Systems Specification for

Wanderer’s Tools

Prepared for:

Ms. Elaine Weltz, Wanderer’s Tools

Prepared by:

Cameron Moe, Software Developer

Prestige Systems

December 2, 2018

**TABLE OF CONTENTS**

**TABLE OF CONTENTS** 2

EXECUTIVE SUMMARY 3

1.0 INTRODUCTION 4

1.1 Problem Statement and Project Vision 4

1.2 System Services 4

1.3 Nonfunctional Requirements and Design Contraints 5

1.4 System Evolution 6

1.5 Document Outline 6

2.0 STRUCTURAL MODEL 7

2.1 Introuction 7

2.2 Class Diagram 7

2.3 Metadata 8

3.0 ARCHITECTURE DESIGN 22

3.1 Introduction 22

3.2 Infrastructure Model 22

3.3 Hardware and Software Requirements 23

3.4 Security Plan 23

4.0 USER-INTERFACE 24

4.1 User-Interface Requirements and Constraints 24

4.2 Windows Navigation Diagram 24

4.3 Forms: Screen/User-Interaction Design 25

5.0 APPENDICES 29

5.1 Bibliography 29

**EXECUTIVE SUMMARY**

Ms. Weltz and Wanderer’s Tools has hired Prestige Systems to design a system for their new travel application. Prestige Systems is developing Travel Assist System (TAS) to help traveler’s organize all aspects of their travel, including media uploads, expenses, and creation of a travelogue to share their travels with others.

In the Systems Proposal, Prestige Systems considered feasibility and requirements and determined there was a limited amount of risk, and that the project is feasible. After approval from Wanderer’s Tools, Prestige Systems is moving to the development stage of TAS. The rest of this document includes an overview of the system, with a structural model, infrastructure model, security plan and UI planning.

1. **INTRODUCTION**

**1.1 Problem Statement and Project Vision**

Currently, it is difficult to keep all the information that goes into a trip together, as well as document the experience while it is ongoing. Prestige Systems is developing an application will allow travelers to use various electronic devices to capture information, pictures, receipts and thoughts about the trip, and easily organize this information on the go with an easy to use application.

Prestige Systems resolves to develop a Travel Journal system for Wanderer’s Tools called Travel Assist System (TAS) to accomplish the tasks laid out by the company. TAS will allow users to keep a written journal with pictures, videos, notes, purchases, and any other useful travel information in one easily accessible space. TAS will allow users to login on multiple devices and upload all data to the cloud when there is either internet access or data reception. TAS will not log any travel information like miles traveled or location pictures are taken from, as it is outside the scope of this project.

Stakeholders interested in the completion of this project include people who want to quickly and easily document their travels, people in tourism industries that want to keep track of frequently visited venues in their area, travel agents who want to learn about popular destinations, Ms. Weltz and Wanderer’s Tools who want a profitable application, and the developers at Prestige Systems who want to see the application completed to the client’s standards, with a manageable timeframe.

**1.2 System Services**

Below is a list of the functional requirements for TAS. For more information, refer to section 4 and 5 of the Systems Proposal.

**User Account**

* The user must create an account using a username, an email or phone number, and a password. Refer to Section 5.3 Use Case 4
* The user will create a username that can be used to login on all subsequent devices. Refer to Section 5.3 Use Case 4
* TAS should allow optional notifications to be turned on so that at the end of each travel they remind the user to update their travel log with the day’s activities. Refer to Section 5.3 Use Case 4

**Media upload**

* User will select to upload either a picture, video, map or text document. Refer to Section 5.3 Use Case 2
* User will be able to document the capture of the media upload with a calendar day. Refer to Section 5.3 Use Case 2
* TAS will organize the media uploads by date for the user. Refer to Use Case 2
* TAS could allow uploading of images straight from some smart cameras. This will likely come in a future update.

