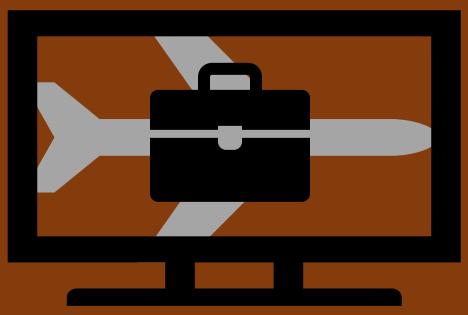


Travollio (Travel Journal) System Specification

Hyperion-Gold Ltd.

December 5, 2018

Table of Contents

Executive Summary

This document lays out the design of the Travollio System, including class diagrams, system architecture, and UI mock-ups. Travollio is, on its base level, an application that will allow users to put a travel journal together on the fly. Travollio supports many data types for the Journal so the user can add whatever trip artifacts they deem necessary.

Travollio was conceive and its design initiated on October 16th when a system proposal document was created in response to a system request Hyperion-Gold received from Wanderer's Tools. Since then the system proposal created the general criteria for the system as well as laying out the basic requirements and design of the system. Through this document the concepts of this system have been more fully fledged out and the system is now ready to be put into a production.

1.0 Introduction

1.1 Problem Statement and Project Vision

Hyperion Gold has agreed to help Wanderer's Tools with a new system intended to make the creation of Travel Journals easy and stress-free. People always put together scrap books, or photo albums to commemorate travels and trips they have taken throughout their life-time, Travollio (our travel journal system) aims to make that easier to do. Travollio is intended to be used throughout the time the Traveler is journeying so that they don't have to set day(s) aside to try and pull all their trip artifacts together after the fact. People who would benefit from this system or our stakeholders are Travelers, Travel Companies, friends and family of the Travelers, Wanderer's Tools, and Hyperion-Gold. Travollio will help Travelers pull together compendiums of their travels, their friends will be able to enjoy them, and Travel Companies will be able to learn what destinations and activities Travelers are enjoying now, so they can cater experiences more Travelers can enjoy.

1.2 System Services

Here a few brief descriptions of the basic functional requirements for Travollio will be listed. Note that these requirements are discussed in more depth in the System Proposal for Travollio, in sections 4, and 5.

- **Travelers**
  - **Account:** Facilitates the creation, removal, and edit of a user account
  - **Create Travollio:** Allows for the initialization of a Travollio project
  - **New Entry:** Allows for the initialization of a new Journal entry in an existing project
  - **Edit Journal:** Facilitates the editing of Journal aspects
  - **Add Entry Data:** Allows for the addition of new data objects onto an entry page
  - **Compile Journal:** Pulls together all a Journals information and packages it for easier sharing
  - **Share Journal:** Allows for the sending of a compiled, or an editable Journal to another user
  - **View Journal:** Enables a user to view a compiled Journal
  - **Access Generalized User Information:** Enables an administrator or other interested party to access desired information about users of Travollio

1.3 Nonfunctional Requirements & Constraints

These requirements and constraints for Travollio are based off the project information we received from Wanderer's Tools for this project and are recommended to keep this system inline with its initial vision. More in depth information about these requirements/constraints as well as additional requirements can be found in the Travollio System Proposal document.

- Travollio must have versions for all major platforms including Windows, Android, IOS, and Mac
- Interface must be intuitive and easy to understand/use
- System Operation must not be reliant on a stable internet connection
- Information shared with outside businesses must not open users up to travel spam
- Travollio must automatically save user work to prevent lost work from sudden or accidental shutdown of the application
- User information must be stored in a secure location and be encrypted to prevent unauthorized access

1.4 System Evolution

Permitting there is additional time/budget we have prepared a few additional features for Travollio to be implemented after the release/completion of version 1.0. These features can be added post-production provided there is sufficient interest in Travollio.

- **Social Media Publishing:** Compiling a finished journal to share is already a planned feature, this however would allow users to push their journals onto social media platforms in a proper format. It would however require users to be finished with their journal.

- **Multiple Traveler Journals:** Sharing raw journal files so that other travelers can edit the same journal will already be supported, however this feature would allow users to host their projects on a cloud server so that multiple travelers can work on a project simultaneously.

Additional information regarding these planned features can be found in the Travollio System Proposal document under the system evolution entry.

