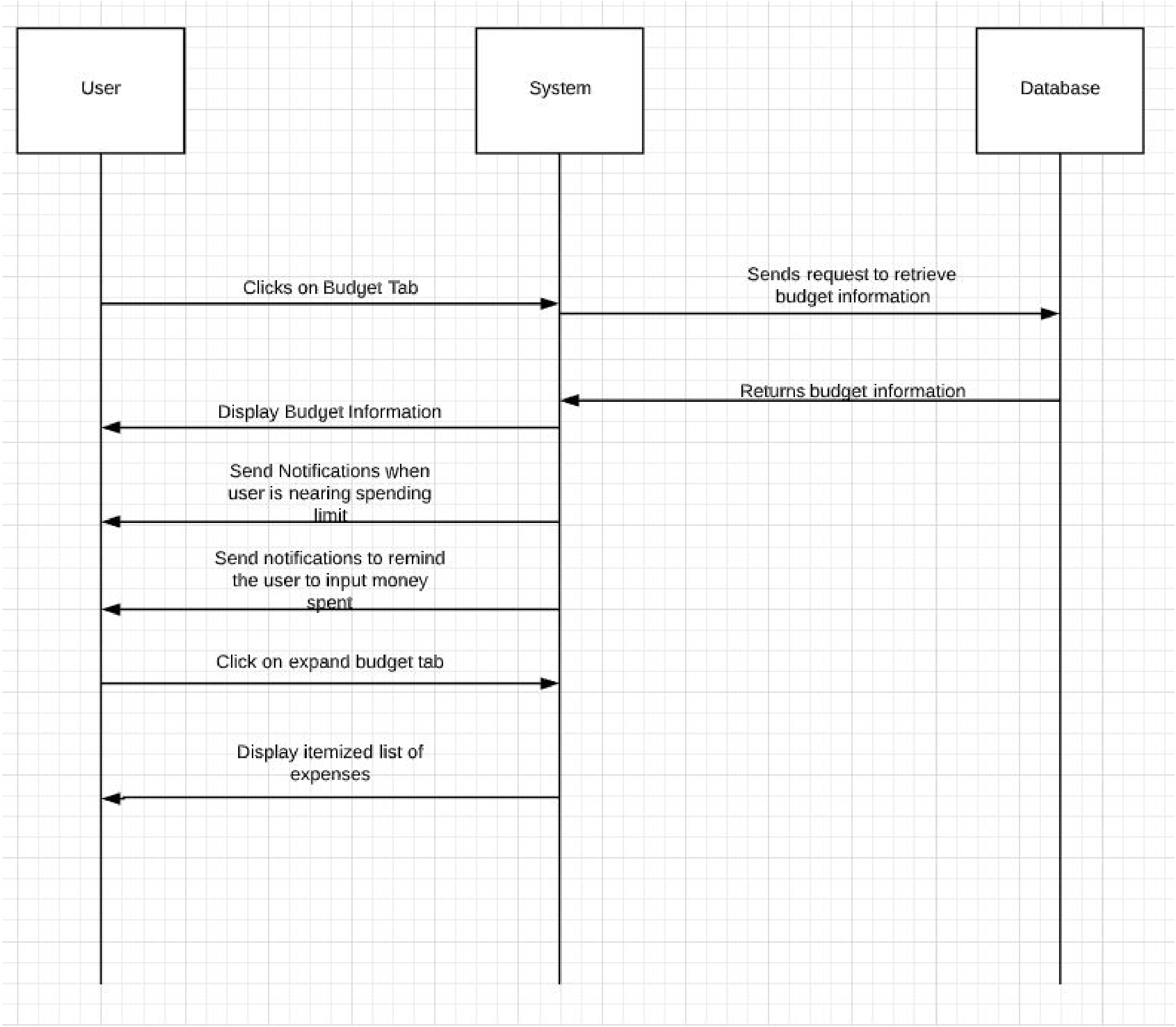
UC-6: ViewMap


User
System
Database
Select Function("viewPhoto")
Request sent for pinpoints on a given date
Returns coordinates of pinpoints
Renders Map

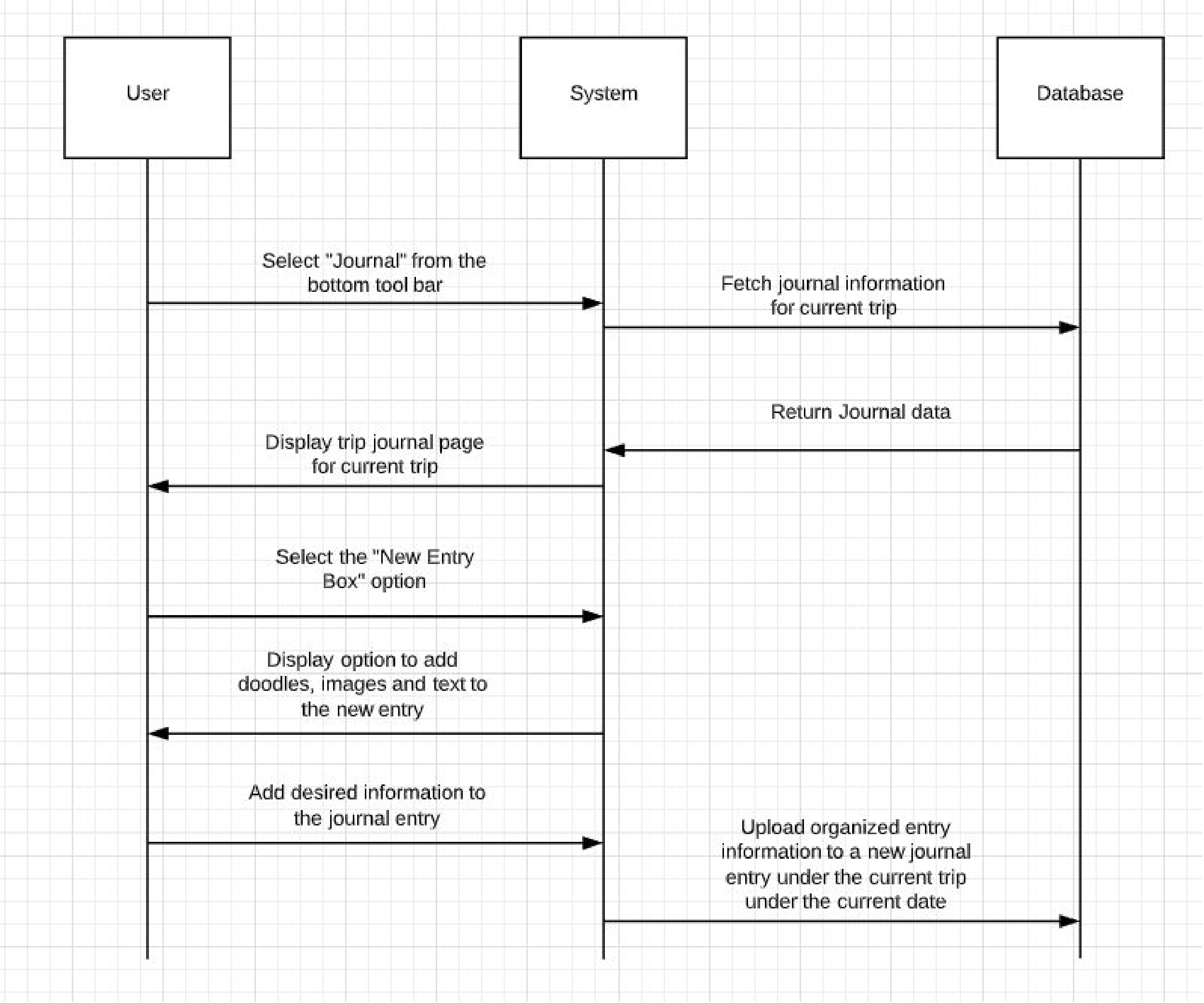
UC-10: Find Attractions


User
System
Database
Click on future or current date on calendar
Make Request to find top attractions
Return top attractions
Display top attractions in area
Filter By Date/Category
Make request to find top attractions
Return top attractions with filtered output
Display Attractions by Filter

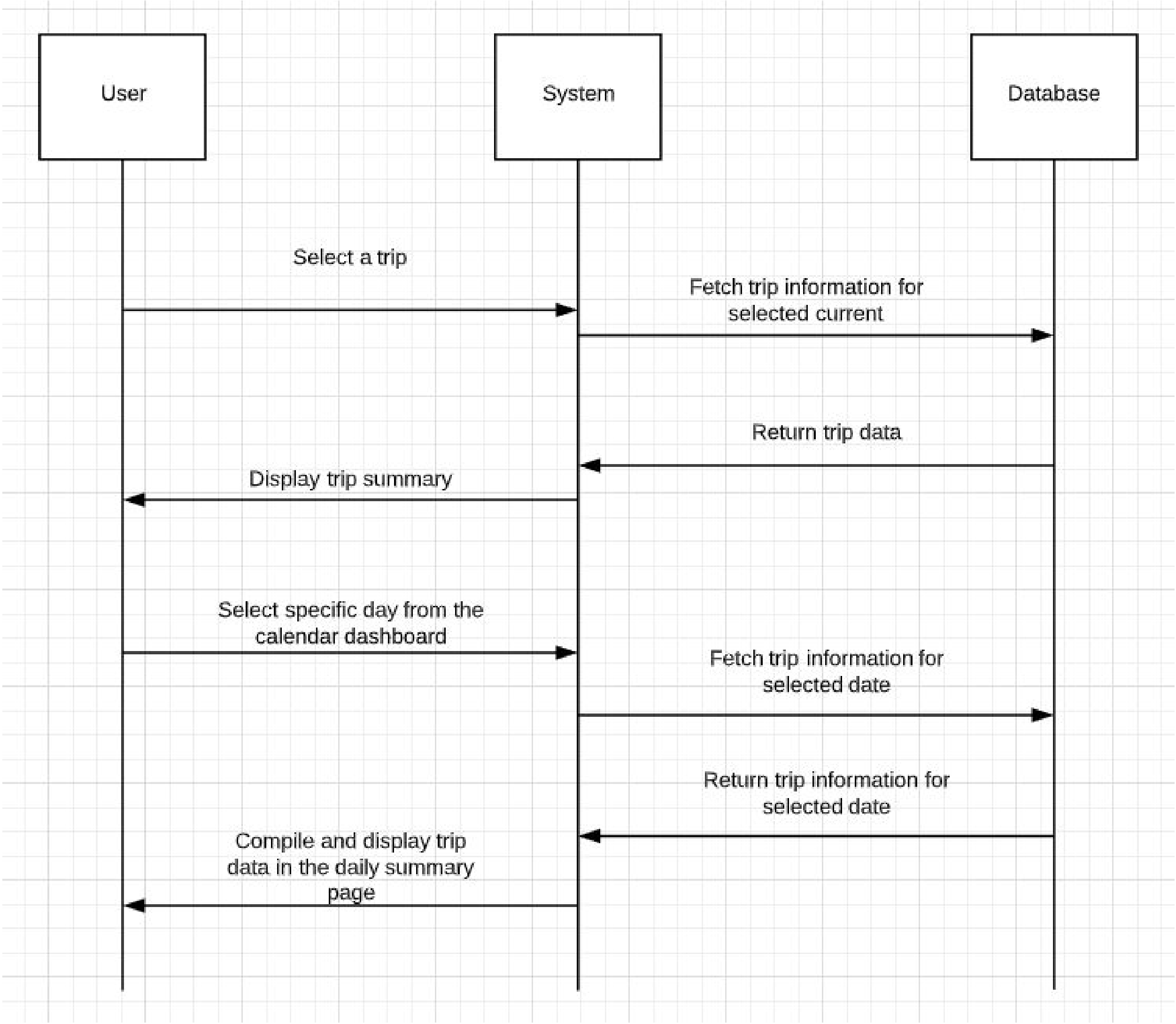
UC-18: View Budget


User
System
Database
Clicks on Budget Tab
Sends request to retrieve budget information
Returns budget information
Display Budget Information
Send Notifications when user is nearing spending limit
Send notifications to remind the user to input money spent
Click on expand budget tab
Display itemized list of expenses