**Journal Entry**

* TAS must allow users to write and save journal entries on different styles of virtual paper. Refer to Section 5.3 Use Case 1
* The user should be able to take a picture of a written journal and be able to upload it as a journal entry. Refer to Section 5.3 Use Case 1

**Expenses Page**

* TAS should have an expenses page with a running total that will calculate total expenses, taxes, and any reimbursements for employees on work travel. Refer to Section 5.3 Use Case 1
* User will be able to create a new payment event, and add date, location, and amount paid. Refer to Section 5.3 Use Case 1
* Event must be able to be viewed alongside the journal, as well as in a separate expenses tab. Refer to Section 5.3 Use Case 1

**Travelogue Creation**

* TAS should contain simple editing tools to enable users to create a travelogue to showcase the highlights of the trip. This should be able to be done offline. Refer to Section 5.3 Use Case 5

**1.3 Nonfunctional Requirements and Design Constraints**

Listed below are the most important constraints, nonfunctional requirements, and feasibility concerns for TAS.

* TAS must have a secure 2-step login system on all devices so personal data cannot easily be compromised.
* TAS must allow upload of all media to the cloud to be accessed by other devices
* TAS must be compatible between PC, Mac, iOS and Android devices.
* TAS must be able to stay small in storage size so data can be stored offline until an internet connection is accessible.
* TAS must be user-friendly for people with every level of technical experience.
* TAS must allow the user to quickly input information without having to navigate through unnecessary tabs or misdirection.

**1.4 System Evolution**

Prestige Systems will continue to work on TAS after the initial version is released. These subsequent versions will improve on the existing features of the system, as well as adding some new luxury features, but will not be essential for use right away. Some content to look forward to in the future include the following:

* The ability to upload pictures straight from smart cameras to the application.
* Provide analytics about number and location of user accounts.
* Provide analytics on most popular travel destinations.
* Provide analytics on most expensive travel destinations.
* Implement an auto-correct feature for journal entries that can be used offline.

**1.5 Document Outline**

Listed below is the outline for the rest of the document:

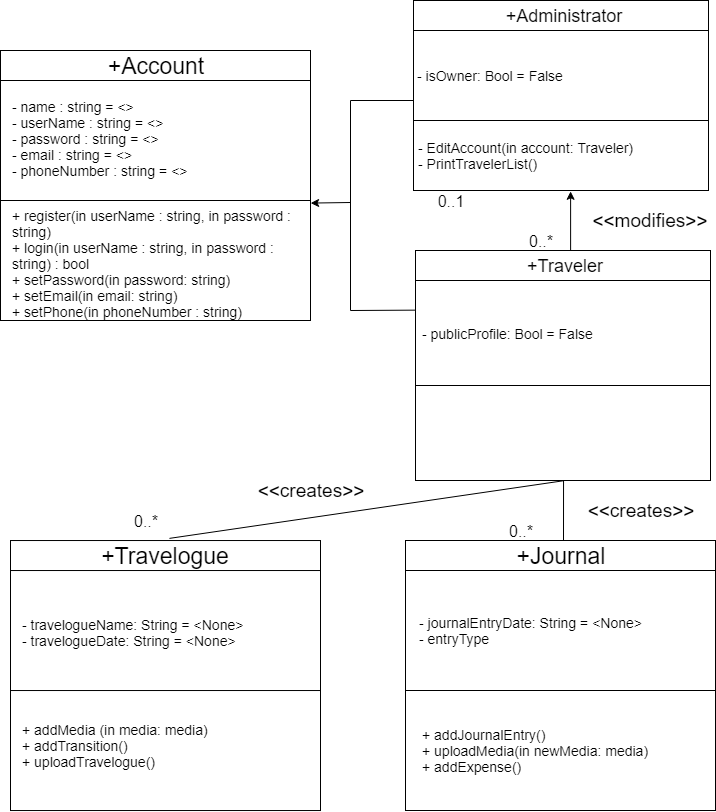
* **Structural Model:** This section contains a class diagram, and more detailed information on each class which shows how the various objects in TAS will interact with one another
* **Architecture Design:** This section contains deployment diagrams, Hardware/Software requirements, and system security design describe the architecture of the system.
* **User-Interface:** This section contains images that model what the UI should look like on completion of the system
* **Appendices:** This section contains the bibliography