1.5 Document Outline

The rest of this document will contain various models to convey the structure of Travollio and how we currently intend for it to appear in its final form. Each section will be geared toward a specific aspect of the system. These sections are as follows.

1. **Structural Model**: This section will contain an outline of the systems current class concept and discuss how each component will interact with various elements and actors within the system.

2. **Architecture Design:** This section will contain a model of the planned infrastructure for the system as well as plans for protecting that infrastructure from various threats.

3. **User Interface:** This section will contain examples of how the interface should look to users and will discuss the requirement for the systems UI.

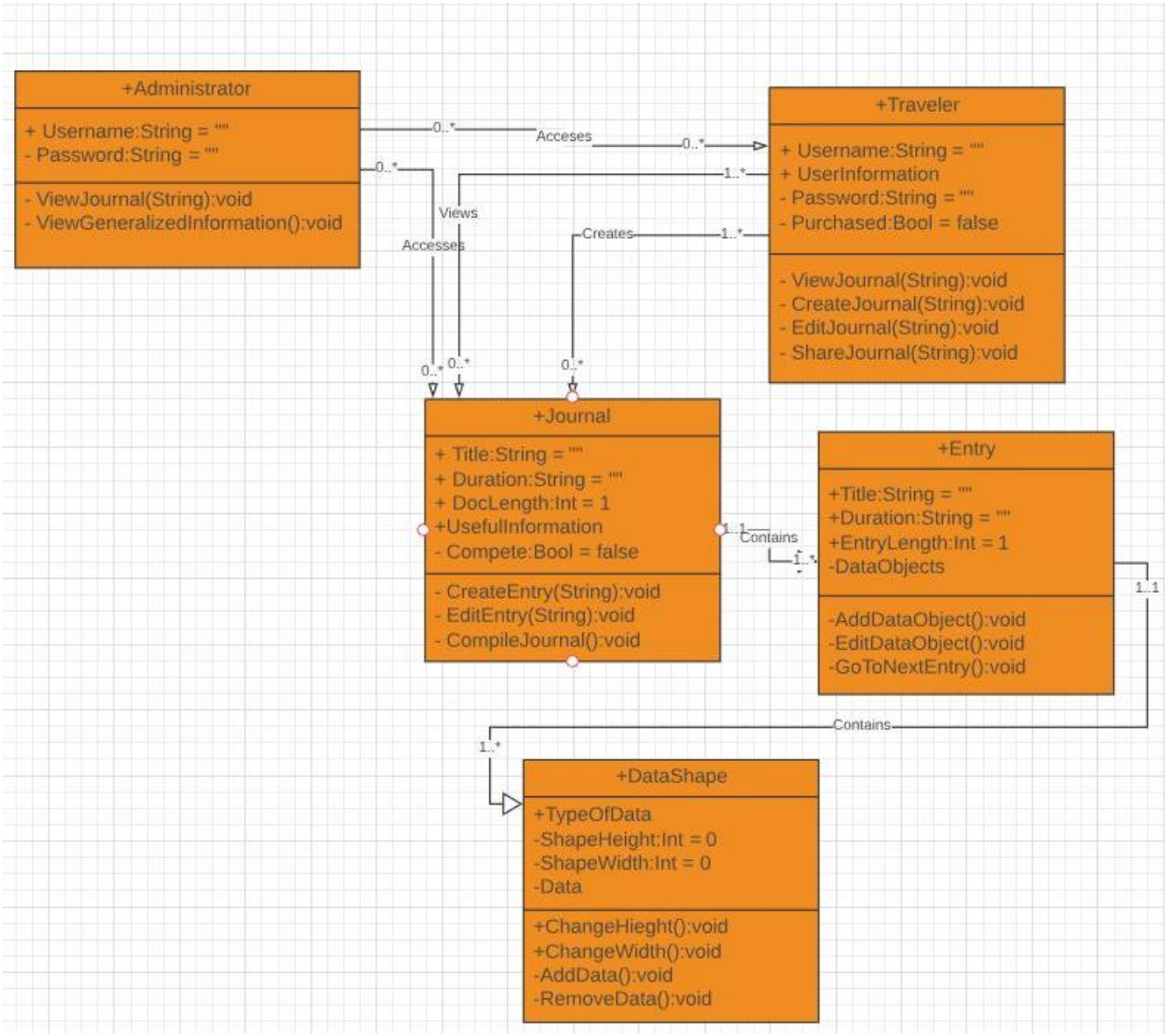
4. **Appendices:** This section will contain a bibliography that will reference each outside document that inspired and was utilized in the creation of this document.

