azuria

development group

TraveLayout

System Specification

Prepared For:

Ms. Weltz, Wanderer's Tools

Prepared By:

Kelby Sandvick, Azuria Development Group

Contents

Executive Summary

Ms. Weltz, a representative of Wanderer's Tools came to Azuria Development Group to design, develop, and maintain their second application, TraveLayout. This is the second application Wanderer's Tools will put out, as well as the second application which Azuria Development Group will create for Wanderer's Tools.

In TraveLayout, users will be able to use this application to plan trips, keep track of reservations they have made, and get recommendations for destinations to visit while they are travelling. Users of the system will also be able to post recommendations for other users to see.

Azuria Development Group previously delivered a System Proposal for the TraveLayout application, thus completing the overall design of the program. This document will focus more on the implementation of the application.

This document will be extremely valuable in ensuring that Wanderer's Tools and Azuria Development Group are still on the same page regarding every aspect of the implementation TraveLayout application. It will also provide a technical outline of the application.

1.0 Introduction

1.1 Problem Statement / Project Vision

TraveLayout will be a web application used to plan trips, store reservations, make recommendations based on user's current or inputted location, and export the user's trip itinerary to Wanderer's Tools' first application, TraveLogue.

TraveLayout will be designed as a web application, making it easy for users to plan trips in advance, and to keep track of their itinerary while they are traveling. Wanderer's Tools wants to make this program in order to create a midpoint between paying a travel agent to book a vacation and planning everything by hand.

Wanderer's Tools seeks to create a product for the users who want to be self-sufficient, and to plan their own trips, but still want help finding exactly where they want to go and storing their itinerary neatly in one place. They're also creating a product for customers needing more help in their trip planning. TraveLayout will be usable for users of all abilities.

More information can be found in *section 1.1 Problem Statement,* and *section 1.2 Project Vision and Scope* of the System Proposal.

1.2 System Services

The following are the basic functional requirements of the TraveLayout system. Included at the end of each description is the use-case identification number. More information about each requirement can be found in *section 5.3 Use-Case Descriptions* of the System Proposal.

- Users will be able to create a secure account by inputting data including their first and last name, home state, username, password, and email address. (1)
- If the user has a preexisting account with TraveLogue, they may use those credentials in order to access their TraveLogue account. (2)
- User will be able to input a reservation they have made previously into a standardized form, which the system will save to their personal account. (3)
- User will be able to input information about plans they have made into a standard form similar to the one used to input reservations, and the system will save their input. (4)
- If the user chooses, the application will send a notification to the user for any upcoming reservations or plans for a specific time which have been saved in the system. (5)
- After obtaining consent, TraveLayout will be able to either access the user's location the user can input location manually. (6)
- System will be able to bring up recommendations based on what is near the user's location. (7)
- It will be possible for both moderators and users to input information about destination recommendations by filling out a standard form. (8)
- A moderator will be able to delete user-suggested recommendations from the database if they are deemed inappropriate. (9)

1.3 Nonfunctional Requirements and Design Constraints

The following are the nonfunctional requirements and design constraints which need to be

considered when implementing the TraveLayout system. More information can be found in *section 1.6 constraints* and *section 4.4 Nonfunctional Requirements* of the System Proposal.

- Users must be able to access the system on mobile devices, laptops, and desktop computers using the browser of their choice.
- The interface will be similar to TraveLogue, logically laid out, and with simple-to-learn functionality to heighten accessibility.
- Users will be able to gain access to their information by logging in with valid username and password, which are encrypted and saved within the system.
- Extra security measures will need to be implemented to ensure that users will be safe, and that their location is kept safe
- TraveLayout must run efficiently and effectively and be easy to maintain.

1.4 System Evolution

While Azuria plans to deliver a fully functional program in the first version, there are some aspects of the functionality which may be pushed to a future version for the sake of getting a functional program available to customers as quickly as possible. The initial version of the TraveLayout system will only be compatible with planning within the United States. The next versions of the system, however, will be compatible with other countries, expanding first to the rest of the Americas, then over to Europe, and beyond.

More information about planned and recommended upgrades to the system can be found in *section 6.1 Planned Upgrades* of the System Proposal.

1.5 Document Outline

This document contains the following five sections:

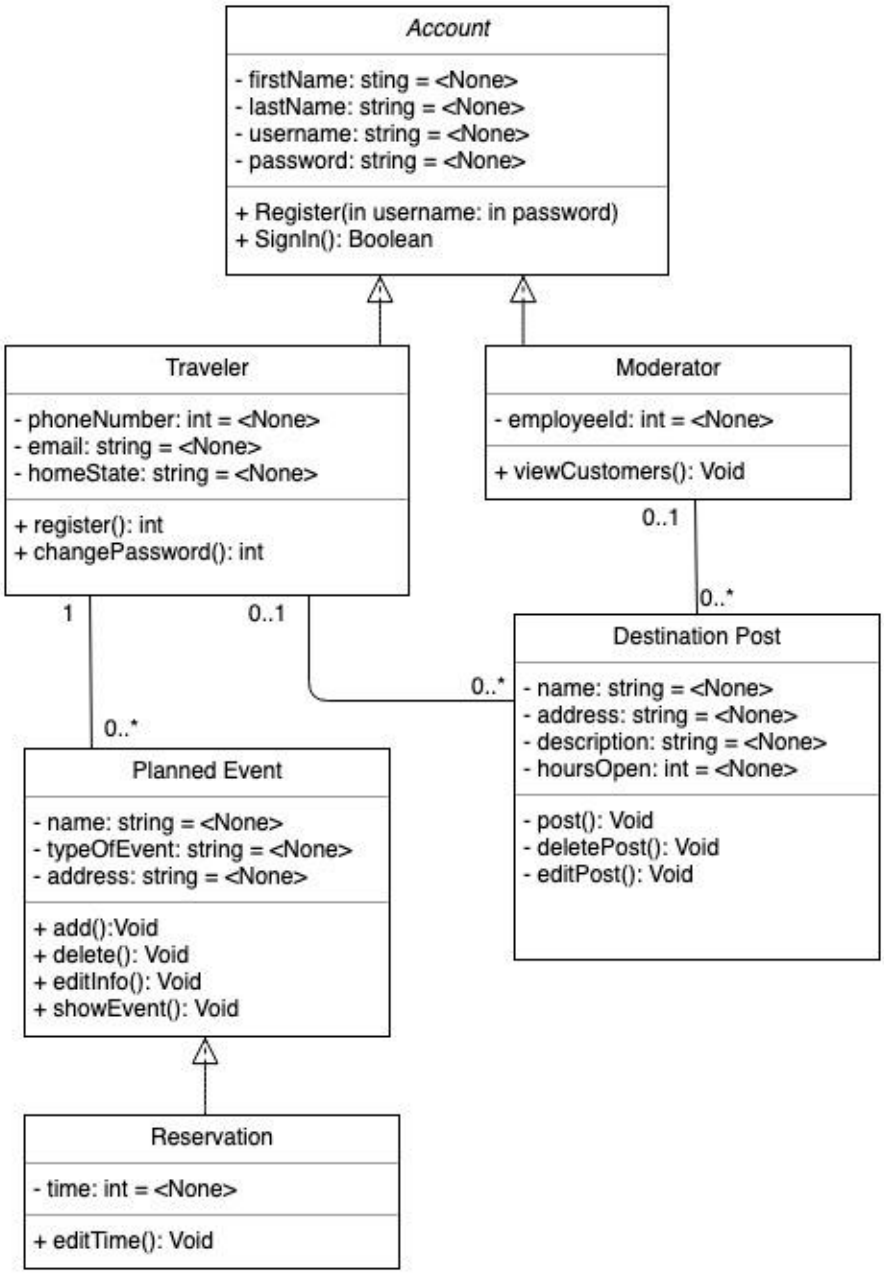
- *Introduction*: An overview of the current project, including its requirements, and plans for additions in the future
- *Structural Model*: A class diagram with associated metadata which shows how objects and stored data will interact with TraveLayout
- *Architecture Design*: The infrastructure model, hardware and software requirements and the security plan for the system
- *Appendices*: A bibliography and all supporting references as well as all supporting documentation