1. **STRUCTURAL MODEL**

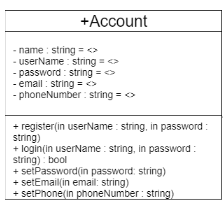
**2.1 Introduction**

This section provides a class diagram for TAS. Section 2.2 shows an overview of the diagram, and section 2.3 goes into more detail for each of the classes.

**2.2 Class Diagram**

****

**2.3 Metadata**



Description: Abstract class for various account types

Visibility: Public

Is Abstract: Yes

**Attributes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Data Type | Is Derived | Is Read Only | Visibility | Multiplicity | Default Value |
| name | User’s name | String | No | Yes | Private | 1 | <None> |
| userName | User’s UserName | String | No | Yes | Private | 1 | <None> |
| password | User’s Password | String | No | Yes | Private | 1 | <None> |
| email | User’s email | String | No | Yes | Private | 1 | <None> |
| phoneNumber | User’s Phone Number | String | No | Yes | Private | 1 | <None> |

**Operations:**

|  |  |
| --- | --- |
| Name | register |
| Description | User can register for an account that can be used for an administrator or traveler. |
| Return Type | bool |
| Parameters | Username  Direction: In  String  Default: None  Password  Direction: In  String  Default: None |
| Visibility | Public |
| Scope | Instance |
| Is Query | No |

**Processing Outline:**

User inputs username and password

If Valid

Allow user registration and add to user database

Else

Inform user of registration fail

Return False

|  |  |
| --- | --- |
| Name | login |
| Description | User can login to their account using a valid username and password |
| Return Type | None |
| Parameters | Username  Direction: In  String  Default: None  Password  Direction: In  String  Default: None |
| Visibility | Public |
| Scope | Instance |
| Is Query | No |

**Processing Outline:**

User inputs username and password

If Valid

Allow user to login and take them to their home page

Else

Inform user of registration fail

|  |  |
| --- | --- |
| Name | setPassword |
| Description | User can change their password and set it to something new if they choose, or forget it. |
| Return Type | None |
| Parameters | Password  Direction: In  String  Default: None |
| Visibility | Public |
| Scope | Instance |
| Is Query | No |

**Processing Outline:**

User inputs password

If valid password

Set User’s password to the new password

Else

Report invalid password

|  |  |
| --- | --- |
| Name | setEmail |
| Description | User can change their email after creating an account and set it to something new. |
| Return Type | None |
| Parameters | Email  Direction: In  String  Default: None |
| Visibility | Public |
| Scope | Instance |
| Is Query | No |

**Processing Outline:**

User inputs email

If valid email

Set User’s email to the new email

Sends confirmation message to new email

Else

Report invalid email

|  |  |
| --- | --- |
| Name | setPhone |
| Description | User can change their phone number after creating an account and set it to something new. |
| Return Type | None |
| Parameters | phoneNumber  Direction: In  String  Default: None |
| Visibility | Public |
| Scope | Instance |
| Is Query | No |

**Processing Outline:**

User inputs phone number

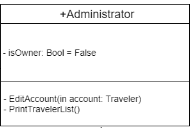
If valid phone number

Set User’s phone number to new phone number

Sends confirmation text to new phone number

Else

Report invalid phone number



Description: Class for administrator account

Visibility: Public

Is Abstract: No

**Attributes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Data Type | Is Derived | Is Read Only | Visibility | Multiplicity | Default Value |
| isOwner | Owner of System | Bool | No | Yes | Private | 1 | <None> |