2.0 Structural Model

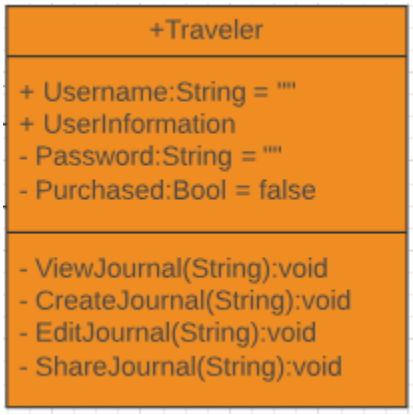
2.1 Introduction

This section contains a diagram of all the main classes and components of Travollio and tables of all their attributes and operations. Beyond the diagram is a more detailed description of the attributes and operations of each class/component that will outline how they operate and interact with other elements of the system. Within these descriptions are also instructions on how each operation is performed and the general affect of the operation on the other parts of the system. This will hopefully convey the main principles and design plan of the Travollio system to the people working on it.

2.2 Class Diagram



2.3 Metadata



Description: Represents a user account for someone who uses the Travollio Application
Visibility: Public
IsAbstract: No

Attributes

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
User Name	Username for the account	String	No	No	Public	1	Empty String
User Info	General demographic data	Varied	No	No	Public	1	NULL
Password	Password tied to the account	String	No	No	Private	1	Empty String
Purchased	Marker for paying user	Bool	No	No	Private	1	False

Operations

Name	Description	Return Type	Parameters	Visibility	Scope	IsQuery	IsPolymorphic
ViewJournal	Opens a Journal in a read format	Void	Direction: In Journal Name	Private	Instance	Yes	No
CreateJournal	Creates a journal with name	Void	Direction: In Journal Name	Private	Instance	Yes	No
EditJournal	Edits Journal of name	Void	Direction: In Journal Name	Private	Instance	No	No
ShareJournal	Shares Journal of name	Void	Direction: In Journal Name	Private	Instance	Yes	No

Processing Outlines

ViewJournal: User will select a Journal to view

System will initialize the read mode

System will open the selected Journal by its name to the read mode

CreateJournal: User will select Create Journal and enter a name

System will initialize a Journal project under the specified name

System will open the edit mode on the new empty project

EditJournal: User will select Open Journal and select the Journal to open

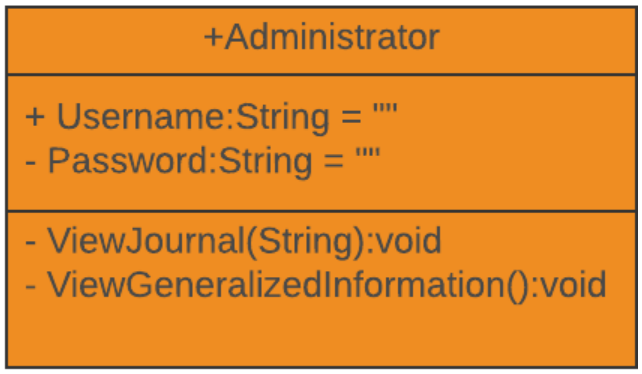
System will initialize the edit mode

System will load the Journal under the name selected into the edit mode

ShareJournal: User will select Share Journal and select the Journal to share, then specify the other User to send it to

System will determine if the Journal is a completed project file

System will either send a raw project to the specified User if the selected project was not completed, or it will send a completed project file to the specified user



Description: Represents an Admin account for someone who has access to the user and Journal database
Visibility: Public
IsAbstract: No

Attributes

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
User Name	Username for the account	String	No	No	Public	1	Empty String
Password	Password for the account	String	No	No	Private	1	Empty String

Operations

Name	Description	Return Type	Parameters	Visibility	Scope	IsQuery	IsPolymorphic
ViewJournal	Opens a Journal in a read format	Void	Direction: In Journal Name	private	Instance	Yes	No
ViewGeneralizedInfo	Opens collected user info in a read format	Void	NULL	private	Instance	No	No

Processing Outlines

ViewJournal: Administrator will select a Journal to view from list of Journals in the Database

System will initialize the read mode

System will open the selected Journal by its name to the read mode

ViewGeneralizedInfo: Administrator will select the number of Users they want to see the info from