2.0 Structural Model

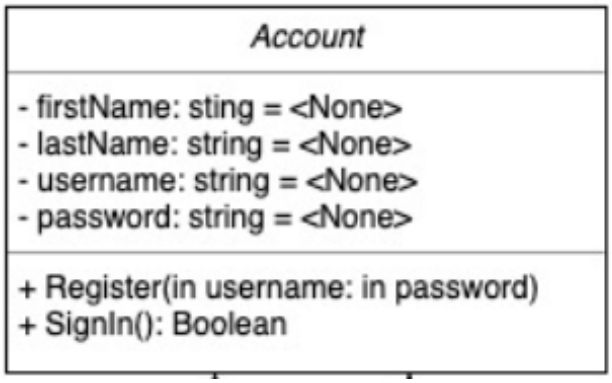
2.1 Introduction

This section contains the class diagrams of TraveLayout, and the relationships between the classes shown. The class diagram shows the relationships between the classes in the system and is modeled in UML format. *Section 2.3 Metadata* contains more detailed information about each class including class attributes and operation details.

2.2 Class Diagram



2.3 Metadata



- Description: Abstract class for all accounts of each type in system
- Visibility: Public
- Is Abstract: Yes

Attributes

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
firstName	First name of user	string	no	yes	private	1	<None>
lastName	Last name of user	string	no	yes	private	1	<None>
username	Unique username	string	no	yes	private	1	<None>
password	Secure password	string	no	yes	private	1	<None>

Operations

Name	Description	Is Query	Is Polymorphic
Register	New user register for account	No	No
SignIn	Current user sign into account	No	No

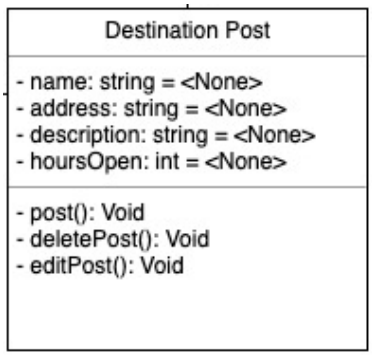
Processing Outlines

Register:

- User will input a username and password, and confirm password
- System will validate username and password
- User will automatically be logged into new account

Sign In:

- User will input their username and password
- System will validate username and password
- User will be logged into account



- Description: Represents a destination recommendation post
- Visibility: Public
- Is Abstract: No

Attributes

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
name	Name of destination	string	no	yes	private	1	<None>
address	Address of destination	string	no	yes	private	1	<None>
description	Short description of destination	string	no	yes	private	1	<None>
hoursOpen	Operating hours of destination	int	no	yes	private	1	<None>

Operations

Name	Description	Is Query	Is Polymorphic
post	Make a recommendation Post	No	No
deletePost	Delete a recommendation Post	No	No
editPost	Edit a recommendation Post	No	No

Processing Outlines

post:

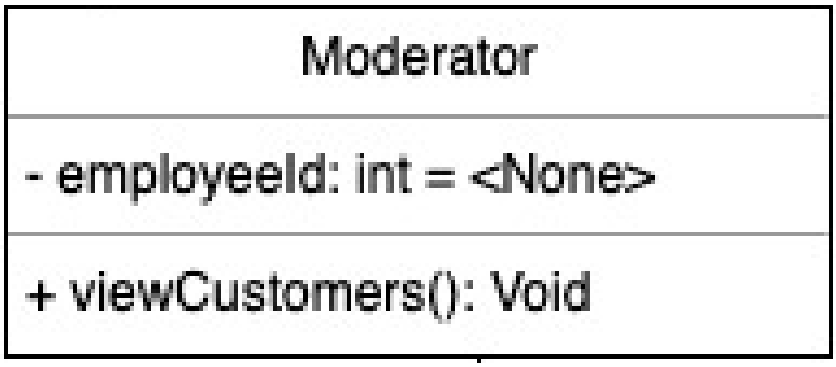
- User will select to post recommendation
- User will input all required information
- System will validate inputs
- Destination recommendation will be added to database

deletePost:

- User will select to delete posted recommendation
- Permissions will be verified that user is allowed to delete specific post
- If verified, post will be deleted from database
- Else, user will be informed that they may not delete post

editPost

- User will select to edit posted recommendation
- Permissions will be verified that user is allowed to edit specific post
- If verified, user may edit post information
- System will validate inputs and post to database
- Else, user will be informed that they may not edit post



- Description: Represents a class of user called moderators
- Visibility: Public
- Is Abstract: No

Attributes

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
employeeId	Employee Identification Number	int	no	yes	private	1	<None>

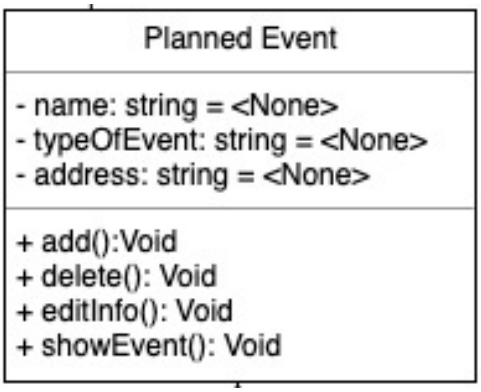
Operations

Name	Description	Is Query	Is Polymorphic
viewCustomers	View a list of all current users of the system	No	No

Processing Outlines

viewCustomers:

- User will select to view all current users
- User permissions will be verified if they are allowed to view customers
- If permission granted, system will display list of current customers
- If permission is denied, system will inform user



- Description: Represents a user-inputted event
- Visibility: Public
- Is Abstract: Yes