**Operations:**

|  |  |
| --- | --- |
| Name | EditAccount |
| Description | Administrator can edit the account information of a traveler account |
| Return Type | none |
| Parameters | account  Direction: In  Traveler  Default: None |
| Visibility | Public |
| Scope | Instance |
| Is Query | No |

**Processing Outline:**

Administrator edits a traveler’s username, password, email or phone number

If valid information

Allow the account information to be changed and update the user database

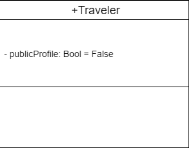
Else

Inform Administrator that account information is invalid

|  |  |
| --- | --- |
| Name | PrintTravelerList |
| Description | Prints a list of the traveler’s that have accounts in the system |
| Return Type | none |
| Parameters | none |
| Visibility | Public |
| Scope | Instance |
| Is Query | No |

**Processing Outline:**

Administrator clicks print traveler list button and receives of a list of traveler’s in the system.



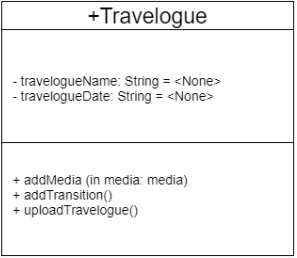
Description: Class for traveler account

Visibility: Public

Is Abstract: No

**Attributes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Data Type | Is Derived | Is Read Only | Visibility | Multiplicity | Default Value |
| publicProfile | Profile is public for anyone to view | Bool | No | Yes | Private | 1 | <False> |



Description: Class for Travelogue to show off travel to friends

Visibility: Public

Is Abstract: No

**Attributes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Data Type | Is Derived | Is Read Only | Visibility | Multiplicity | Default Value |
| travelogueName | Name | String | No | Yes | Private | 1 | <None> |
| travelogueDate | Date Created | String | No | Yes | Private | 1 | <None> |

**Operations:**

|  |  |
| --- | --- |
| Name | addMedia |
| Description | Adds media (picture/video/map/text) to the travelogue |
| Return Type | none |
| Parameters | media  Direction: In  media  Default: None |
| Visibility | Public |
| Scope | Instance |
| Is Query | No |

**Processing Outline:**

Traveler clicks add media and selects a file previously uploaded to the journal

If valid file

Adds media to the travelogue with the date and allows user to name it

Else

Tells user that it is an invalid file (or too large)

|  |  |
| --- | --- |
| Name | addTransition |
| Description | Traveler can add transitions between pieces of media to edit the travelogue into a video |
| Return Type | none |
| Parameters | none |
| Visibility | Public |
| Scope | Instance |
| Is Query | No |

**Processing Outline:**

Traveler selects “add transition” and selects from available transitions to be placed between pieces of media in the travelogue.

|  |  |
| --- | --- |
| Name | uploadTravelogue |
| Description | Traveler can upload completed travelogue to share with family and friends |
| Return Type | none |
| Parameters | none |
| Visibility | Public |
| Scope | Instance |
| Is Query | No |

**Processing Outline:**

Traveler selects upload travelogue and is given options on format of upload

If Traveler selects public

Any viewer without an account will be able to search and view their travelogue

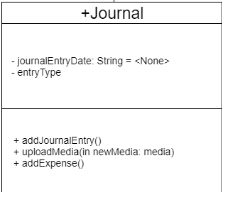
If Traveler selects private

Only the traveler will be able to view the travelogue on their devices, and will not be able to share it with anyone else

If Traveler selects provide link

A link will be provided that can be sent to family and friends to view the travelogue. Only people with the link will be able to view the travelogue and the link can be later changed if it gets out to the public on accident.