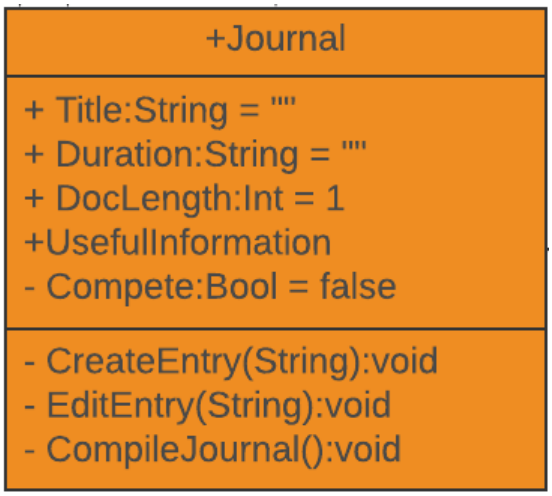
System will pull data from Users stored in the database for the amount specified by the administrator

System will sort data into types of data that might be useful to the Administrator including, destinations, time spent travelling, activities, and age

System will present types of data to be looked at to the Administrator

Administrator will select what types of data they wish to view

System will open the specified data types in a view format for the Administrator



Description: Represents a Journal project, either completed or in progress
Visibility: Public
IsAbstract: No

Attributes

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
Title	Name of the Journal	String	No	No	Public	1	Empty String
Duration	Time covered by the Journal	String	No	No	Public	1	Empty String
DocLength	Number of Journal Entries	Integer	No	No	Public	1	1
UsefulInfo	Information that could be useful to admins	Varied	No	No	Public	1	NULL
Complete	Marker to tell if project is finished	Bool	No	No	Private	1	False

Operations

Name	Description	Return Type	Parameters	Visibility	Scope	IsQuery	IsPolymorphic
CreateEntry	Creates a new Entry for the Journal	Void	Direction: In Entry Name	Private	Instance	Yes	No
EditEntry	Opens the Edit view for an existing Entry	Void	Direction: In Entry Name	Private	Instance	No	No
CompileJournal	Compiles the Journal into a finished project	Void	NULL	Private	Instance	No	No

Processing Outlines

CreateEntry: User will select Create New Entry from the Journal

System will initialize a New Entry into the Journal and increase DocLength by one

System will present the New Entry in the edit mode

EditEntry: User will select Edit Entry from the Journal and specify what entry they wish to edit

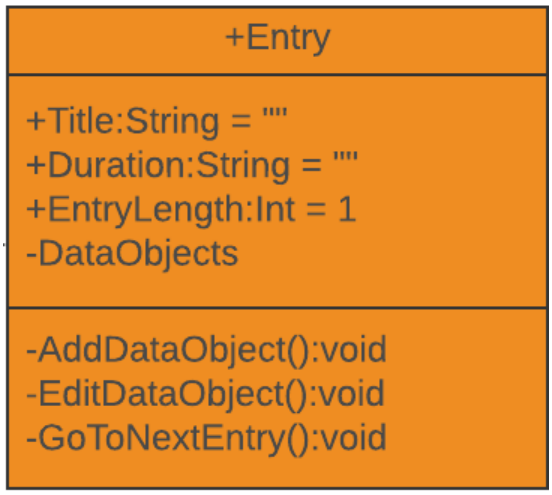
System will initialize the edit mode

System will present the specified entry in the edit mode

CompileJournal: User will select Compile Journal and specify their format

System will compress data and format all the Journal entries into the specified format

System will present the user with a finished and compressed project packaged in a Journal file type .tjrnl



Description: Represents an Entry within a Journal project
Visibility: Public
IsAbstract: No

Attributes

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
Title	Name of the Entry	String	No	No	Public	1	Empty String
Duration	Time covered by the Entry	String	No	No	Public	1	Empty String
EntryLength	Number of pages in the Entry	Integer	No	No	Public	1	1
DataObjects	A list of the Data Objects contained in this Entry	Varied	No	No	Private	1	NULL

Operations

Name	Description	Return Type	Parameters	Visibility	Scope	IsQuery	IsPolymorphic
AddDataObject	Adds a new Data Object into the entry	Void	NULL	Private	Instance	No	No
EditDataObject	Edits an existing Data Object	Void	NULL	Private	Instance	No	No
GoToNextEntry	If another Entry exists will travel to the next Entry	Void	NULL	Private	Instance	Yes	No