Attributes

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
name	Name of Event	string	no	yes	private	1	<None>
typeOfEvent	Type of event	string	no	yes	private	1	<None>
address	Address of event	string	no	yes	private	1	<None>

Operations

Name	Description	Is Query	Is Polymorphic
add	Add a new planned event	No	No
delete	Delete an event	No	No
edit	Edit details of a planned event	No	No
showEvent	Show all information about a planned event	No	No

Processing Outlines

add:

- User will select to add a new planned event
- User will input all required information
- System will validate inputs
- Event will be added to user's stored events

delete:

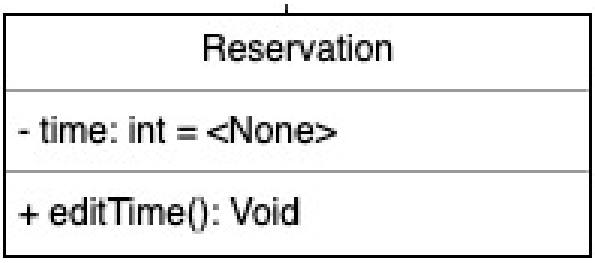
- User will select to delete a planned event
- Event will be deleted

edit:

- User will select to edit a planned event
- User will input all information to be edited
- System will validate inputs
- New information will be saved to stored event

showEvent:

- User will select to show the details of a planned event
- System will display all inputted details of the event



- Description: Represents a reservation, inherits attributes form *Planned Event*
- Visibility: Public
- Is Abstract: Yes

Attributes

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
time	Time of reservation	string	no	yes	private	1	<None>

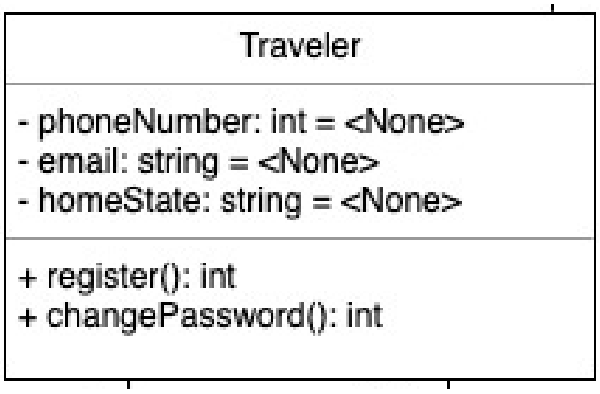
Operations

Name	Description	Is Query	Is Polymorphic
editTime	Set the time of a reservation	No	No

Processing Outlines

editTime:

- User will select to add add or edit the time of a reservation
- System will validate input
- Time and reservation will be added to user's stored events



- Description: Represents a user-inputted event
- Visibility: Public
- Is Abstract: Yes

Attributes

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
phoneNumber	Phone number of user	string	no	yes	private	1	<None>
email	Email address of user	string	no	yes	private	1	<None>
homeState	Home state of user	string	no	yes	private	1	<None>

Operations

Name	Description	Is Query	Is Polymorphic
register	Register for a new account	No	No
changePassword	Change the password attached to account	No	No

Processing Outlines

Register:

- User will input a username and password, and confirm password
- System will validate password and confirmed password are identical
- System will validate username and password
- User will automatically be logged into new account

changePassword:

- User will input their username and password
- System will validate username and password
- If username and password are validated, allow user to change password
- Logout user to login again with new password
- Else ask user to login again

3.0 Architecture Design

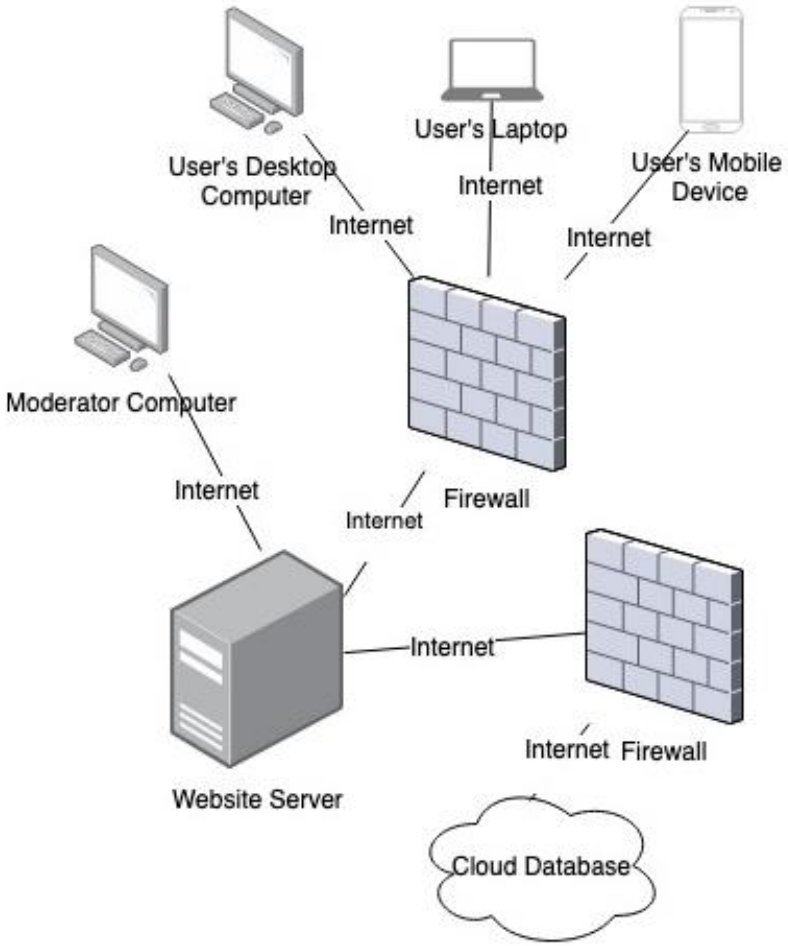
3.1 Introduction

Included in this section are two infrastructure models of the TraveLayout system. Also outlined is the necessary hardware and software for operation and an overview of the initial security plan for the system.