UC-23: Add Journal Entry

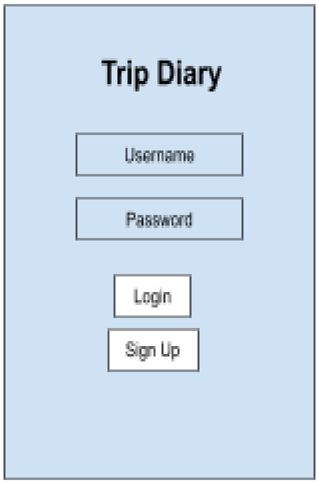
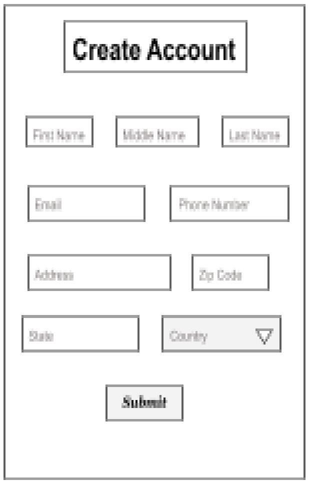

User
System
Database
Select "Journal" from the bottom tool bar
Fetch journal information for current trip
Return Journal data
Display trip journal page for current trip
Select the "New Entry Box" option
Display option to add doodles, images and text to the new entry
Add desired information to the journal entry
Upload organized entry information to a new journal entry under the current trip under the current date

UC-21: View Daily Summary

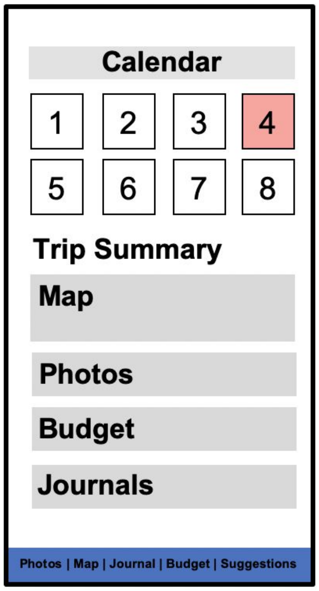
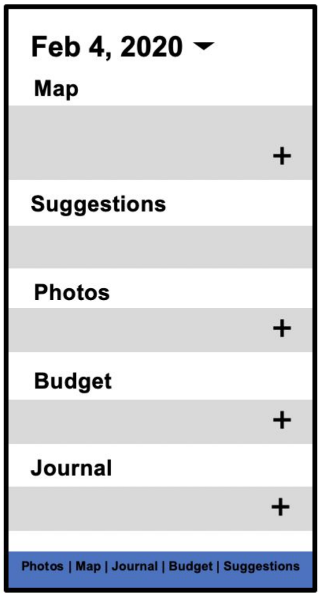

User
System
Database
Select a trip
Fetch trip information for selected current
Return trip data
Display trip summary
Select specific day from the calendar dashboard
Fetch trip information for selected date
Return trip information for selected date
Compile and display trip data in the daily summary page

4. User Interface Specification

A. Preliminary Design & User Effort Estimation

UC-1, UC-2: Create Account, Log-In (1 Navigation)			
Step #	**View**	**Description**	**User Click Rate**
1	 Figure 1	1. The user must enter his or her username and password to enter into the application a. There will be two text fields for the user to enter his information. b. The user may then press the 'Login' button to be taken to the homepage 2. If the user does not have an account then he or she must click 'Sign Up'	1. **Navigation:** Open application
2	 Create Account First Name Middle Name Last Name Email Phone Number Address Zip Code State Country Submit Figure 2	1. The user is taken to the create account view when the 'Sign Up' button is clicked a. The user has to enter information including his name, email, phone number, address, zip code, state, and country. 2. Once the user has entered all of his or her information then they may click the 'Submit' button to return to the login page to login	1. **Navigation:** Click 'Sign Up' on login page

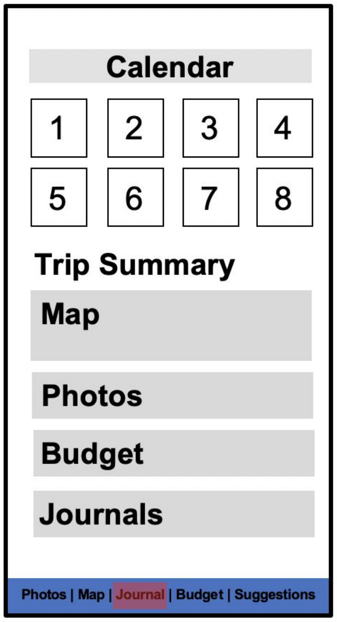
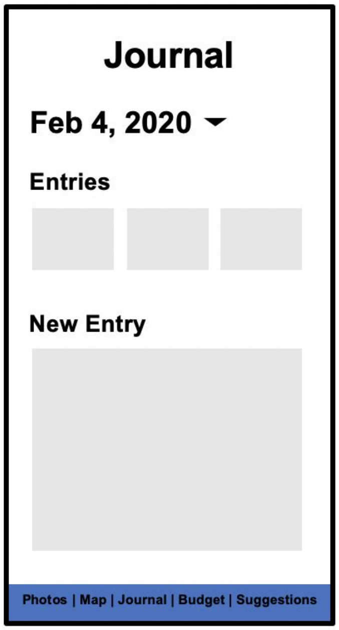
Number of Clicks: 2

UC-21: View Daily Summary (1 Navigation)			
Step #	**View**	**Description**	**User Click Rate**
1	 Calendar 1 2 3 4 5 6 7 8 Trip Summary Map Photos Budget Journals Photos | Map | Journal | Budget | Suggestions Figure 2	3. Once a user selects a trip, they will be lead to their calendar dashboard a. As shown in the image, there will be a display of the dates the user had previously specified when creating a new trip (ex. Feb 1-8 is displayed in Figure 2) 4. From this calendar dashboard, they are able to select a specific date on their trip (ex. Feb 4 is selected, as indicated by the red highlight)	2. **Navigation** Click 'Journal'
2	 Feb 4, 2020 Map + Suggestions Photos + Budget + Journal + Photos | Map | Journal | Budget | Suggestions Figure 3	3. Once the calendar date was selected, the user is able to view a daily summary of their trip a. If they would like to change the date of which they are viewing a summary of, they are able to view the arrow next to the date 4. On the summary page, the user can view an aggregated collection of their landmarks (map), suggestions, photos, budget, and journal	2. **Navigation:** Click different subsections to view full pages of each

Number of Clicks: 2

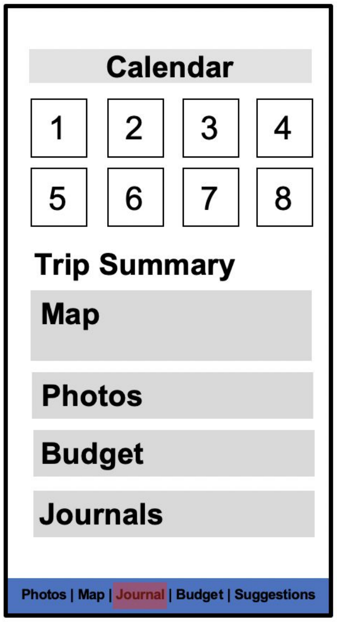
UC-23: Add Journal Entry (2 Navigations)

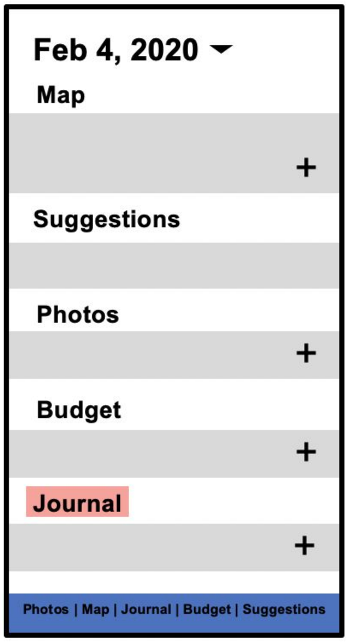
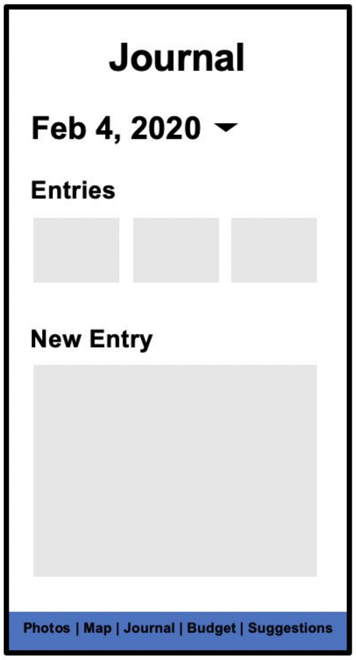
Navigation 1

Step #	View	Description	User Click Rate
1	 Calendar 1 2 3 4 5 6 7 8 Trip Summary Map Photos Budget Journals Photos | Map | Journal | Budget | Suggestions Figure 4	1. Once a user selects a trip, they will be lead to their calendar dashboard 2. A toolbar is presented on the bottom of each page while a trip is selected a. Includes shortcuts within the application to photos, map, journal, budget, and suggestions 3. The user is able to click journal from the toolbar, and it will redirect them to another page	1. **Navigation:** Click 'Journal'
2	 Journal Feb 4, 2020 Entries New Entry Photos | Map | Journal | Budget | Suggestions Figure 5	1. The journal page is presented, in which a user is able to view various entries for that date and add new entries 2. Considering we are accessing the journal from the toolbar and not the calendar, the date will be determined from: a. If the trip is still ongoing, the date will default to the current date b. If the trip is completed (past the last date the user inputted), the date will default to the first date of your trip i. The user is able to change	**Different Scenarios:** 1. **Data Entry:** Click the box for 'New Entry' a. Click on an option whether to type, doodle, or add a picture b. Click enter once written 2. **Navigation:** Click the box for 'Existing Entries' to view previous journals a. Click back button 3. **Navigation:** Click the arrow to change the date (optional)