Processing Outlines

AddDataObject: User will select add Data Object

System will prompt them to select the shape of the Object

User will select a shape

User will draw the shape onto the Entry Page

System will turn the shape into a Data Object

User will add Data to the shape or finish creating the shape

EditDataObject: User will select edit Data Object

System will prompt them to change the Data in the shape

User will change or keep the Data

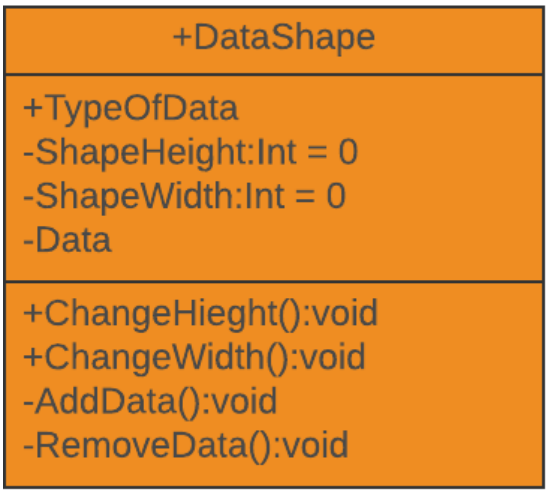
System will prompt the user to modify the shape or size of the shape

User will change or keep the shape

GoToNextEntry: User will select to move to the next Entry

System will check if there is another existing Entry

If another Entry exists System will open this Entry on the edit mode, otherwise User will be prompted that there is not another entry and will be given the option of creating a new Entry



Description: A Represents a Data Object within an Entry
Visibility: Public
IsAbstract: No

Attributes

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
TypeOfData	Identifies the type of data stored in this shape	Varied	No	No	Public	1	NULL
ShapeHeight	Tracks the height of the shape	Integer	No	No	Private	1	0
ShapeWidth	Tracks the width of the shape	Integer	No	No	Private	1	0
Data	Holds the data for the shape	Varied	No	No	Private	1	NULL

Operations

Name	Description	Return Type	Parameters	Visibility	Scope	IsQuery	IsPolymorphic
ChangeHeight	Changes the height of the shape	Void	NULL	Public	Instance	No	No
ChangeWidth	Changes the width of the shape	Void	NULL	Public	Instance	No	No
AddData	Adds Data to the shape	Void	NULL	Private	Instance	No	No
RemoveData	Removes Data on the shape	Void	NULL	Private	Instance	No	No

Processing Outlines

ChangeHeight: User selects to change the Data Shape height

System changes the height as the User desires

ChangeWidth: User selects to change the Data Shape width

System changes the width as the User desires

AddData: User selects to add Data to the Data Shape

System puts the requested Data onto the Shape overwriting existing Data on the Shape

RemoveData: User selects to remove data from the Data Shape

System removes the data stored on the Data Shape

3.0 Architecture Design

3.1 Introduction

This section will discuss everything related to the hardware design of the Travollio system. A few diagrams will visualize the envisioned hardware infrastructure we have in plan for this system. After the diagrams we will discuss what hardware and software will be required for the successful deployment of our hardware design for Travollio. This will hopefully create a full understanding of how we intend the infrastructure for this system to be laid out and how it should ideally operate.