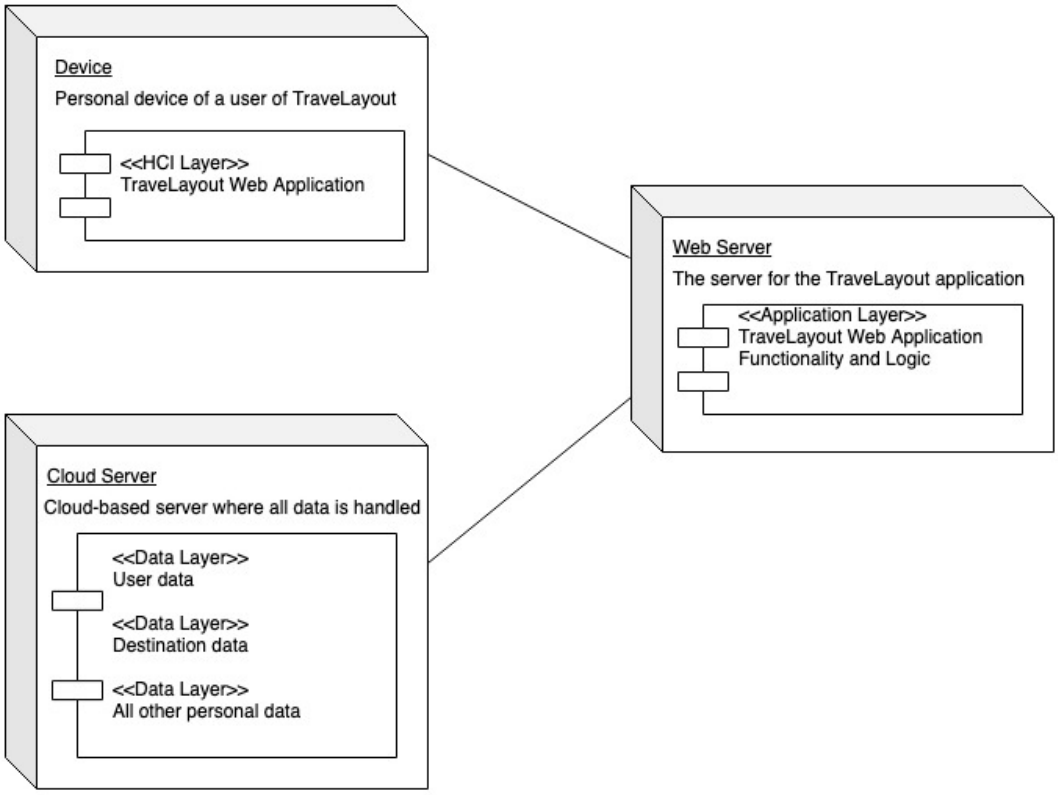
The infrastructure models include an architecture overview diagram showing system's nodes and their relationships with each other. Also included is a deployment diagram which shows the distribution of the information and workload across the system. As shown in the architecture overview diagram, TraveLayout will be using a 3-tier client-server system. All presentation logic will be shown on the user's computers, the website server will run all system logic and functionality, and all system data will be stored in the cloud.

3.2 Infrastructure Model

Architecture Overview Diagram



Nodes and Artifacts Diagram



3.3 Hardware and Software Requirements

Required Hardware Components

- Web Application Server: required to support the TraveLayout website. The same server as TraveLogue may be used temporarily if required while system is still growing in popularity.
- Cloud Database: will be used to store all data of the system not immediately required for functionality. This includes user data and the destination recommendation database. TraveLogue's current cloud server may be used temporarily if necessary.
- Moderator Computers: employees of Wanderer's Tools selected as moderators for the destination recommendations will need either a desktop or laptop computer to access the TraveLayout system.
- User Devices: in order to use the TraveLayout system, the user must have access to some sort of computer.

Required Software Components

- Windows OS: recommended for all moderators and users to access the system.
- Anti-Virus Software: no specific one is preferred, however implementing an antivirus system is imperative.
- Microsoft SQL: necessary for storing all data in the system.

3.4 Security Plan

For Wanderer's Tools and the TraveLayout system, there are some threat which need to be controlled.

All physical threats to the system and company will be handled as follows. In terms of natural disasters such as fire, storm, earthquake, flood, etcetera, the website server and the cloud server are vulnerable. Backups of all vital data will be kept, and Azuria Development will layout a full disaster recovery plan in future documents. All business standard safety measures will also be put into place including an anti-fire system and door and item locks and alarms where applicable. There will be fail safes in place in the case of a potential hardware malfunction. These will include surge protectors on all outlets, as well as backup generators available for all servers in case of power outage. Similarly, insurance will be purchased for all hardware related to and used by TraveLayout.

All virtual attacks to the system will be handled as follows. In order to deter hackers attempting to steal the personal information of our users, anti-virus software will be installed on all company computers, and strong firewalls will be put in place within the system between the client and web server and between the web server and the cloud server. Since the application can specifically retrieves the user's current location, the security must be particularly strong in order to keep users safe from physical attacks as well as keep their homes and personal belongings safe.

Users will be encouraged to secure their devices with passwords in case of theft, however there is not a lot that Wanderer's Tools may do in order to protect the physical safety of their personal devices. TraveLayout will be a system which is accessible on multiple devices, so if a user's personal device is somehow damaged or lost, their TraveLayout account will not be affected in any way.

For virtual safety, users must be informed of the risk of hackers attempting to access their account and gain their information. Users will be encouraged to choose a secure password in order to make sure that it is harder for their account to be gotten into by an unauthorized party. Users will also be shown information on how to protect their account from hackers. Users will specifically authorize when the system may access their location, and will be informed when the system is accessing their location so they will know if it is being done at a time they did not authorize.