Else will automatically be private



Description: Class for Journal to add pieces of media to

Visibility: Public

Is Abstract: No

**Attributes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Data Type | Is Derived | Is Read Only | Visibility | Multiplicity | Default Value |
| journalEntryDate | Date journal is created | String | No | Yes | Private | 1 | <None> |
| entryType | Type of Media | Media | No | Yes | Private | 1 | <None> |

**Operations:**

|  |  |
| --- | --- |
| Name | addJournalEntry |
| Description | Traveler can add a text journal entry to the journal |
| Return Type | none |
| Parameters | none |
| Visibility | Public |
| Scope | Instance |
| Is Query | No |

**Processing Outline:**

Traveler selects “Add Journal Entry”

If Valid journal entry

Saves and returns to journal

Else

Reports unsuccessful save of journal entry

|  |  |
| --- | --- |
| Name | uploadMedia |
| Description | Traveler can add a media file (picture/map/video) to the journal |
| Return Type | none |
| Parameters | media  Direction: In  media  Default: None |
| Visibility | Public |
| Scope | Instance |
| Is Query | No |

**Processing Outline:**

Traveler selects “Upload Media Entry”

If valid media file

Saves and returns to journal

Else

Reports unsuccessful save of media

|  |  |
| --- | --- |
| Name | addExpense |
| Description | Traveler can add a transaction to the Expenses page |
| Return Type | none |
| Parameters | none |
| Visibility | Public |
| Scope | Instance |
| Is Query | No |

**Processing Outline:**

Traveler selects “Add Expense”

If valid expense

Saves and returns to journal

Else

Reports unsuccessful save of transaction/expense

1. **ARCHITECTURE DESIGN**

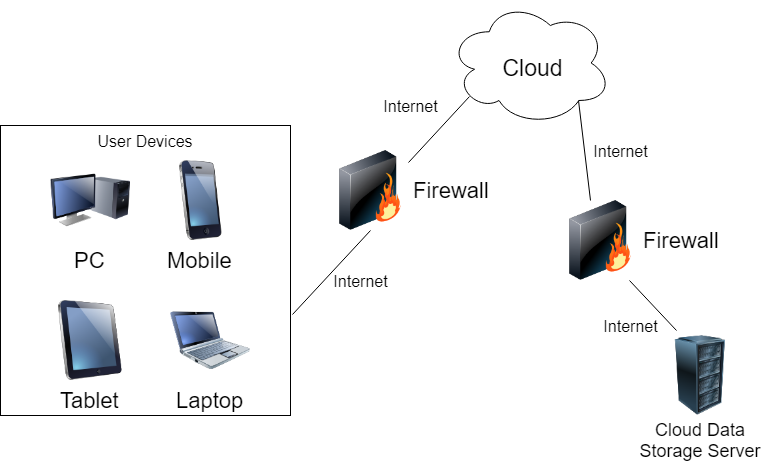
**3.1 Introduction**

This section contains two infrastructure models, one of the architecture overview and, and the other of the nodes of artifacts for the Travel Assist System (TAS). It also contains the hardware and software needed to complete the system, and a brief paragraph about security.

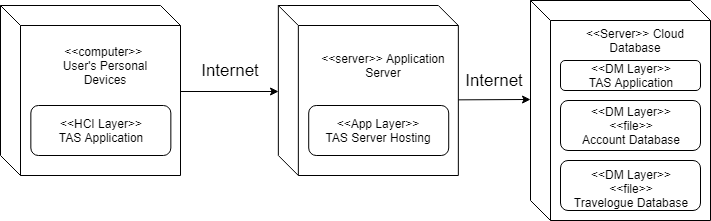
The cloud will be used to store data from user accounts, so uploads can be accessed on any device the user logs into. This adds reliability in what can be a hectic organizational time for users while traveling. The cloud-server will also handle purchases of account subscriptions so users will not have to store large amounts of data on their systems using the application.