3.2 Infrastructure Model

Diagram 1: Architecture Overview

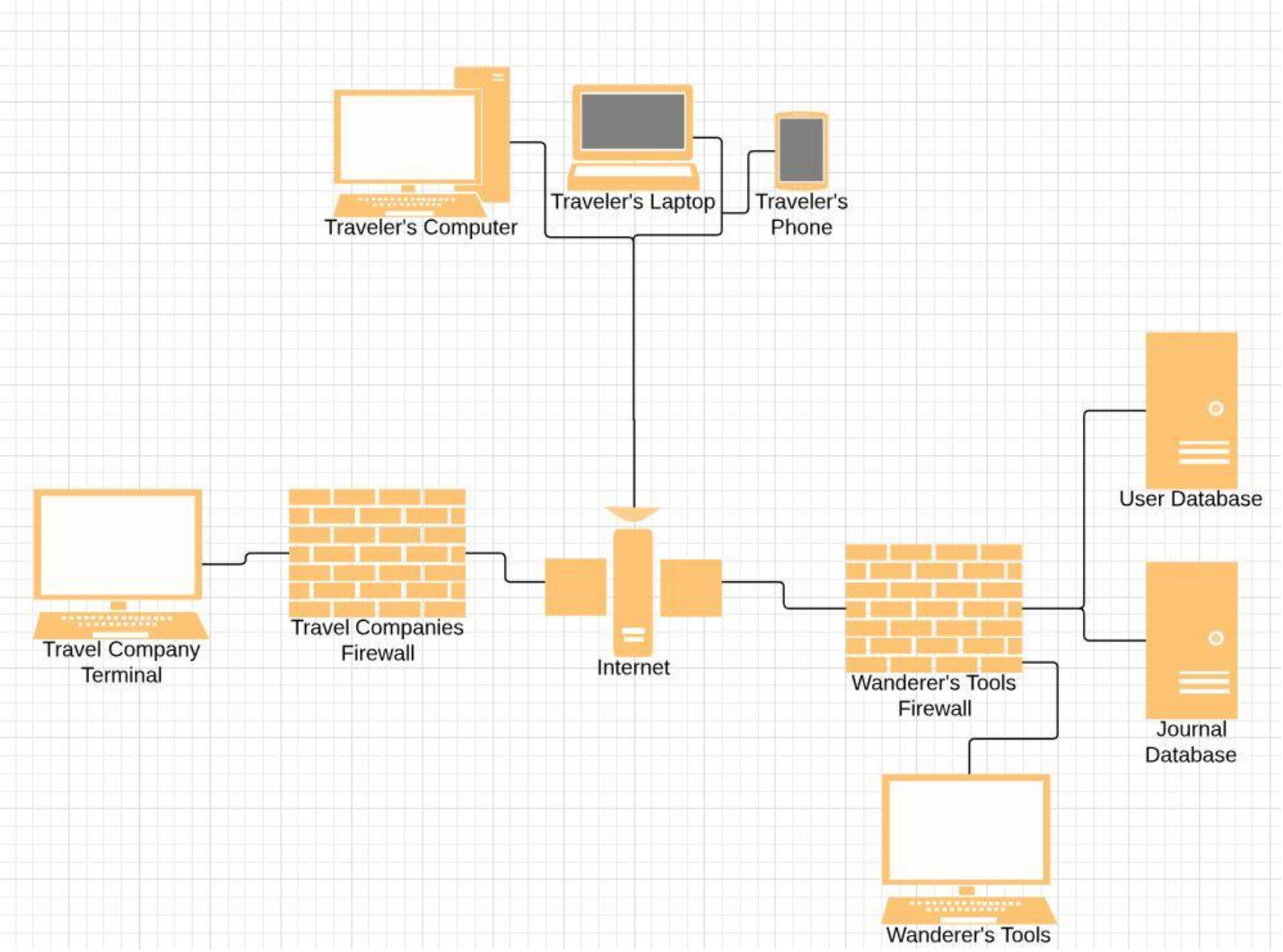
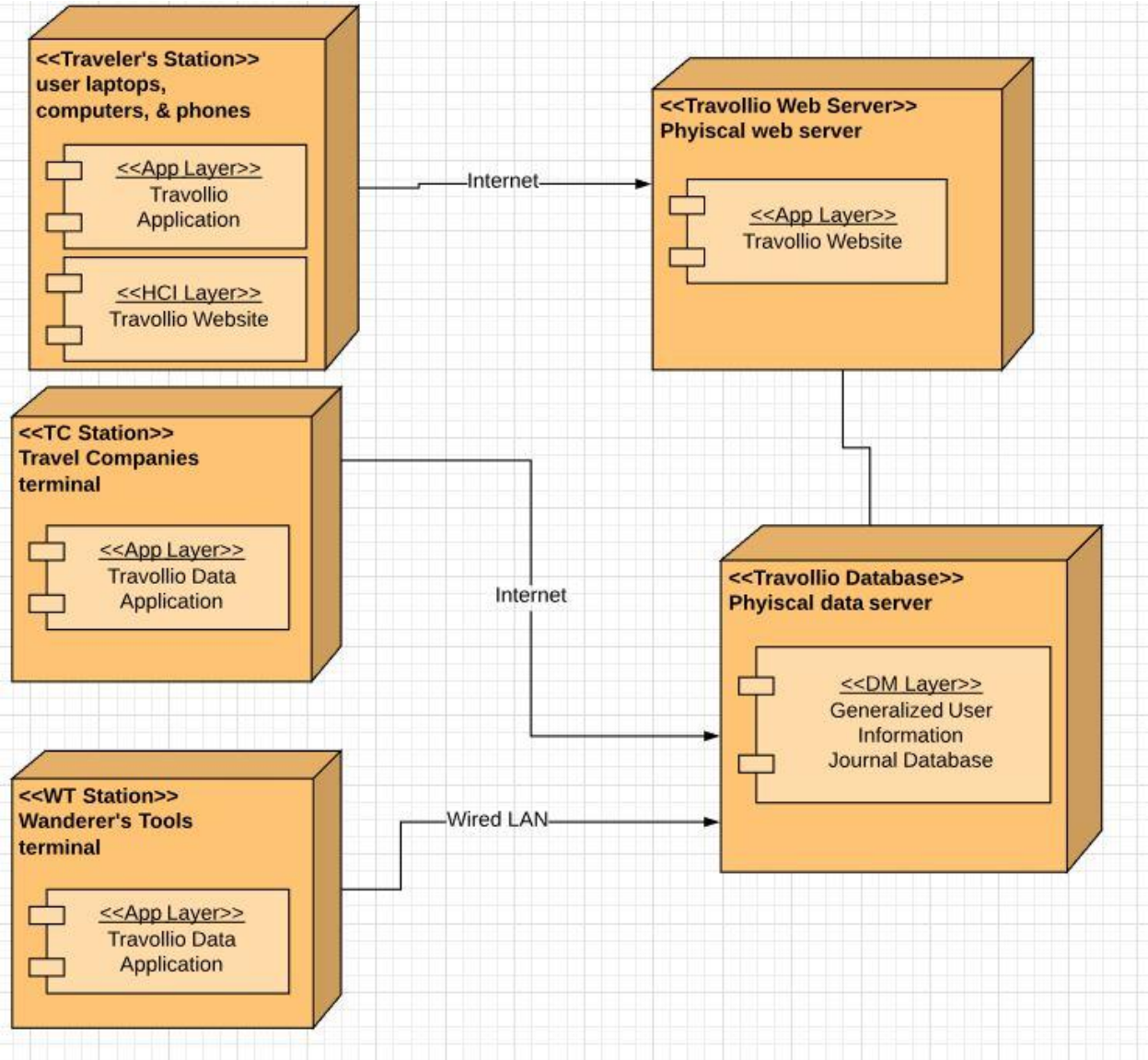