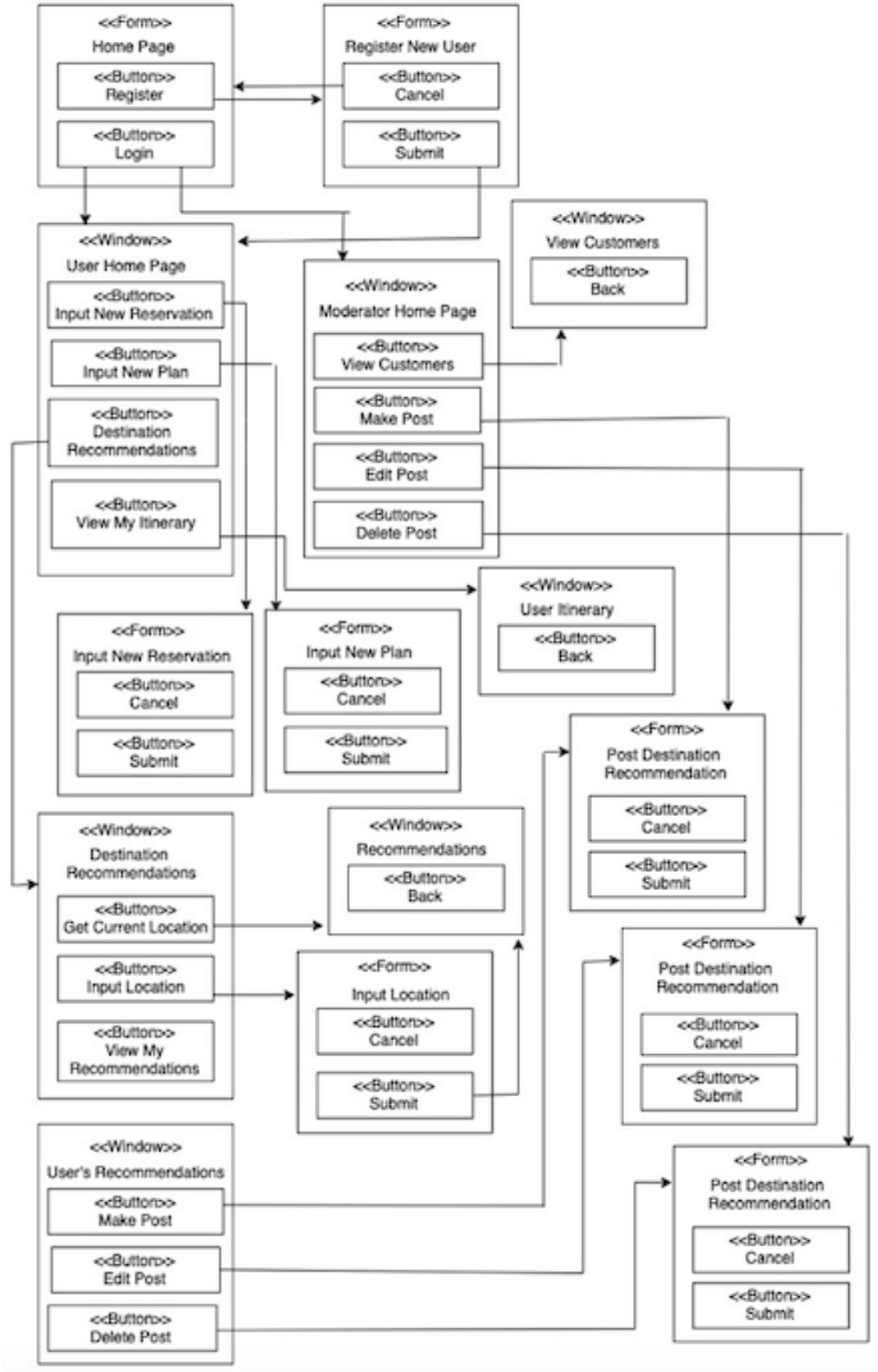
4.0 Requirements Definition

4.1 User-Interface Requirements and Constraints

The following sections contain basic diagrams of what the graphic user interface (GUI) of TraveLayout will look like. The design is guided by a goal of creating a clear, easy to learn interface for users. A main goal of the design of this system is to make it similar to TraveLogue. Since the system is not overly complicated in its functionality, the goal of the GUI is to be simple and clean. In order to do this, we will be following basic design rules such as the "three-click-rule", implementing meaningful metaphors, and including all possible affordances.

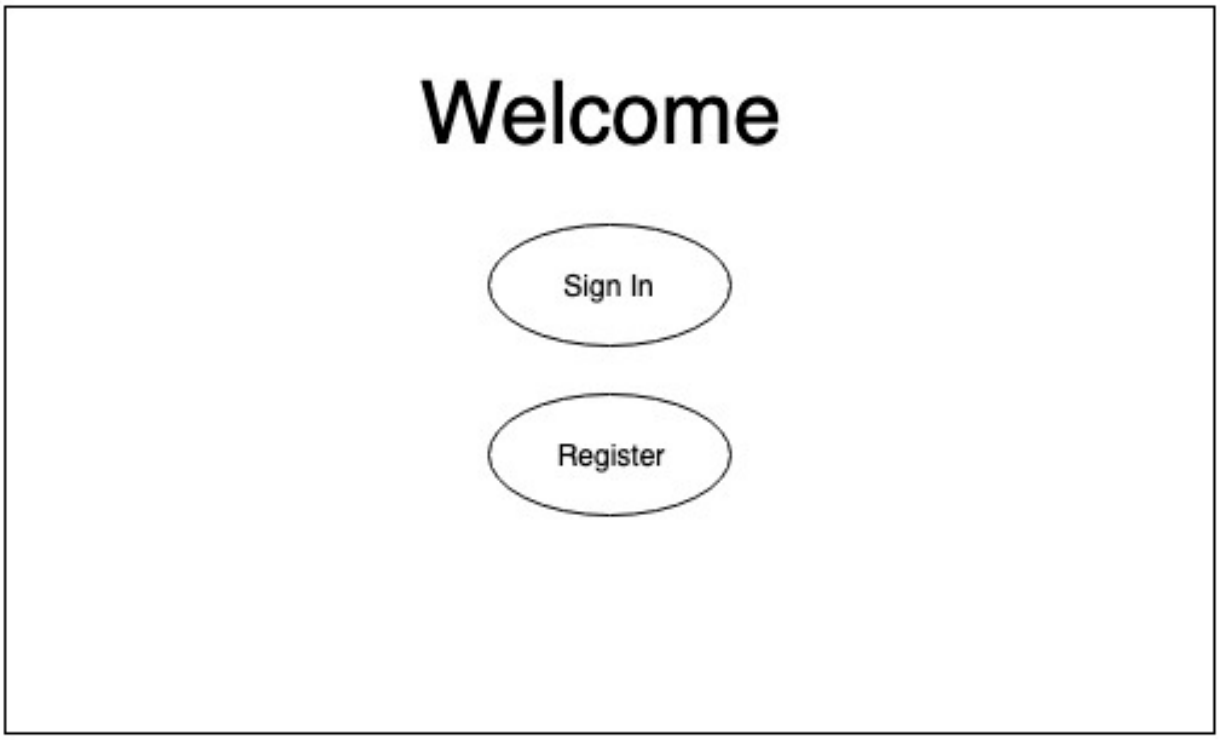
Beginning on the next page are diagrams including the *Window Navigation Diagram* which shows what the flow between different screens in the program will look like, the *User-Interaction Design* which displays the content of each of the screens, and the *Reports and Formal Output Design* which displays how reports will look for any printed outputs.