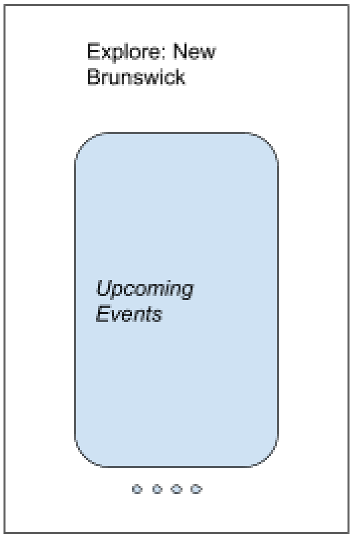
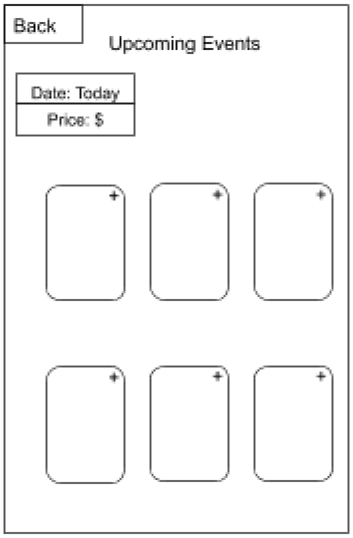
		the date from the arrow next to the presented date (Figure 5) 3. Users are able to add the following features to their new entries a. Doodles b. Images c. Text	

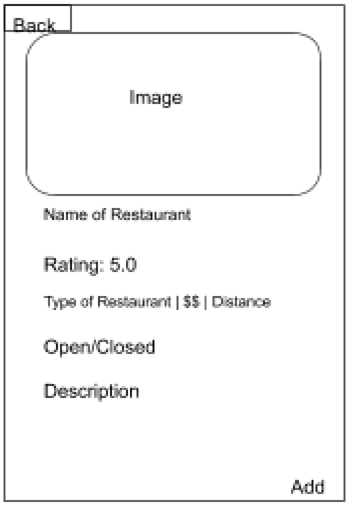
Number of Clicks: 3 (per scenario)

Navigation 2			
Step #	**View**	**Description**	**User Click Rate**
1	 Calendar 1 2 3 4 5 6 7 8 Trip Summary Map Photos Budget Journals Photos | Map | Journal | Budget | Suggestions Figure 6	4. Once a user selects a trip, they will be lead to their calendar dashboard a. As shown in the image, there will be a display of the dates the user had previously specified when creating a new trip (ex. Feb 1-8 is displayed in Figure 6) 5. From this calendar dashboard, they are able to select a specific date on their trip (ex. Feb 4 is selected, as indicated by the red highlight)	1. **Navigation:** Click on the desired date from the calendar

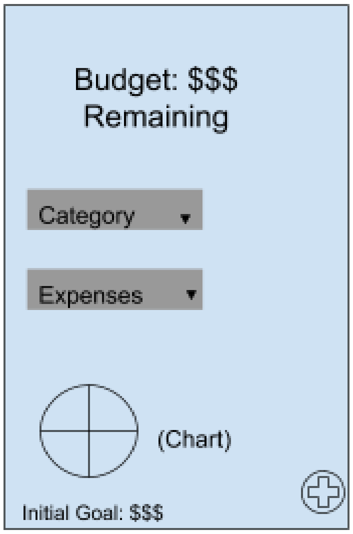
2	 Feb 4, 2020 Map + Suggestions Photos + Budget + Journal + Photos | Map | Journal | Budget | Suggestions Figure 7	1. Once the calendar date was selected, the user is able to view a daily summary of their trip a. If they would like to change the date of which they are viewing a summary of, they are able to view the arrow next to the date 2. On the summary page, the user can view the journal, which will direct them to the journal page (ex. Journal is selected, as indicated by the red highlight)	1. **Navigation:** Click on the 'Journal' subsection
3	 Journal Feb 4, 2020 Entries New Entry Photos | Map | Journal | Budget | Suggestions Figure 8	11. The journal page is presented, in which a user is able to view various entries for that date and add new entries 12. As shown in Figure 8, the user is additionally able to change the date of the journal entries they are inputting 13. Users are able to add the following features to their new entries a. Doodles b. Images c. Text	**Different Scenarios:** 14. **Data Entry:** Click the box for 'New Entry' a. Click on an option whether to type, doodle, or add a picture b. Click enter once written 15. **Navigation:** Click the box for 'Existing Entries' to view previous journals a. Click back button 16. **Navigation:** Click the arrow to change the date

Number of Clicks: 4 (per scenario)

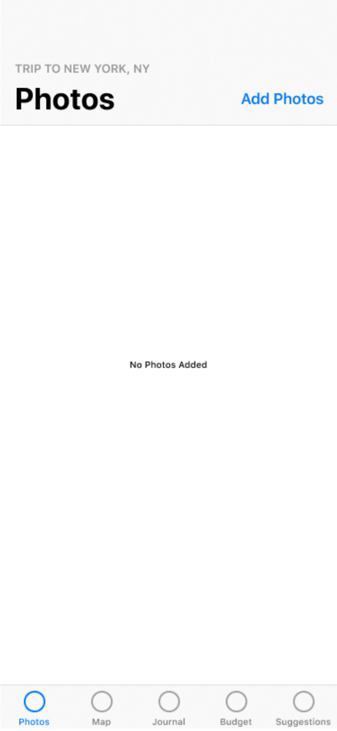
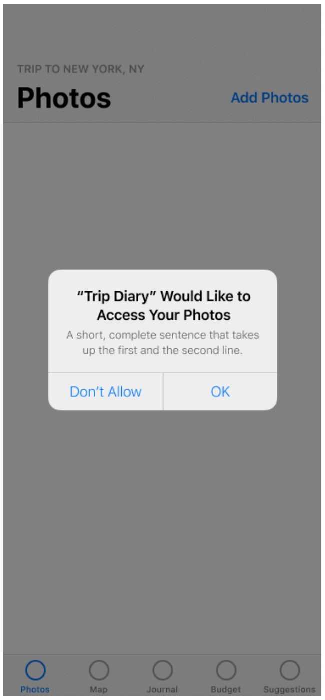
UC-10: Find Attractions (1 Navigation)			
Step #	**View**	**Description**	**User Click Rate**
1	 Figure 9	1. Once a user selects the suggestions tab on the toolbar or clicks on a future date in the calendar they will be lead to a new page a. As shown in the image, there will be a display of different categories of attractions that the user can swipe and select (ex. Restaurants, popular attractions)	1. **Navigation:** Click on the desired category of attraction a. Swipe to see different categories
2	 Figure 10	1. Once a user selects a category of attractions, they will be presented with a list of specific attractions within that category. 2. The user will be able to filter the attractions by price and date	1. **Navigation:** Click on the desired attraction to view more information 2. Click on the filter options to filter attractions 3. Click on the plus symbol to save the attraction for future viewing 4. Click on the back button to return to the categories screen

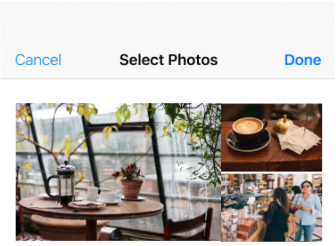
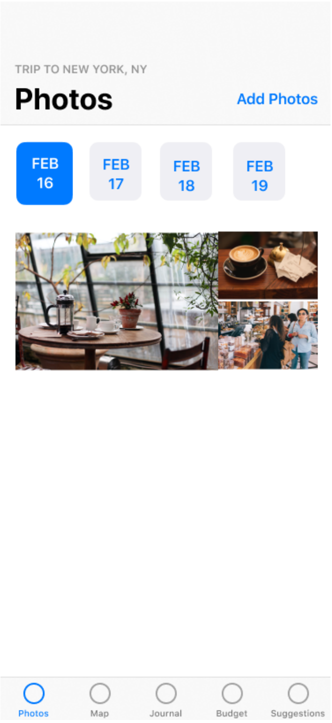
3	 Back Image Name of Restaurant Rating: 5.0 Type of Restaurant | $$ | Distance Open/Closed Description Add Figure 11	1. If a user clicks on a specific attraction they will be brought to an information screen, so they can learn more about the attraction 2. A user can either save an attraction for future viewing here or return back to the attractions list	1. **Navigation:** Click on add to save the attraction 2. Click back to return to the attraction screen

Number of Clicks: 3 (per scenario)

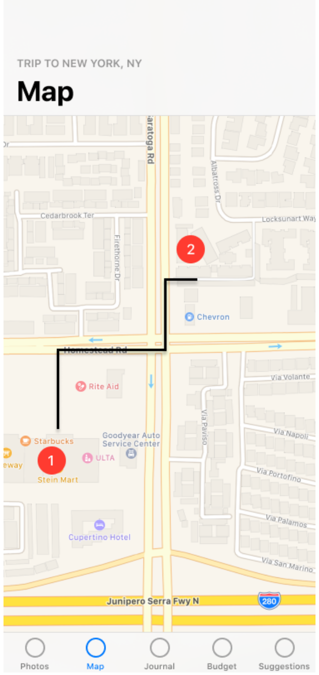
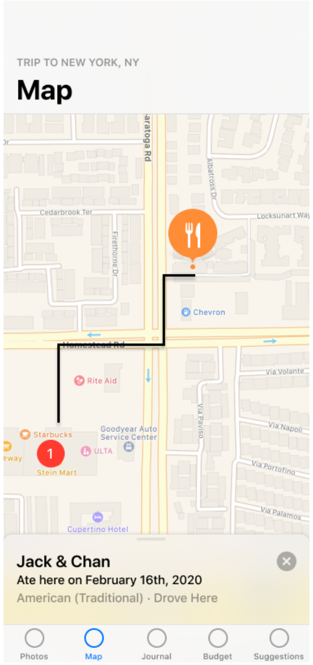
UC-18: View Budget (1 Navigation)			
Step #	**View**	**Description**	**User Click Rate**
1	 Budget: $$$ Remaining Category Expenses (Chart) Initial Goal: $$$ Figure 12	1. Once a user selects the budget tab on the toolbar they will be brought to the budget page b. As shown in the image, a user will be able to see their current total budget remaining, their initial budget, and subcategories of budgets that they specified, as well as an itemized expense list for each of those budgets	2. **Navigation:** Click on the category button to view different sub-budgets (Food, Souvenirs, etc.) 3. Click on the expenses button to see the specific items that the user spent their money on

Number of Clicks: 2 (per scenario)

UC-5: Add Photo			
Step #	**View**	**Description**	**User Click Rate**
1	 TRIP TO NEW YORK, NY Photos Add Photos No Photos Added Photos Map Journal Budget Suggestions Figure 13	1. Once a User creates a new trip and selects the “Photos Tab”, they will be presented with an empty view which displays the empty photo list and a button to direct the user to add photos to their diary. a. As shown in the image, we inform the user of which trip they are currently	1. **Navigation**: Click on Add Photos to import more photos from the camera roll.
2	 TRIP TO NEW YORK, NY Photos Add Photos “Trip Diary” Would Like to Access Your Photos A short, complete sentence that takes up the first and the second line. Don’t Allow OK Photos Map Journal Budget Suggestions Figure 14	2. Once a User clicks on “Add Photo”, the System checks if the user has granted it the permission to view photos from their personal camera roll. a. If the permission has not been granted, the system prompts the user with an alert dialog to request the permission.	1. **Navigation:** From this view, the user can click on OK to allow the system to read photos from the phone.