Diagram 2: Nodes & Artifacts



3.3 Hardware & Software Requirements

Required Hardware:

- **Web Server:** A capable web server will be required to host the Travollio website which will host account creation, purchase product, and download pages.

- **Database Server:** A capable database server will be required to contain and protect all the user account information and stored journal projects for the Travollio system.

- **Local Terminal(s):** Local terminals capable of accessing and modifying the database and web server are required to maintain each server and to pull information from the database to be used by Wanderer's Tools at their discretion.

Required Software:

- **Firewall:** A proper and secure firewall software needs to be obtained and deployed on both servers to ensure the protection of Wanderer's Tools, and Travollio Users data.

- **Payment Management:** Payment management software needs to be obtained to streamline and properly secure customer transactions on the Travollio website.

- **Encryption:** Reputable encryption software also needs to be obtained to further ensure user data stored on our servers remains safe in the event of unauthorized access to the servers.

- **Web Hosting:** Basic web hosting software is required to host and maintain the Travollio website that users will be using to purchase and download the Travollio software.

3.4 Security Plan

Physical Security:

There are many physical security concerns for Travollio each will be listed here as well as our plans to minimize risk for each concern. Fire, Flooding, or Earthquake damage to our hardware infrastructure is a real risk. In order to address this, we suggest adding sprinklers to the server room to quell fires and hosting our servers on the second floor of a stabilized three-story building in order to avoid flooding and earthquake damage. Unauthorized access to our servers via tailgating, or forced entry is also a real concern. To address this, we suggest safety awareness training for our employees to address tailgating, and the hiring of security guards for the building to address most cases of forced entry. In addition to this we can add additional firewall security between our terminals and our server, so it would be more difficult for an unauthorized user to access stored data from inside of our facility.