**3.2 Infrastructure Model**

**Deployment Diagram 1: Architecture Overview**



**Deployment Diagram 2: Nodes and Artifacts**



**3.3 Hardware and Software Requirements**

**Required Hardware Components:**

* User needs a device that can access the internet and is supported by TAS
* Computer for Administrator
* Cloud Server Subscription

**Required Software Components:**

* TAS must support the most common web browsers, Chrome, Edge, Firefox and Safari
* TAS must support both current and some previous operating systems for laptops, computers, mobile devices and tablets.
* Cloud service must include user account and travelogue database management capabilities

**3.4 Security Plan**

The most essential threats is the privacy and safety of traveler’s. TAS will hire a reputable payment handler to implement the subscription based purchasing system. It is crucial that payment information remain confidential. Additionally, a secure cloud storage provider will be used to keep travel information private. There will be no risk of travelers being tracked in anyway while on their travels with this system.

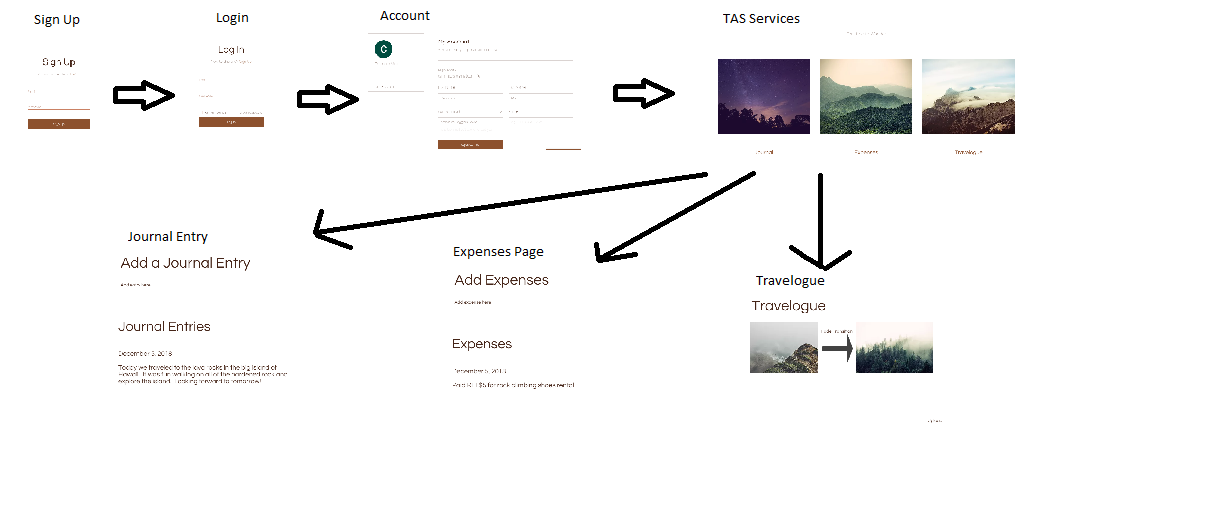
1. **USER-INTERFACE**

**4.1 User-Interface Requirements and Constraints**

In the following sections are GUI models for the UI that will be created for the website. There is a window navigation diagram that shows step by step how the users will interact with the system. The images shown in the windows navigation diagram are shown later in the screen/UI section

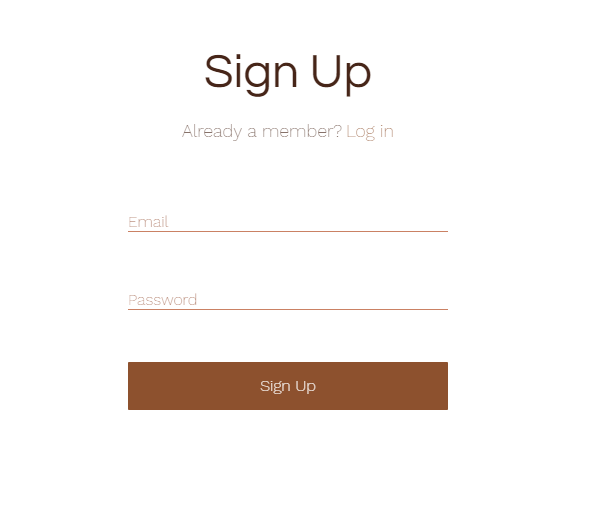
It is important that the UI looks neat and professional. The system will be used by all ages and experiences of people, so all users should be able to use the system effectively regardless of their technology skill level. Because of this, a simple, modern UI was implemented that makes subsequent steps clear.