4.2 Window Navigation Diagram

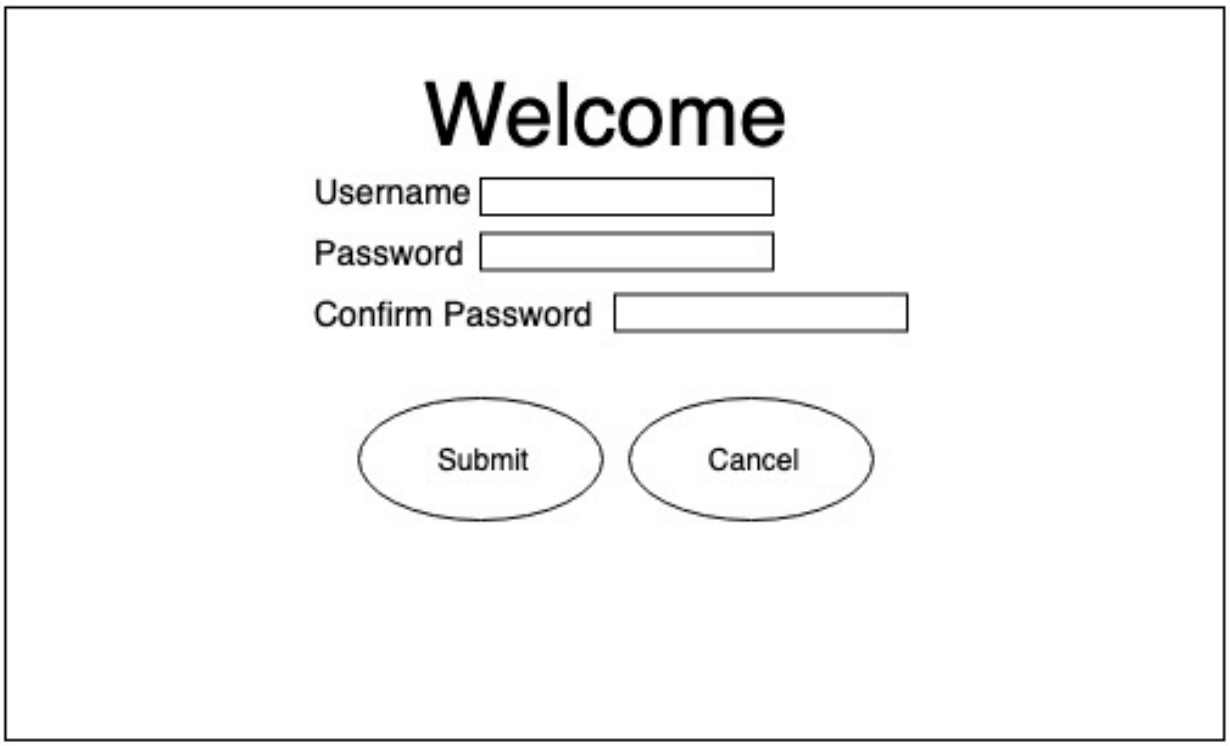


4.3 Forms: Screen / User-Interaction Design

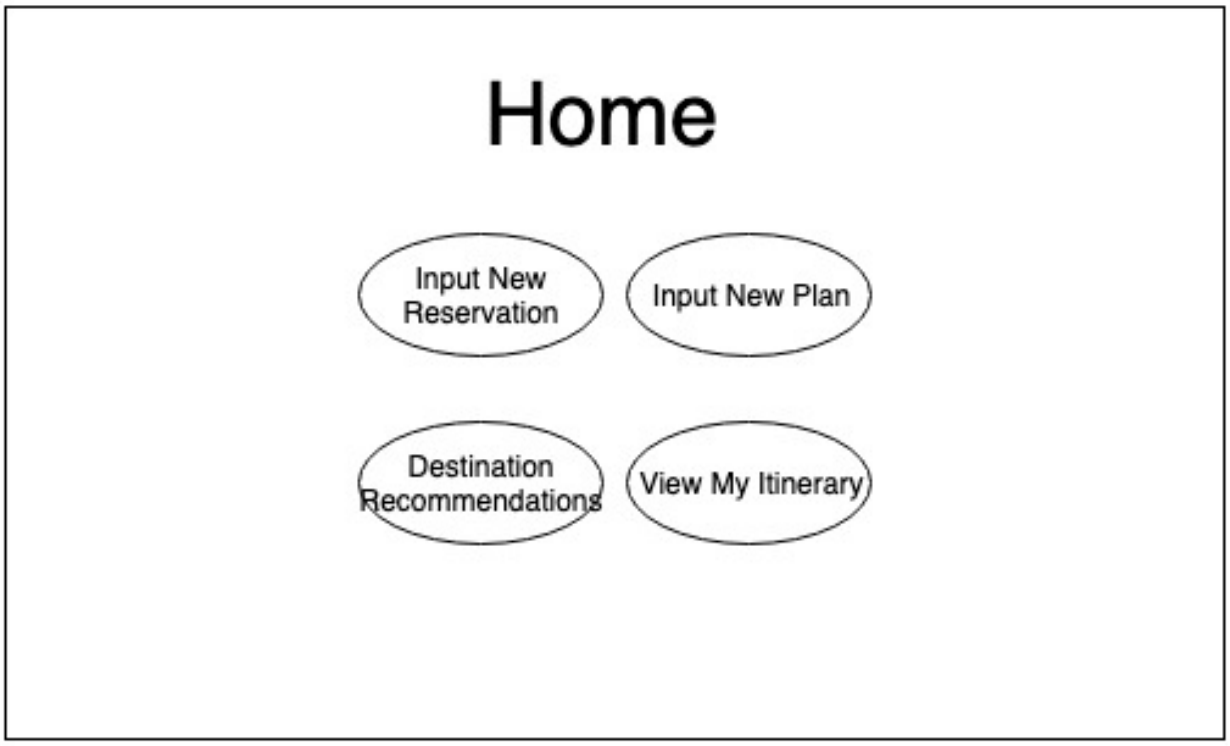
Home Page



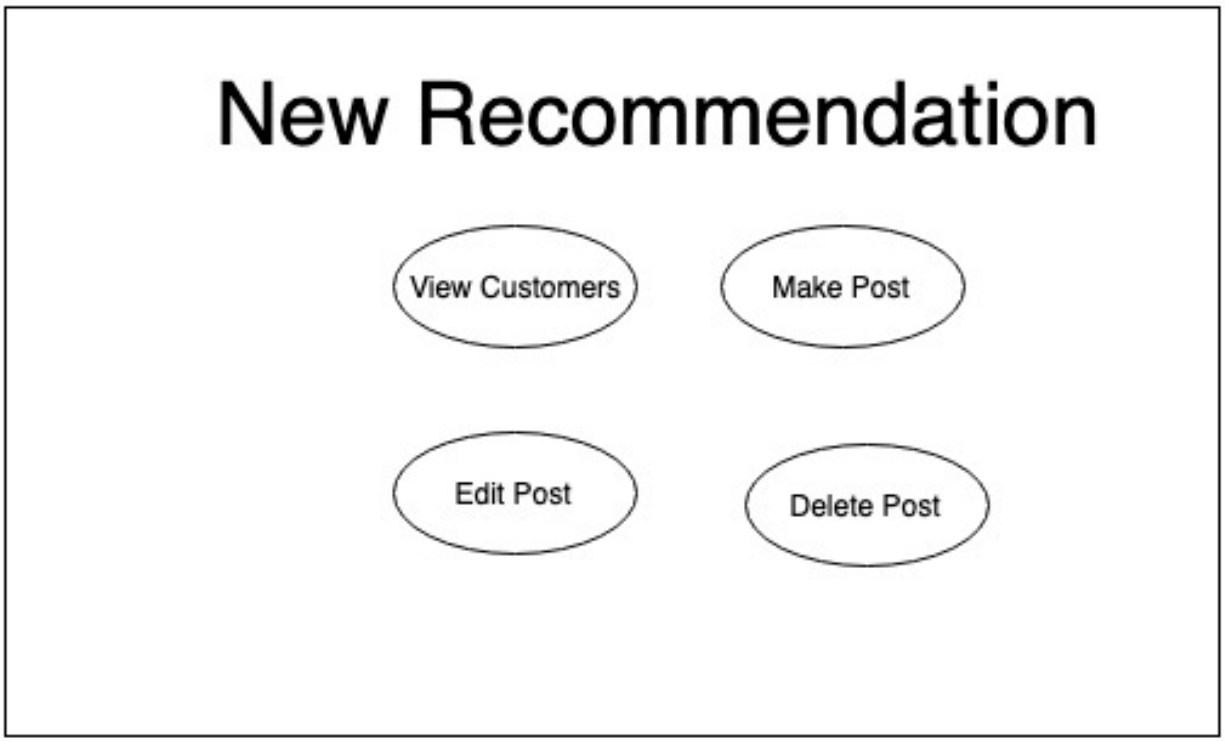
Register New User