3	 Figure 15	1. Once the system has permission to view photos from storage, it presents the following view to allow the user to select photos to add to their diary which relate to this trip. 2. Once the photos are selected, the view will mark a checkbox on the photos to indicate selection.	1. **Navigation**: Once the photos have been selected, the user can click "Done" to save their changes and notify the system of the photos they'd like to add to trip diary 2. **Navigation**: If the user would like to cancel their selection, they can click on "Cancel" to tell the system to do nothing.
4	 TRIP TO NEW YORK, NY Photos Add Photos FEB 16 FEB 17 FEB 18 FEB 19 Photos Map Journal Budget Suggestions Figure 16	1. Once the user selects photos to import into the application, the system uploads the files to a central storage bucket, and stores the link for the photo in the database. 2. As you can see from the image, our system then loads these images from the links stored in our database and presents them in a chronological view.	1. **Navigation** The user can scroll through all of their photos within this trip. 2. **Navigation**: The user can select a day on the top bar to jump to photos taken on that day of their trip. 3. **Navigation**: The user can click on "Add Photos" button to import more photos.

Number of Clicks: 4 (per scenario)

UC-6: View Map			
Step #	**View**	**Description**	**User Click Rate**
1	 TRIP TO NEW YORK, NY Map Saratoga Rd Albatross Dr Cedarbrook Ter Locksunart Way Firethorne Dr 2 Chevron Homestead Rd Via Volante Rite Aid Via Paviso Via Napoli Starbucks Goodyear Auto Service Center 1 ULTA Stein Mart Via Portofino Cupertino Hotel Via Palamos Via San Marino Junipero Serra Fwy N 280 Photos Map Journal Budget Suggestions Figure 17	1. Once the user clicks on the map view for the current trip, the system talks to the database and fetches the points logged along with the route used to visit that point. 2. As you can see from the image, the system then renders the points on a map, and represents the route (should be a dotted line)	3. **Navigation**: The user can now browse the map by panning or zooming to view more pins/get more details of the area. 4. **Navigation**: The user can click on any pin to view more details about that logged location.
1	 TRIP TO NEW YORK, NY Map Saratoga Rd Albatross Dr Cedarbrook Ter Locksunart Way Firethorne Dr Chevron Homestead Rd Via Volante Rite Aid Via Paviso Via Napoli Starbucks Goodyear Auto Service Center 1 ULTA Stein Mart Via Portofino Cupertino Hotel Via Palamos Photos Map Journal Budget Suggestions Jack & Chan Ate here on February 16th, 2020 American (Traditional) · Drove Here Figure 18	1. Once the user clicks on any location pin, the view highlights the location logged with the type, in this example a restaurant was visited. 2. As you can see on the image, a sheet opens up at the bottom to provide more information, such as date and time of visit, mode of transportation used to reach there.	1. **Navigation:** The user can continue to view the map by panning around the area to view more pins 2. **Navigation**: The user can exit the pin using the X on the bottom sheet to clear the pin selection.

Number of Clicks: 2 (per scenario)

5. Domain Analysis

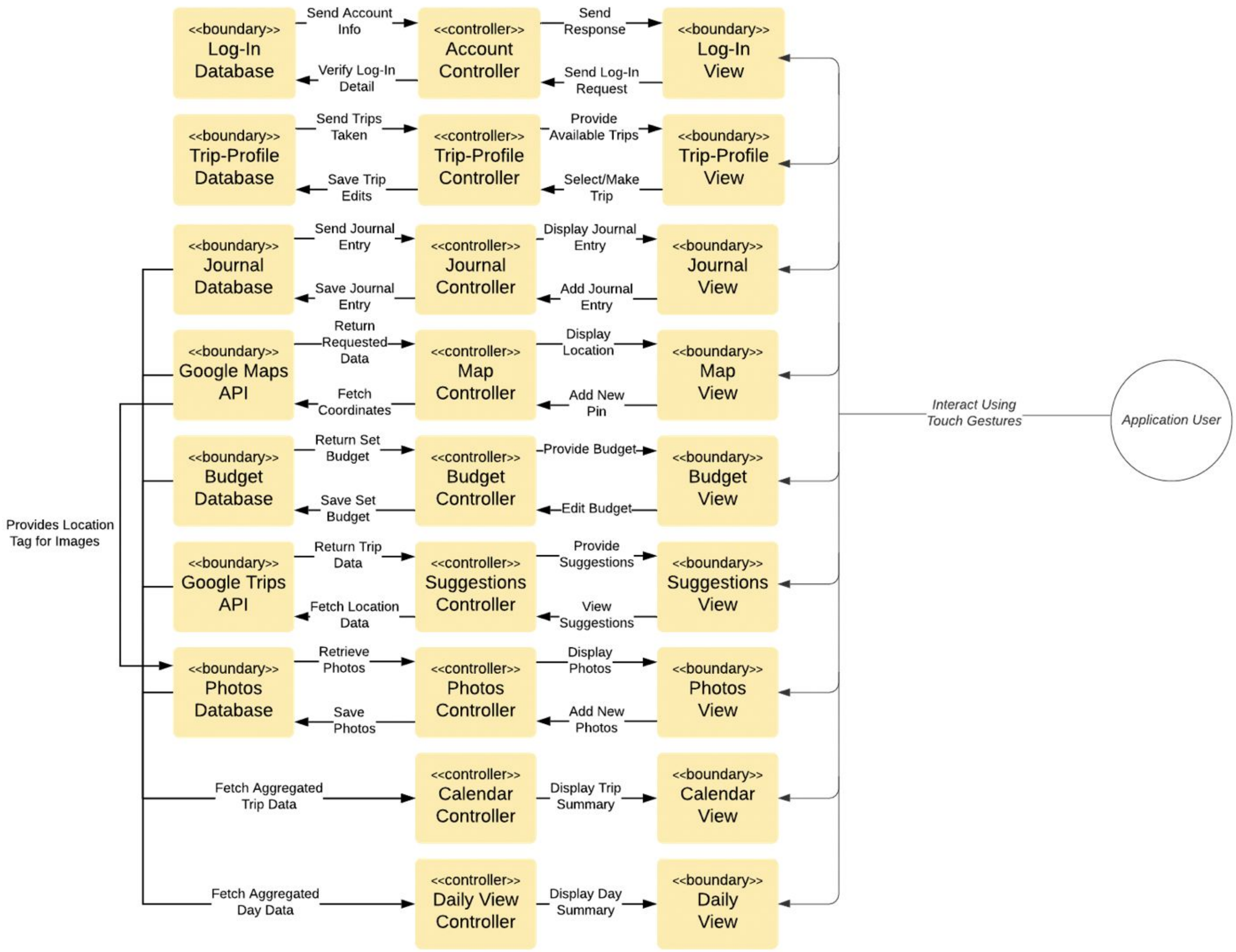
A. Domain Model

I. Concept Definitions

Responsibility	**Type**	**Concept**
R1: Store and prepare a database query that retrieves log-in and account validation records	D	Log-In Database
R2: Store and prepare a database query that retrieves created trips	D	Trip-Profile Database
R3: Store and prepare a database query that retrieves stores journal entries	D	Journal Database
R4 To supply a map interface and GPS coordinates corresponding to location	D	Google Maps API
R5: To supply locations for attractions that are within a certain distance from a location	D	Google Trips API
R6: Store and prepare a database query that retrieves budgets created and items purchased	D	Budget Database
R7: Store and prepare a database query that retrieves photos	D	Photos Database
R8: Coordinates actions to create new account or log-in and delegate work to other concepts	D	Account Controller
R9: Coordinates actions to create new account or log-in	D	Trip Profile Controller
R10: Coordinates actions to add/edit new journals and retrieve existing entries	D	Journal Controller
R11: Coordinates actions to add/edit location pins and retrieve locations visited	D	Map Controller
R12: Coordinates actions to add/create budgets and collect spending trends	D	Budget Controller
R13: Coordinates actions to collect recommendations	D	Suggestions Controller
R14: Coordinates actions to upload new photos	D	Photos Controller
R15: Coordinates actions to provide inputted data for the entire	D	Calendar Controller

trips and dates trip was taken		
R16: Coordinates actions to provide inputted data for a specific day	D	Daily View Controller
R17: Generated code which shows current concepts associated with log-in/making a new account, what actions can be done, and outcomes of previous actions	K	Log-In View
R18: Generated code which shows current concepts associated with trip profile(s) created, what actions can be done, and outcomes of previous actions	K	Trip-Profile View
R19: Generated code which shows current concepts associated with journal entries, what actions can be done (writing, editing, viewing entries), and outcomes of previous actions	K	Journal View
R20: Generated code which shows current concepts associated with locations visited, what actions can be done (adding pins, viewing places visited), and outcomes of previous actions	K	Map View
R21: Generated code which shows current concepts associated with the budget, what actions can be done (editing the budget, adding limitations), and outcomes of previous actions	K	Budget View
R22: Generated code which shows current concepts associated with trip suggestions, what actions can be done (viewing recommended places), and outcomes of previous actions	K	Suggestions View
R23: Generated code which shows current concepts associated with photos, what actions can be done (adding geotags, uploading/deleting photos), and outcomes of previous actions	K	Photos View
R24: Generated code which shows current concepts associated with the general trip,, what actions can be done (selecting different dates, viewing a trip summary), and outcomes of previous actions	K	Calendar View
R25: Generated code which shows current concepts associated with the specific date,, what actions can be done (accessing photos, etc., viewing a daily summary), and outcomes of previous actions	K	Daily View

II. Domain Diagram


<<boundary>> Log-In Database
Send Account Info
Verify Log-In Detail
<<controller>> Account Controller
Send Response
Send Log-In Request
<<boundary>> Log-In View
<<boundary>> Trip-Profile Database
Send Trips Taken
Save Trip Edits
<<controller>> Trip-Profile Controller
Provide Available Trips
Select/Make Trip
<<boundary>> Trip-Profile View
<<boundary>> Journal Database
Send Journal Entry
Save Journal Entry
<<controller>> Journal Controller
Display Journal Entry
Add Journal Entry
<<boundary>> Journal View
<<boundary>> Google Maps API
Return Requested Data
Fetch Coordinates
<<controller>> Map Controller
Display Location
Add New Pin
<<boundary>> Map View
Interact Using Touch Gestures
Application User
<<boundary>> Budget Database
Return Set Budget
Save Set Budget
<<controller>> Budget Controller
Provide Budget
Edit Budget
<<boundary>> Budget View
Provides Location Tag for Images
<<boundary>> Google Trips API
Return Trip Data
Fetch Location Data
<<controller>> Suggestions Controller
Provide Suggestions
View Suggestions
<<boundary>> Suggestions View
<<boundary>> Photos Database
Retrieve Photos
Save Photos
<<controller>> Photos Controller
Display Photos
Add New Photos
<<boundary>> Photos View
Fetch Aggregated Trip Data
<<controller>> Calendar Controller
Display Trip Summary
<<boundary>> Calendar View
Fetch Aggregated Day Data
<<controller>> Daily View Controller
Display Day Summary
<<boundary>> Daily View

Note: In actuality, there is one database that stores information in the system architecture. For practicality in the domain model, it was divided to provide a more detailed interaction schematic.