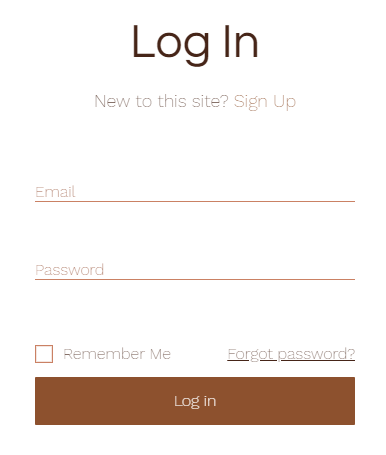
**4.2 Window Navigation Diagram**



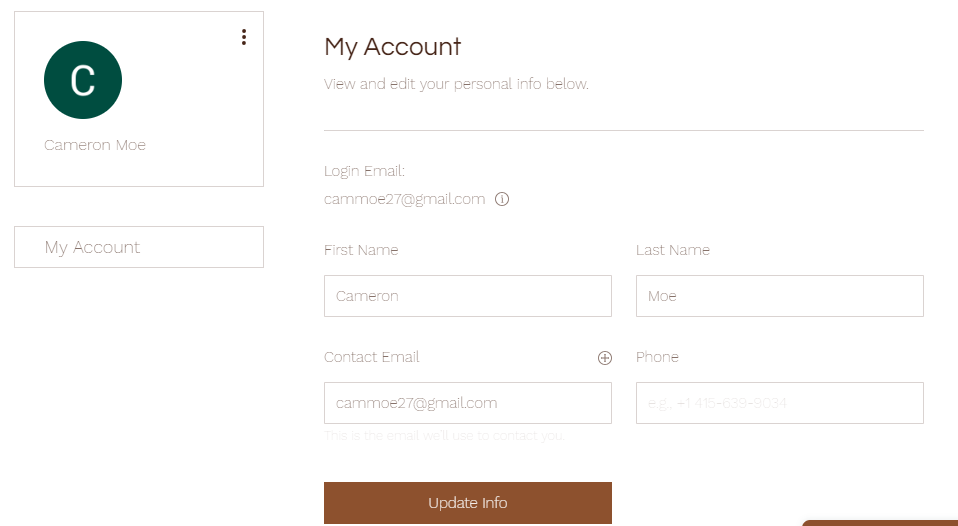
**4.3 Forms: Screen / User-Interaction Design**