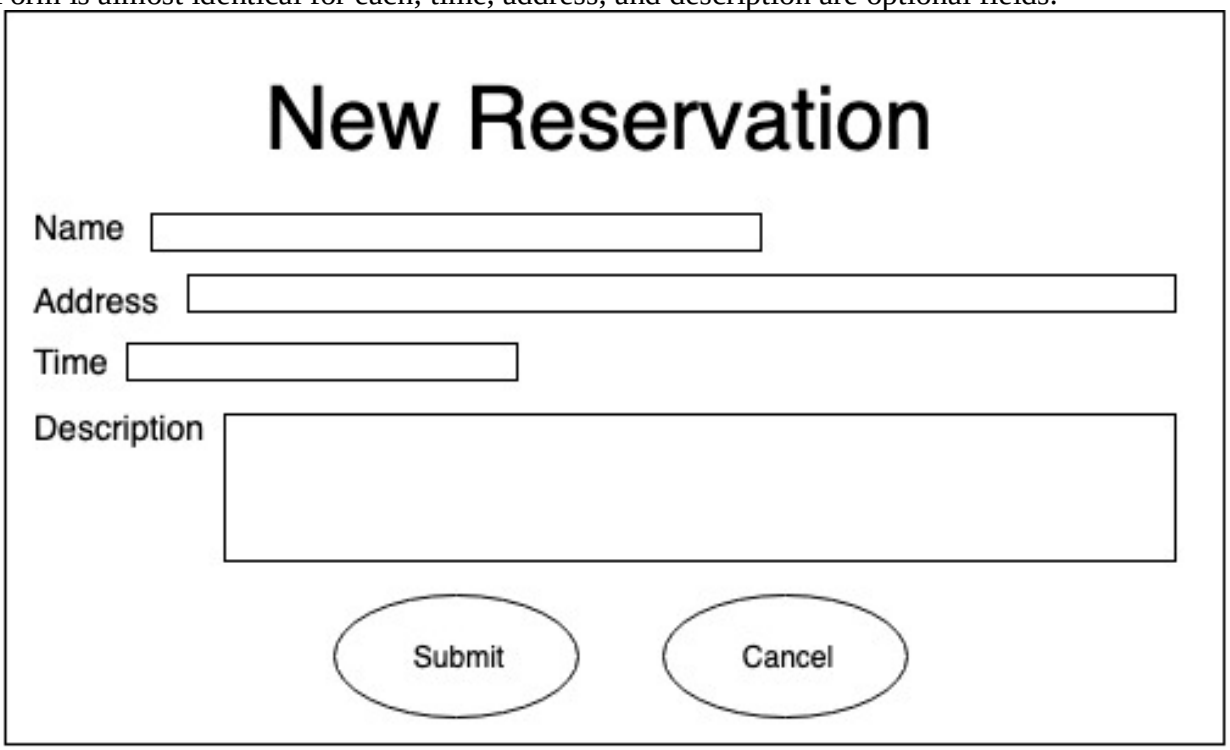
User Home Page



Moderator Home Page

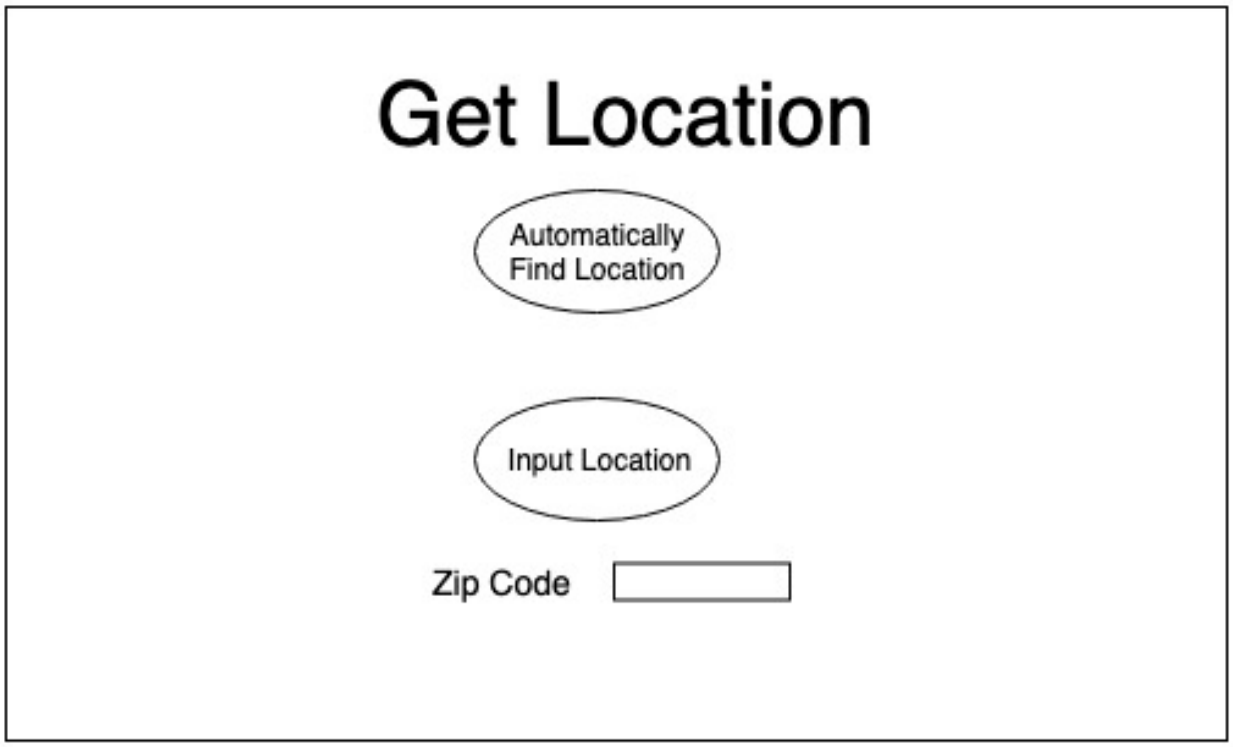


Input New Reservation/Input New Plan
Form is almost identical for each; time, address, and description are optional fields.