III. Association Definitions

Concept Pair	Association Description	Association Name
Log-In Database ↔ Account Controller	Database provides account information after the controller asks for verifications to access a user account	Provides Data
Trip Profile Database ↔ Trip Profile Controller	Database provides trips associated with the user's account to allow the user to	Provides Data

	view their separate diaries	
Journal Database ↔ Journal Controller	Database provides journal entries previously written to allow users and allow users to save them	Provides Data
Google Maps API ↔ Map Controller	API fetches image of requested coordinates of map and sends it to map controller	Provides Data
Budget Database ↔ Budget Controller	Database fetches budget and expense information to allow users to access their previously entered data	Provides Data
Google Trips API ↔ Suggestions Controller	API fetches a list of locations based on a user's location, so that the user is presented with several locations they can travel to	Provides Data
Photos Database ↔ Photos Controller	Database fetches stored photos from the specific trip the user is looking through and sends them to photos controller	Provides Data
Google Maps/Trips API, Journal/Budget/Photos Database ↔ Calendar Controller	Database provides all listed information that users inputted to their trip diaries to provide collected data	Provides Data
Google Maps/Trips API, Journal/Budget/Photos Database ↔ Daily View Controller	Database provides all listed information that users inputted to a specific date on their trip to provide collected data	Provides Data
Account Controller ↔ Log-In View	Displays login textboxes and redirects user to homepage once successful login has been complete	Conveys Request
Trip-Profile Controller ↔ Trip-Profile View		Conveys Request
Journal Controller ↔ Journal View	Displays journal entries for user's view and handles when user writes or edits an entry	Conveys Request
Map Controller ↔ Map View	Displays comprehensive image of map with the tracking information drawn onto it and pins placed onto it	Conveys Request
Budget Controller ↔ Budget View	Displays budget information that the user entered in	Conveys Request
Suggestions Controller ↔ Suggestions	Displays list of suggestions for user to	Conveys Request

View	pick from to save to their trip schedule	
Photos Controller ↔ Photos View	Displays a list of all the photos saved by the user into the app and allows user to interact with pictures (zoom in and out)	Conveys Request
Calendar Controller ↔ Calendar View	Displays aggregated information from entire trip and trip dates for the user to interact and view a trip summary of the inputs to their diary (photos, locations, etc.)	Conveys Request
Daily View Controller ↔ Daily View	Displays aggregated information from entire trip and trip dates for the user to interact and view a specific day's summary of the inputs to their diary (photos, locations, etc.)	Conveys Request
Google Maps API ↔ Photos Database	Every time a photo is uploaded, the Google Maps API will fetch the user's current location and save the coordinates of the user with the photo	Provides Data

IV. Attributes Definitions

Concepts	**Attributes**	**Description**
Log in View	1. User Identification	1. Username and Password
Budget	1. Budget Creation/Edit 2. Statistics 3. Notifications	1. Can add budget, set category, set reminders, etc. 2. Average amount spent per day, and which category they spend on the most 3. Reminds user when to add expenses to budget
Suggestions	1. Suggestion Retrieval 2. Saved Attractions 3. Notifications	Retrieves several locations and location information around a user's destination Can save attractions for future viewing Can commit to an attraction, which will send notifications to remind the user when to go
Calendar View	1. Dates/Length of Trip 2. Current date 3. Days	1. Conveys when the trip takes place and for how long 2. Current date to be displayed in calendar

		3. The calendar has access to different day objects of the trip
Daily View	1. Day Identifier 2. Events List	1. Identifies which day of the trip is selected 2. Events that the user has or plans to participate in. This includes journal entries, photos, events, suggestions, etc.
Journal View	1. Creation Method 2. Random Prompt	1. Can be writing, drawing, taking a photo, uploading a video, etc. 2. A random writing prompt will question the user to help spur creativity and ease the process of writing

V. Traceability Matrix

Use Case	PW	Domain Concepts							
		Google Maps API	Google Trips API	Login View	Trip Profile View	Map View	Budget View	Journal View	Daily View
UC-1	5			X					
UC-2	5			X					
UC-3	4				X				
UC-4	7								
UC-5	3								
UC-6	5	X			X	X			X
UC-7	2				X	X			X
UC-8	4	X				X			X
UC-9	6				X				X
UC-10	4								
UC-11	4								
UC-12	5								

UC-13	1								
UC-14	5		X						
UC-15	5						X		
UC-16	3						X		
UC-17	7						X		
UC-18	9						X		
UC-19	6						X		
UC-20	16				X				
UC-21	32								X
UC-22	16							X	
UC-23	9							X	
UC-24	8							X	
Max PW		5	5	5	6	5	9	16	32
Total PW		9	5	10	17	11	30	33	49

Use Case	PW	Domain Concepts							
		Login DB	**Trip Prof DB**	**Journal DB**	**Budget DB**	**Photo DB**	**Sug. View**	**Photos View**	**Cal View**
UC-1	5	X							
UC-2	5	X							
UC-3	4		X						X
UC-4	7					X		X	
UC-5	3					X		X	
UC-6	5		X						
UC-7	2		X						

UC-8	4								
UC-9	6		X						
UC-10	4						X		
UC-11	4						X		
UC-12	5						X		
UC-13	1						X		
UC-14	5						X		
UC-15	5				X				
UC-16	3				X				
UC-17	7				X				
UC-18	9				X				
UC-19	6				X				
UC-20	16		X						X
UC-21	32								
UC-22	16			X					
UC-23	9			X					
UC-24	8			X					
Max PW		5	6	16	9	7	5	7	16
Total PW		10	17	33	30	10	19	10	20

Use Case	**PW**	**Domain Concepts**								
		Account Control.	**Trip Prof Control.**	**Journal Control.**	**Map Control.**	**Budget Control.**	**Sug. Control.**	**Photos Control.**	**Cal. Control.**	**Daily View Control.**
UC-1	5	X								
UC-2	5	X								

UC-3	4		X						X	
UC-4	7							X		
UC-5	3							X		
UC-6	5		X		X					X
UC-7	2		X		X					X
UC-8	4				X					X
UC-9	6		X							X
UC-10	4						X			
UC-11	4						X			
UC-12	5						X			
UC-13	1						X			
UC-14	5						X			
UC-15	5					X				
UC-16	3					X				
UC-17	7					X				
UC-18	9					X				
UC-19	6					X				
UC-20	16		X						X	
UC-21	32									X
UC-22	16			X						
UC-23	9			X						
UC-24	8			X						
Max PW		5	6	16	5	9	5	7	16	32
Total PW		10	17	33	11	30	19	10	20	49

B. System Operation Contracts

1) Photos

Responsibilities: Photos need to be uploaded to a given album. The system must store photos by querying a database and also be efficient in storing photos, meaning it doesn't exceed memory. It must also notify you of failure.

Cross References: UC-5: AddPhoto, UC-4: ViewPhotos

Expectations: As discussed, the system should be able to query a database in order to add a new entry for a photo. If there is a failure to access the database, then the system should provide an alert detailing the causes of failure.

Preconditions: The application must be given permission to access the photo. There should be access to a database that is either set up or can be set up.

Postconditions: Notify the user that the picture has been uploaded and refresh the page. If it failed, then there should be an alert detailing why.

2) Map

Responsibilities: The system should load the pinpoints that the user has saved during their trip onto a map. It needs to store metadata about each location such as time visited, coordinates, etc. The system should load the map so that all pinpoints can be viewed.

Cross References: UC-6: ViewMap, UC-7: ViewMap

Expectations: The system should query the database to retrieve all necessary information needed to form the map, particularly the pinpoints. Then the system should calibrate the map so that it fits all pinpoints into a view. Also, if there are no pinpoints then the system should center the map based off of a generic coordinate.

Preconditions: Connection is or can be established with the database to retrieve pinpoint metadata. If there are no pinpoints, then there we can come up with a generic location, using an API, to center the map.

Postconditions: The system renders the map around the coordinates of the pinpoints.

3) Budget

Responsibilities: The system should save and delete finances that the user inputs into their budgeting. The system should be able to retrieve information so that budget statistics can be made.

Cross References: UC-18: ViewBudget, UC-17: BudgetStatistics, UC-15: LogFinance, UC-16: DeleteFinance, UC-19: Make/ChangeBudget

Expectations: The information should be stored in a database so that the system can develop charts, graphs, and statistical variables to present to the user as a visual and mathematical view of their spending habits.

Preconditions: Database can store financial information that the user inputs and there are API's used to generate suggestions for finances.

Postconditions: System develops budget statistics with all the budgeting information.

4) Attractions

Responsibility: The system should be able to suggest attractions on certain dates based on the location of the user's travels. The system should also save attractions so that the user can view them later and should also allow the user to delete attractions they are no longer interested in. If a user wishes to commit to an attraction, they should be able to do so.

Cross References: UC-10: FindAttractions, UC-11: SaveAttraction, UC-12: CommitAttraction, UC-13: DeleteAttraction, UC-14: ViewAttractions

Expectations: All this information should be input to the database so that the system can keep track of saved attractions and committed attractions and so that the system can remind the user of upcoming committed attractions.

Preconditions: An API is used to develop suggested attractions/events on a date based on the location of the user's travels.

Postconditions: System has a list of saved attractions and committed attractions.

5) Daily Summary

Responsibility: The system needs to retrieve the user's saved photos, journal entries, events, budget from a database relating to the chosen day. It must then generate a page for the user to see all this information and provide links to add photos, journal entries, etc. It must also notify the user of any failure such as not having access to the Internet.

Cross References: UC-4: View Photos UC-6: View Map UC-14: View Attractions UC-18: View Budget UC-22: Journaling

Expectations: The system requests the information from the database, filtering all the information based on what day they were added. Once the system has the information, it will generate a new page on the mobile device displaying all of the retrieved information. The system will also provide the user a link to add to the chosen day.