Sign up and Log in Pages

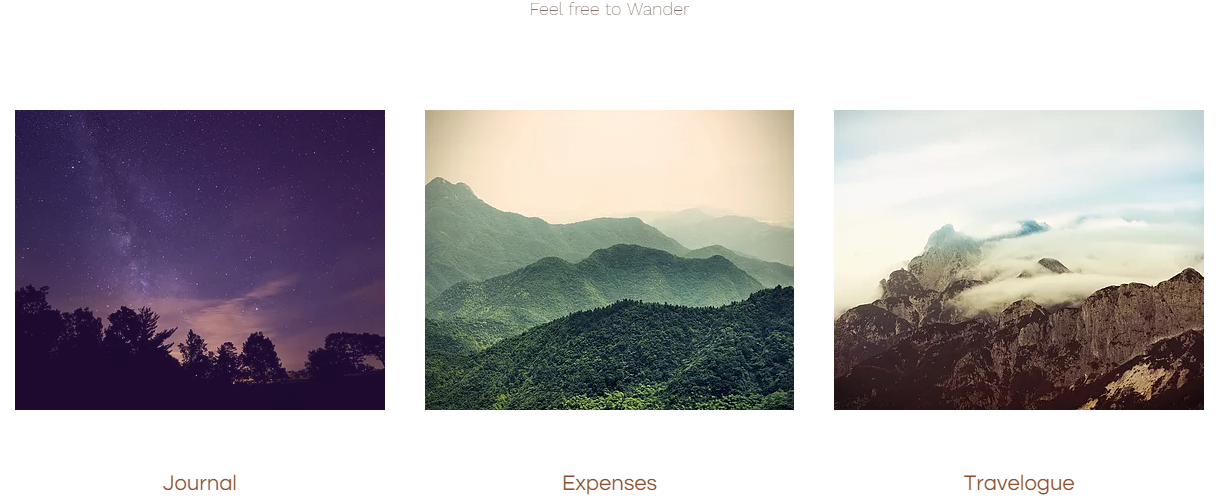




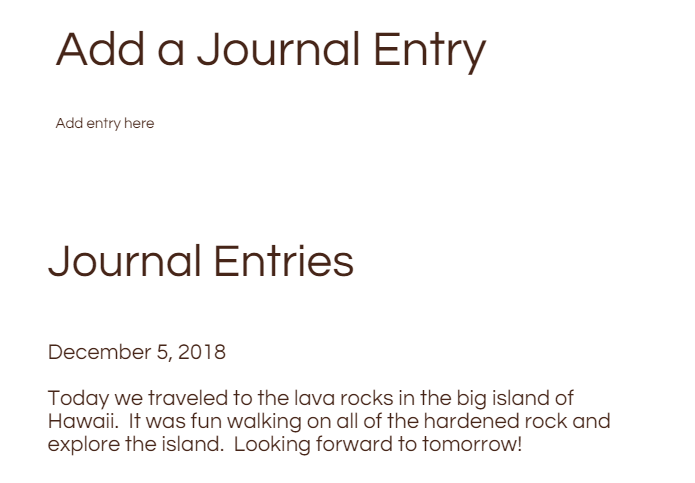
User Account Page



TAS Service Options



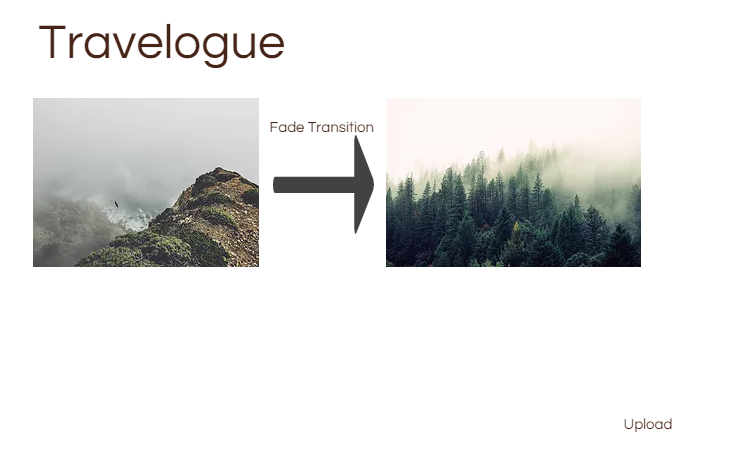
Journal Page



Expenses Page



Travelogue Page



1. **APPENDICIES**

**5.1 Bibliography**

Dennis, Alan, Barbara Haley Wixom, David Paul Tegarden, and Elaine Seeman. System Analysis and Design: An Object-Oriented Approach with UML. Hoboken, NJ: Wiley, 2015. Print.

Pfeiffer, William S. Pocket Guide to Technical Communication. Upper Saddle River, NJ: Prentice Hall, 2011. Print.

Weltz, Elaine. "Requirements - What We're Seeking”, “Project Initiation I”, and “Project Initiation II: Feasibility and Risk." CSC 3150. Seattle Pacific University, Seattle. Oct. 2018. Lecture.

Weltz, Elaine. “3150 Topics and Units”, Units 1-5. Seattle Pacific University. 2018.

All Diagrams made using draw.io

UI Website Design made using Wix