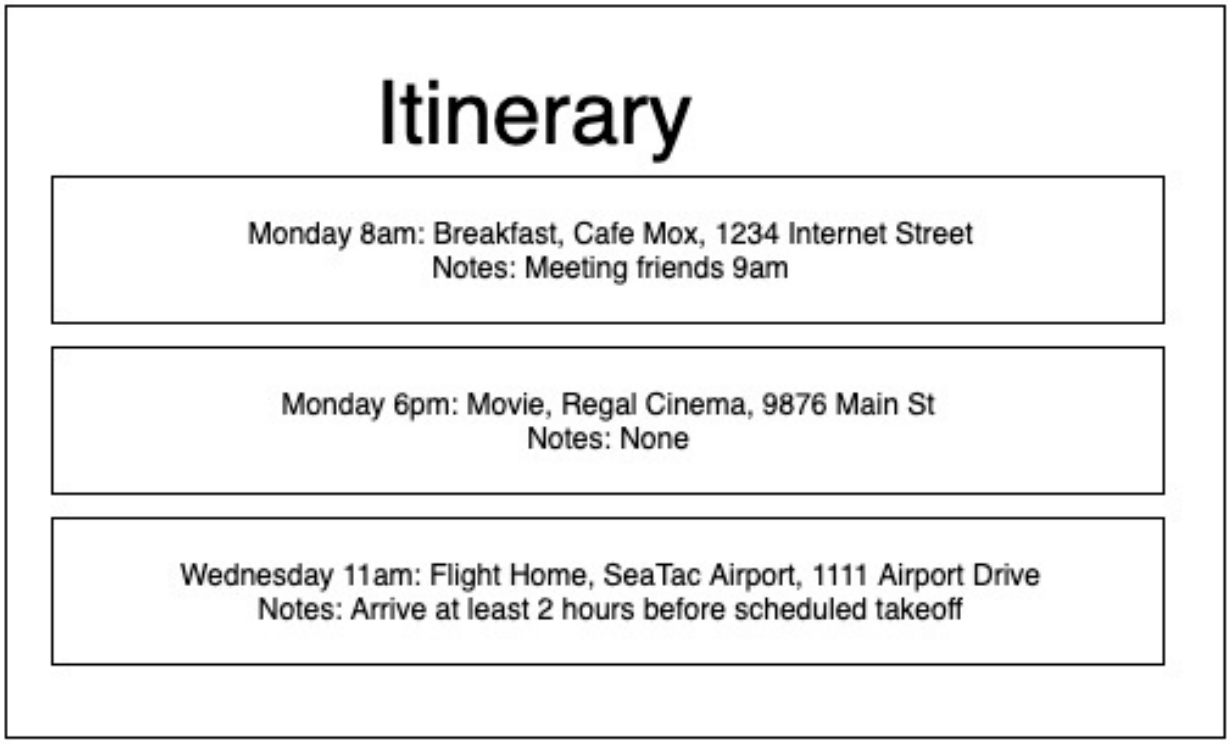


Post Destination Recommendation
User and Moderator versions look essentially identical. To edit a post, the same form is displayed, and input fields are editable.

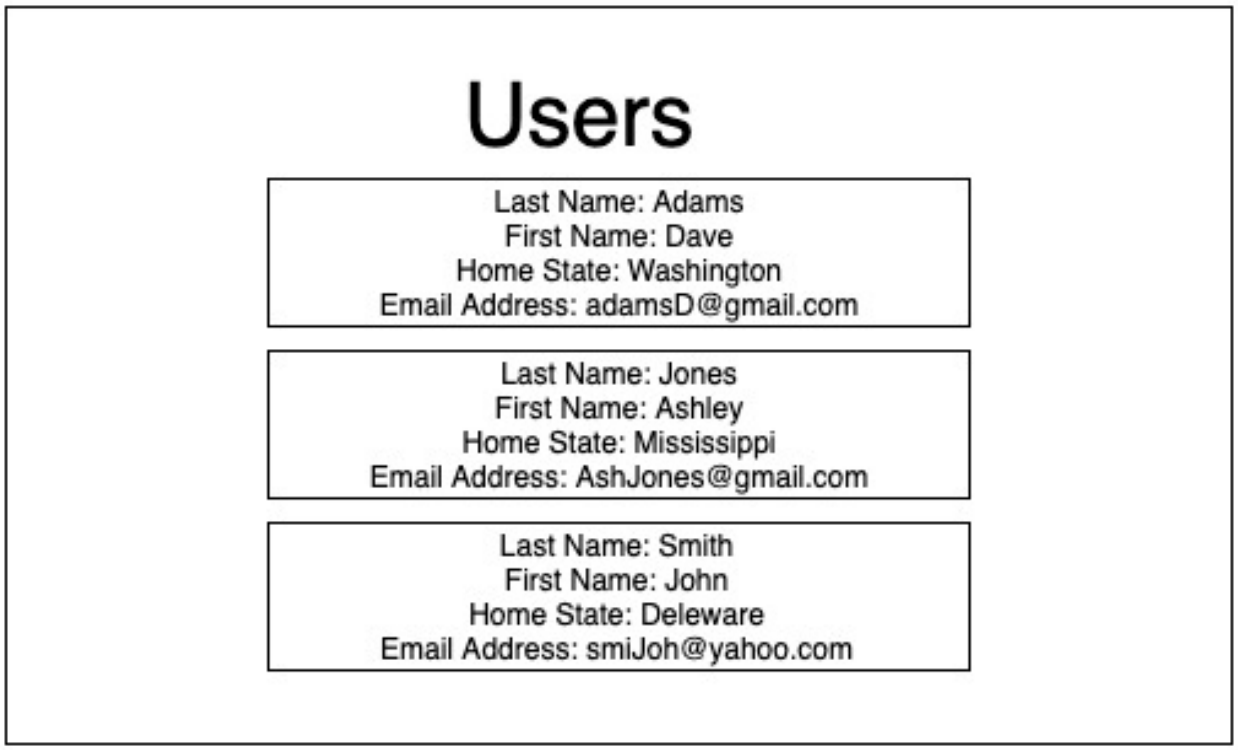
User Location Getting



User Itinerary



View Customers



5.0 Appendices

5.1 Bibliography / References

"Cuisine By Car Computer System." *Canvas*, 7 Dec. 2018, Cuisine By Car Design 1.

"Cuisine By Car Mase System Specification." *Canvas*, Cuisine By Car Design 2.

Dennis, Alan, et al. *System Analysis & Design: an Object-Oriented Approach with UML*. 5th ed., Wiley, 2015.

"Food For Everyone (FFS) System Specification." *Canvas*, 5K Foods SPEC-3.

"Food, Resources, Expenses and Distribution System (FREDS) System Specification." *Canvas*, 5K Foods SPEC-2.

"Pfeiffer, William S. *Pocket Guide to Technical Communication*. 5th ed., Prentice Hall, 2011.

"System Specification FIMS." *Canvas*, 5K Foods SPEC-1.

Weltz, Elaine (2019). Systems Design, various lectures [PowerPoint slides/Word Documents]. Retrieved from Professor Weltz and http://canvas.spu.edu

5.2 Supporting Documentation