Preconditions: The system must be able to establish a connection to the database so that information can be transferred which includes the user retrieving previously added information as well as the user saving information to the database. The system must also have access to the user’s photos/gps location if they want to save those to the specific day.

Postconditions: All of the user’s saved information pertaining to the chosen day is displayed on a new page on their mobile device. Links are given in which they can add or save new information.

6) Add Journal Entry

Responsibility: The system will generate a new page on the mobile device for the user to make their journal entry. The system will then generate a random prompt as well as provide the user with links/buttons to direct them to a page where they can add photos, videos, or drawings. The system can then save the journal entry to the desired day/trip in the database.

Cross References: UC-22: Journaling, UC-23: Add Journal Entry, UC-24, Delete/Edit Journal Entry

Expectations: The system should be able to operate on its own without the database to generate the page, and create the prompt. However, the database will be used if the user decides to add a photo or save the journal entry.

Preconditions: The user must be able to navigate through the daily summary or trip summary page to where they can add a journal entry to a specific trip/day. There must also be a database connection so that the user can access their photos on that chosen day or save the journal entry.

Postconditions: The user creates a journal entry with photos, videos, drawings, writing and saves it to the desired day or trip.

6. Project Size Estimation

Unadjusted Use Case Points (UUCP) quantifies our project's functional features based on the amount and complexity of our use cases. To find the UUCP, we must first define our Unadjusted Use Case Weight (UUCW) scheme; in other words we will have to decide what weight value gets assigned to each complexity and why.

Use Case Complexity	Number of Steps	Weight Value
Simple	1 to 3 Steps	5
Average	4 to 7 Steps	10
Complex	8 or more Steps	15

Unadjusted Use Case Weights (UUCW)

Use Case	Description	Category	Weight
UC-1: CreateAccount	2 steps to achieve the main success scenario : to allow new users to create an account. Involves 3 main actors: User and Database	Simple	5
UC-2: Login	1 steps to achieve the main success scenario : to allow users to login to pre existing accounts. Involves 3 main actors: User and Database	Simple	5
UC-3: CreateTrip	3 steps to achieve the main success scenario : to allow users to create new trips with starting and ending date, title, location, and other general information. Involves 3 main actors: User, System and Database	Average	10
UC-4: ViewPhotos	4 steps to achieve the main success scenario : users should be able to sort photos taken on the trip either by day, or by the location they were taken,	Complex	15

	should be able to scroll through photos and zoom into them for more detail. Involves 3 main actors: User, System and Database		
UC-5: Add Photo	Average User Interface. 3-4 steps to complete the main success scenario of adding photos into the photos view. 3 Participating Actors (User, Database, PhotoStorage).	Average	10
UC-6: View Map	Complex User Interface. Includes user interaction types such as panning, zooming, and clicking within one view. Includes multiple layers of views: the underlying map, layer of pins for photos, layer of pins for attractions, layer of routes for location history data. 4 Participating Actors (User, Database, GeoLocation, MappingService)	Complex	15
UC-7: AddPin	3 steps to achieve the main success scenario : users should be able to add a pin with a title and should be able to add photos, specific details, and other information about that pin. The location of the pin should be based on their real time location. Involves 4 main actors: User, System, Google Maps API and Database	Average	10
UC-8: TrackUser	4 steps to achieve the main success scenario : to allow users to save where exactly they went during the day by tracking their real time location and storing it. Involves 4 main actors: User, System, Google Maps API and Database	Complex	15
UC-9: ViewDailySummary	4 steps to achieve the main success scenario : allow users to view a page that has	Complex	15

	the daily summary of their trip consisting of photos, location tracking, and pins placed on the map from where they can edit all these details. Involves 4 main actors: User, System, Google Maps API and Database		
UC-10: FindAttractions	4 steps to achieve the main scenario: Retrieves location information on different places near a user’s location, search for locations on their own, or filter locations based on certain criteria. Involves 3 main actors: User, System, Trips api	Complex	15
UC-11: SaveAttraction	3 steps to achieve the main scenario: Users can bookmark locations and location information to view at a future time. Involves 3 main actors: User, database, and system	Average	10
UC-12 CommitAttraction	4 steps to achieve the main scenario: Users can commit attractions to set up reminder notifications about locations they want to visit Involves 3 main actors: User, database, and system	Average	10
UC-13 DeleteAttraction	2 steps to achieve the main scenario: Users can remove attractions they saved or committed to. Involves 3 main actors: User, database, and system	Simple	5
UC-14 ViewAttractions	4 steps to achieve the main success scenario: Users can view different categories of attractions and information cards for each location	Average	10

	Involves 3 main actors: User, trips api, system		
UC-15 LogFinance	3 steps to achieve the main success scenario: Users can add expenses that they made to their budget through manual entry. Involves 3 main actors: User, system, database	Simple	5
UC-16 DeleteFinance	3 steps to achieve the main success scenario: Users can remove expenses that they manually entered; Involves 3 main actors: User, system, database	Simple	5
UC-17: BudgetStatistics	3 steps to achieve the main success scenario: Users will be able to view different statistics based on their spending habits, and a graph will be displayed. Involves 3 main actors: User, system, database	Average	10
UC-18: ViewBudget	2 steps to achieve the main success scenario: to display each budget that the user created. Involves 3 main actors: User, system, and database	Simple	5
UC-19: Make/Change budget	4 steps to achieve the main success scenario: to allow users to enter and edit their budget information Involves 3 main actors: user, system, and database	Average	10
UC-20: ViewCalendar/ TripSummary	4 steps to achieve main success scenario: to display trip summaries in a calendar format. Involves 3 main actors: User, System and Database	Average	10
UC-21: ViewDailySummary	2 steps to achieve main success scenario: to display a single daily summary of a specific trip. Involves 3 main actors: User,	Simple	5

	System and Database		
UC-22: ViewJournal	3 steps to achieve the main success scenario: to view a journal entry. Involves 3 main actors: User, System and Database	Simple	5
UC-23: AddJournalEntry	8 steps to achieve main success scenario: to complete a journal entry and upload to database. Involves 4 main actors: User, System, Database and Geolocation API	Simple	15
UC-24: Delete/EditJournalEntry	4 steps to achieve main success scenario: to remove a journal entry from the database. Involves 3 main actors: User, System and Database	Average	10

UUCW = *(Total Number of Simple Use Cases x 5)* +
(Total Number of Average Use Case x 10) + *(Total Number of Complex Use Cases x 15)*

Then, the UUCW for our system would be (8 * 5) + (10 * 10) + (6 * 15) = 230

Unadjusted Actor Weights (UAW)

Similar to the UUCP, the Unadjusted Actor Weight (UAW) quantifies our project's functional features. However, the UAW is based on the type of our functional actors. The following table shows the standard scheme that we use to classify actors into types:

Actor Classification	Class Description	Weight Value
Simple	External system that must interact with the system using a well-defined API	1
Average	External system that must interact with the system using standard communication protocols (e.g. TCP/IP, FTP, HTTP, database)	2
Complex	Human actor using a GUI application interface	3

Actor	Description	Complexity	Weight
User	User is interacting with the system via a graphical user interface presented in a mobile application	Complex	3
Database	The database holds our data and interacts with the system through a defined API.	Simple	1
PhotoStorage	PhotoStorage is the system used to store the photos a user uploads onto a remote server and interacts with our system through a defined API.	Simple	1
System	The system holds the controller and therefore is the main “decision maker” for our system. It also reads and compiles information received from the database and will display that information through the user interface (touch-screen display)	Complex	3
GeoLocation	GeoLocation is the system used to fetch user locations and resolve coordinates and interact with our system through a defined API.	Simple	1
Places	Places is the system used to fetch locations around a user’s location and interact with our system through a defined API	Simple	1
MappingService	MappingService is the system used to fetch map images and render the user interactions with the map.	Average	2

Since the Unadjusted Actor weight is defined as:

> ***UAW*** *= (Total Number of Simple actors x 1) +*
> *(Total Number of Average actors x 2) + (Total Number of Complex actors x 3)*

Then the UAW for our system would be **(4 x 1) + (1 x 2) + (2 x 3) = 12**

Technical Complexity Factor (TCF)

TCF is found by assigning weights between 0 (useless) to 5 (essential) to all of the following technical factors which may come up:

Factor	Description	Weight	Perceived Complexity	Calculated Factor
T1	Distributed System, Web based system	2	4	8
T2	Response time/performance objectives should be exceptional	1	3	3
T3	End-user efficiency	1	3	3
T4	Internal processing complexity is simple	1	3	3
T5	Code reusability should not be too important	1	0	0
T6	Easy to install as an application is very important	0.5	1	0.5
T7	Easy to use as an application is very important	0.5	4	2
T8	Portability to other platforms such as android, iOS, and Web is very important	2	4	8
T9	System maintenance should be easy to have to make changes	1	3	3
T10	Concurrent/parallel processing is important to have geolocation running concurrently	1	3	3
T11	Security features are important since we hold sensitive information	1	2	2
T12	Access for third parties is irrelevant	1	0	0
T13	End user training is not needed	1	0	0
Total:				35.5

Since the Technical Complexity Factors is defined as:

$$\textbf{\textit{TCF}} = 0.6 + 0.01 \times \textit{Technical Factor Total}$$

Then the TCF for our system would be .6 + .355 = 0.955

Environmental Complexity Factor (ECF)

The last factor contributing to the total Use Case Points would be the Environmental Complexity Factor or ECF. The ECF is determined by taking into consideration any environmental conditions that the system will be working in. For the purposes of this system, we will assume ECF to be 1. In other words, the ECF will have no contribution to our UCP calculations.

Calculation

UCP = 242 * 0.955 * 1 = 231.11 ≈ 231
ECF = 1
UUCP = 12 + 230 = 242
UAW = 12
UUCW = 230
TCF = 0.955

7. Plan of Work

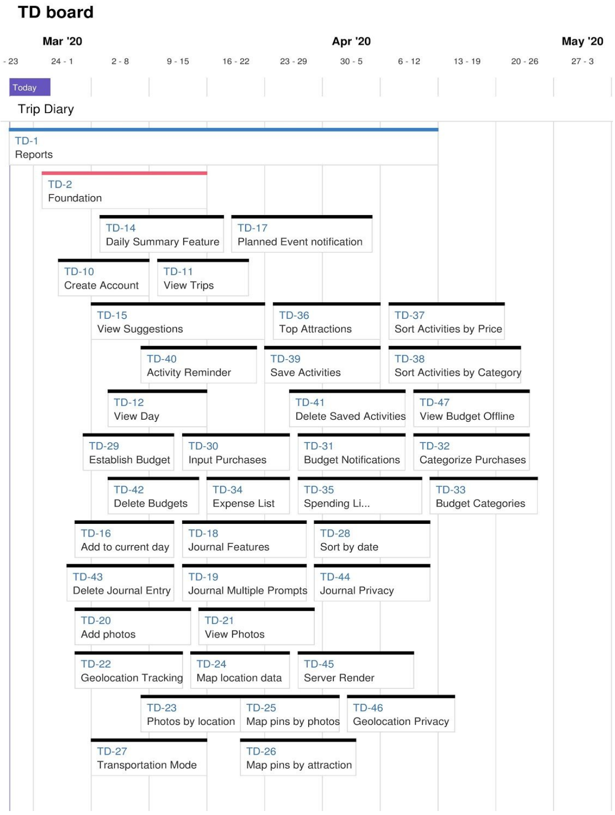
Project Roadmap and Gantt Charts

We first provide a high-level view of our timeline with the project roadmap and then delve into the specifics of our expectations for each major part.

We are using the JIRA software management tool to setup an initial timeline for our project. We have transferred all system requirements into individual tickets, each of which is allocated time proportional to its complexity. Then, we used a third-party roadmap integration to visualize our timeline. We have many tickets running concurrently because of how we decided to divide our group into subgroups, each tackling a subproblem at the same time. Within these smaller groups we can apply a division of labor. Each ticket also has a priority associated with it. Tickets pushed to later dates have a smaller priority because they aren’t as fundamental to the application. We are striving to get the fundamentals done in time for the first demo in the week of March 23.

We have concluded that the major pieces in our application are: the base of the app, called the fundamental; the calendar; the journal; the map; photos; suggestions; geolocation; and budget. For each of these, we have created a gantt chart displaying the amount of work dedicated to each part, along with who is responsible for it and a visual of the timeline.

Overall



TD board

Trip Diary

TD-1 Reports

TD-2 Foundation

TD-14 Daily Summary Feature

TD-17 Planned Event notification

TD-10 Create Account

TD-11 View Trips

TD-15 View Suggestions

TD-36 Top Attractions

TD-37 Sort Activities by Price

TD-40 Activity Reminder

TD-39 Save Activities

TD-38 Sort Activities by Category

TD-12 View Day

TD-41 Delete Saved Activities

TD-47 View Budget Offline

TD-29 Establish Budget

TD-30 Input Purchases

TD-31 Budget Notifications

TD-32 Categorize Purchases

TD-42 Delete Budgets

TD-34 Expense List

TD-35 Spending Li...

TD-33 Budget Categories

TD-16 Add to current day

TD-18 Journal Features

TD-28 Sort by date

TD-43 Delete Journal Entry

TD-19 Journal Multiple Prompts

TD-44 Journal Privacy

TD-20 Add photos

TD-21 View Photos

TD-22 Geolocation Tracking

TD-24 Map location data

TD-45 Server Render

TD-23 Photos by location

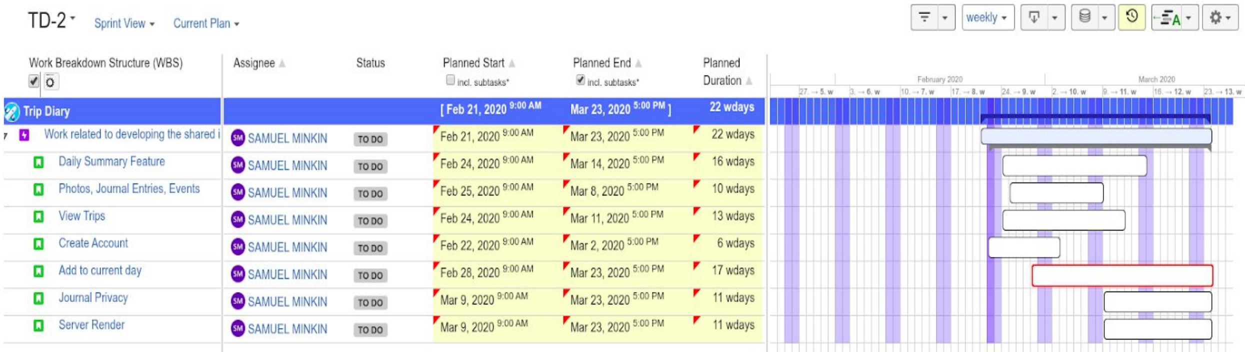
TD-25 Map pins by photos

TD-46 Geolocation Privacy

TD-27 Transportation Mode

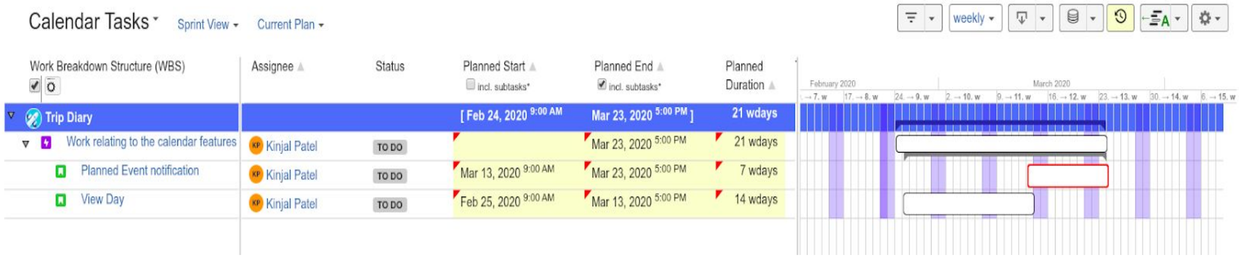
TD-26 Map pins by attraction

Fundamental



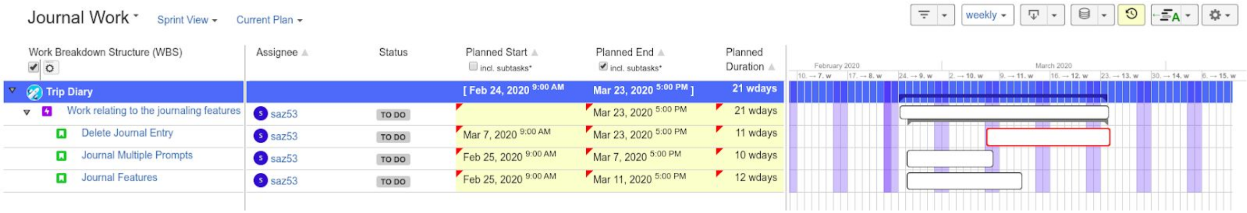
Work Breakdown Structure (WBS)	Assignee	Status	Planned Start ☐ incl. subtasks*	Planned End ☑ incl. subtasks*	Planned Duration
Trip Diary			[Feb 21, 2020 9:00 AM	Mar 23, 2020 5:00 PM]	22 wdays
Work related to developing the shared i	SAMUEL MINKIN	TO DO	Feb 21, 2020 9:00 AM	Mar 23, 2020 5:00 PM	22 wdays
Daily Summary Feature	SAMUEL MINKIN	TO DO	Feb 24, 2020 9:00 AM	Mar 14, 2020 5:00 PM	16 wdays
Photos, Journal Entries, Events	SAMUEL MINKIN	TO DO	Feb 25, 2020 9:00 AM	Mar 8, 2020 5:00 PM	10 wdays
View Trips	SAMUEL MINKIN	TO DO	Feb 24, 2020 9:00 AM	Mar 11, 2020 5:00 PM	13 wdays
Create Account	SAMUEL MINKIN	TO DO	Feb 22, 2020 9:00 AM	Mar 2, 2020 5:00 PM	6 wdays
Add to current day	SAMUEL MINKIN	TO DO	Feb 28, 2020 9:00 AM	Mar 23, 2020 5:00 PM	17 wdays
Journal Privacy	SAMUEL MINKIN	TO DO	Mar 9, 2020 9:00 AM	Mar 23, 2020 5:00 PM	11 wdays
Server Render	SAMUEL MINKIN	TO DO	Mar 9, 2020 9:00 AM	Mar 23, 2020 5:00 PM	11 wdays

Calendar



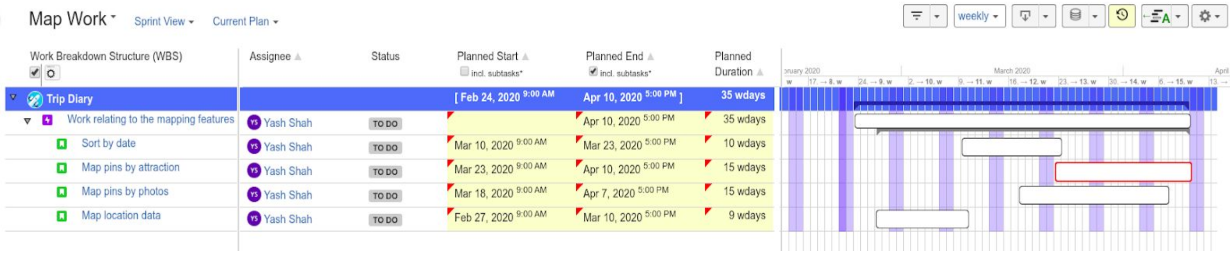
Work Breakdown Structure (WBS)	Assignee	Status	Planned Start ☐ incl. subtasks*	Planned End ☑ incl. subtasks*	Planned Duration
Trip Diary			[Feb 24, 2020 9:00 AM	Mar 23, 2020 5:00 PM]	21 wdays
Work relating to the calendar features	Kinjal Patel	TO DO		Mar 23, 2020 5:00 PM	21 wdays
Planned Event notification	Kinjal Patel	TO DO	Mar 13, 2020 9:00 AM	Mar 23, 2020 5:00 PM	7 wdays
View Day	Kinjal Patel	TO DO	Feb 25, 2020 9:00 AM	Mar 13, 2020 5:00 PM	14 wdays

Journal



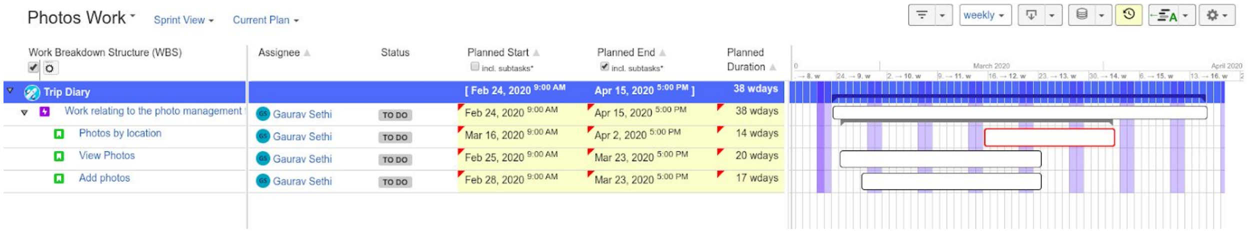
Work Breakdown Structure (WBS) ☑ ○	Assignee	Status	Planned Start □ incl. subtasks*	Planned End ☑ incl. subtasks*	Planned Duration
Trip Diary			[Feb 24, 2020 9:00 AM	Mar 23, 2020 5:00 PM]	21 wdays
Work relating to the journaling features	saz53	TO DO		Mar 23, 2020 5:00 PM	21 wdays
Delete Journal Entry	saz53	TO DO	Mar 7, 2020 9:00 AM	Mar 23, 2020 5:00 PM	11 wdays
Journal Multiple Prompts	saz53	TO DO	Feb 25, 2020 9:00 AM	Mar 7, 2020 5:00 PM	10 wdays
Journal Features	saz53	TO DO	Feb 25, 2020 9:00 AM	Mar 11, 2020 5:00 PM	12 wdays

Map



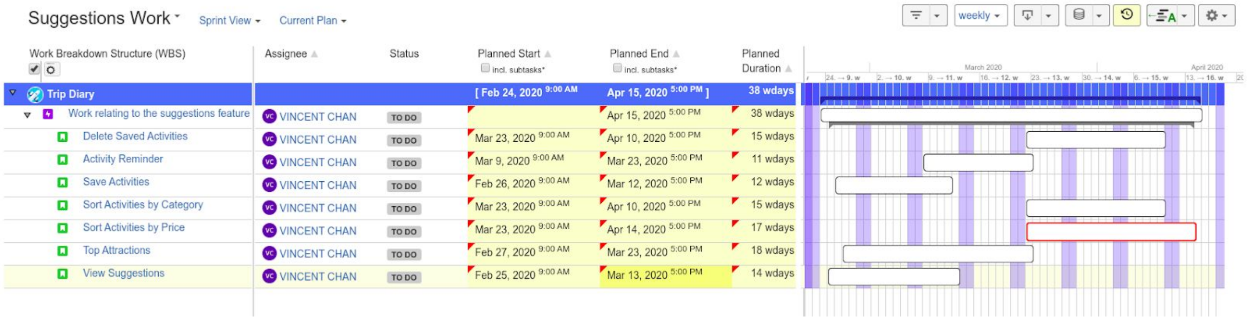
Work Breakdown Structure (WBS) ☑ ○	Assignee	Status	Planned Start □ incl. subtasks*	Planned End ☑ incl. subtasks*	Planned Duration
Trip Diary			[Feb 24, 2020 9:00 AM	Apr 10, 2020 5:00 PM]	35 wdays
Work relating to the mapping features	Yash Shah	TO DO		Apr 10, 2020 5:00 PM	35 wdays
Sort by date	Yash Shah	TO DO	Mar 10, 2020 9:00 AM	Mar 23, 2020 5:00 PM	10 wdays
Map pins by attraction	Yash Shah	TO DO	Mar 23, 2020 9:00 AM	Apr 10, 2020 5:00 PM	15 wdays
Map pins by photos	Yash Shah	TO DO	Mar 18, 2020 9:00 AM	Apr 7, 2020 5:00 PM	15 wdays
Map location data	Yash Shah	TO DO	Feb 27, 2020 9:00 AM	Mar 10, 2020 5:00 PM	9 wdays

Photos



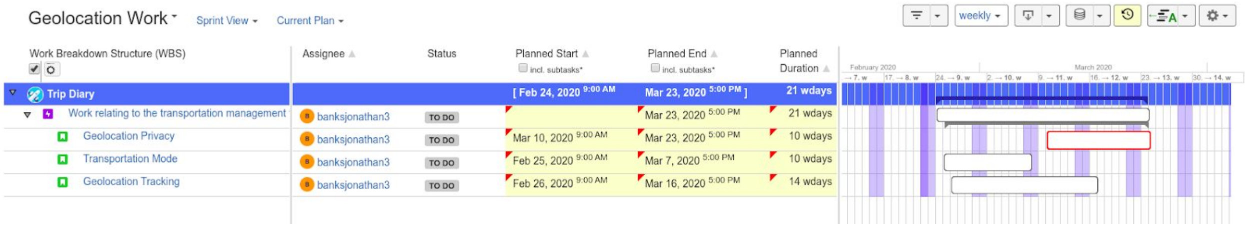
Work Breakdown Structure (WBS) ☑ ○	Assignee	Status	Planned Start □ incl. subtasks*	Planned End ☑ incl. subtasks*	Planned Duration
Trip Diary			[Feb 24, 2020 9:00 AM	Apr 15, 2020 5:00 PM]	38 wdays
Work relating to the photo management	Gaurav Sethi	TO DO	Feb 24, 2020 9:00 AM	Apr 15, 2020 5:00 PM	38 wdays
Photos by location	Gaurav Sethi	TO DO	Mar 16, 2020 9:00 AM	Apr 2, 2020 5:00 PM	14 wdays
View Photos	Gaurav Sethi	TO DO	Feb 25, 2020 9:00 AM	Mar 23, 2020 5:00 PM	20 wdays
Add photos	Gaurav Sethi	TO DO	Feb 28, 2020 9:00 AM	Mar 23, 2020 5:00 PM	17 wdays

Suggestions



Work Breakdown Structure (WBS) ☑ ○	Assignee	Status	Planned Start □ incl. subtasks*	Planned End □ incl. subtasks*	Planned Duration
Trip Diary			[Feb 24, 2020 9:00 AM	Apr 15, 2020 5:00 PM]	38 wdays
Work relating to the suggestions feature	VINCENT CHAN	TO DO		Apr 15, 2020 5:00 PM	38 wdays
Delete Saved Activities	VINCENT CHAN	TO DO	Mar 23, 2020 9:00 AM	Apr 10, 2020 5:00 PM	15 wdays
Activity Reminder	VINCENT CHAN	TO DO	Mar 9, 2020 9:00 AM	Mar 23, 2020 5:00 PM	11 wdays
Save Activities	VINCENT CHAN	TO DO	Feb 26, 2020 9:00 AM	Mar 12, 2020 5:00 PM	12 wdays
Sort Activities by Category	VINCENT CHAN	TO DO	Mar 23, 2020 9:00 AM	Apr 10, 2020 5:00 PM	15 wdays
Sort Activities by Price	VINCENT CHAN	TO DO	Mar 23, 2020 9:00 AM	Apr 14, 2020 5:00 PM	17 wdays
Top Attractions	VINCENT CHAN	TO DO	Feb 27, 2020 9:00 AM	Mar 23, 2020 5:00 PM	18 wdays
View Suggestions	VINCENT CHAN	TO DO	Feb 25, 2020 9:00 AM	Mar 13, 2020 5:00 PM	14 wdays

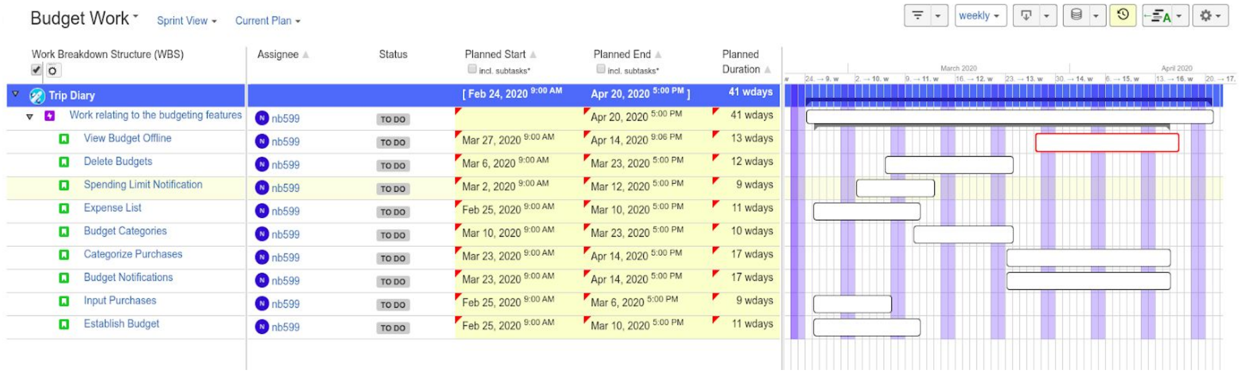
Geolocation



Geolocation Work

Work Breakdown Structure (WBS)	Assignee	Status	Planned Start	Planned End	Planned Duration
Trip Diary			[Feb 24, 2020 9:00 AM	Mar 23, 2020 5:00 PM]	21 wdays
Work relating to the transportation management	banksjonathan3	TO DO		Mar 23, 2020 5:00 PM	21 wdays
Geolocation Privacy	banksjonathan3	TO DO	Mar 10, 2020 9:00 AM	Mar 23, 2020 5:00 PM	10 wdays
Transportation Mode	banksjonathan3	TO DO	Feb 25, 2020 9:00 AM	Mar 7, 2020 5:00 PM	10 wdays
Geolocation Tracking	banksjonathan3	TO DO	Feb 26, 2020 9:00 AM	Mar 16, 2020 5:00 PM	14 wdays

Budget



Budget Work

Work Breakdown Structure (WBS)	Assignee	Status	Planned Start	Planned End	Planned Duration
Trip Diary			[Feb 24, 2020 9:00 AM	Apr 20, 2020 5:00 PM]	41 wdays
Work relating to the budgeting features	nb599	TO DO		Apr 20, 2020 5:00 PM	41 wdays
View Budget Offline	nb599	TO DO	Mar 27, 2020 9:00 AM	Apr 14, 2020 9:06 PM	13 wdays
Delete Budgets	nb599	TO DO	Mar 6, 2020 9:00 AM	Mar 23, 2020 5:00 PM	12 wdays
Spending Limit Notification	nb599	TO DO	Mar 2, 2020 9:00 AM	Mar 12, 2020 5:00 PM	9 wdays
Expense List	nb599	TO DO	Feb 25, 2020 9:00 AM	Mar 10, 2020 5:00 PM	11 wdays
Budget Categories	nb599	TO DO	Mar 10, 2020 9:00 AM	Mar 23, 2020 5:00 PM	10 wdays
Categorize Purchases	nb599	TO DO	Mar 23, 2020 9:00 AM	Apr 14, 2020 5:00 PM	17 wdays
Budget Notifications	nb599	TO DO	Mar 23, 2020 9:00 AM	Apr 14, 2020 5:00 PM	17 wdays
Input Purchases	nb599	TO DO	Feb 25, 2020 9:00 AM	Mar 6, 2020 5:00 PM	9 wdays
Establish Budget	nb599	TO DO	Feb 25, 2020 9:00 AM	Mar 10, 2020 5:00 PM	11 wdays

Project Ownership / Responsibilities

Sub-Problem	Team
Editings & Revisions/Report Combining/Organizing Team Meetings	Kinjal Patel
Calendar/Journal	Kinjal Patel, Samuel Zahner, Jonathan Banks
Map/Photos/Transportation	Gaurav Sethi, Samuel Minkin, Yash Shah
Suggestions/Budget	Vincent Chan, Nisha Bhat

8. References

(1) https://en.wikipedia.org/wiki/FURPS, Used for non-functional requirements
(2) https://christineelder.com/top-10-benefits-of-travel-journaling/, Used to reference actors
(3) https://www.ece.rutgers.edu/~marsic/Teaching/SE/syllabus.html
(4) https://asana.com/resources/gantt-chart-basics, Used to learn gantt chart
(5) https://en.wikipedia.org/wiki/User_story, Used to write user stories
(6) https://en.wikipedia.org/wiki/FURPS, Used to write non-functional requirements