Digital Security:

There are also many digital security concerns for Travollio, these will be listed here along with our plans to minimize risk for each concern. Viruses and hacking attempts are a risk to the security and stability of our website and database. Installing competent anti-virus software and employing high-security firewalls will minimize the threat of a hack or virus crippling our servers. Users will be concerned about the security of data that we've collected from them landing in the wrong hands. To address this, we suggest releasing a list of data partners somewhere on the Travollio website, so users know who we are sharing their data with. In addition to this high-security encryption should be conducted on user data to make any unauthorized access less likely to result in the unwanted use of our collected data.

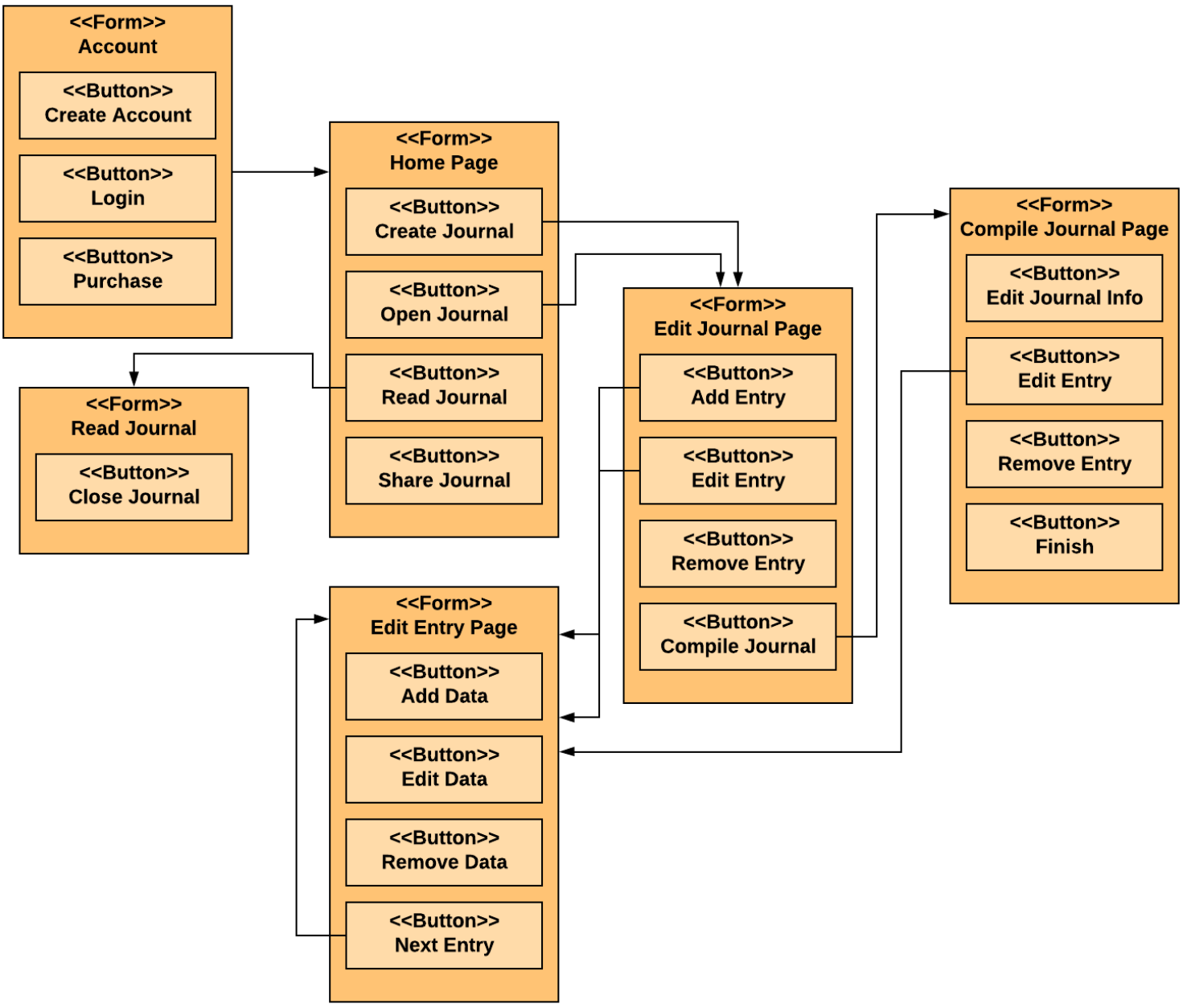
4.0 User Interface

4.1 UI Requirements & Constraints

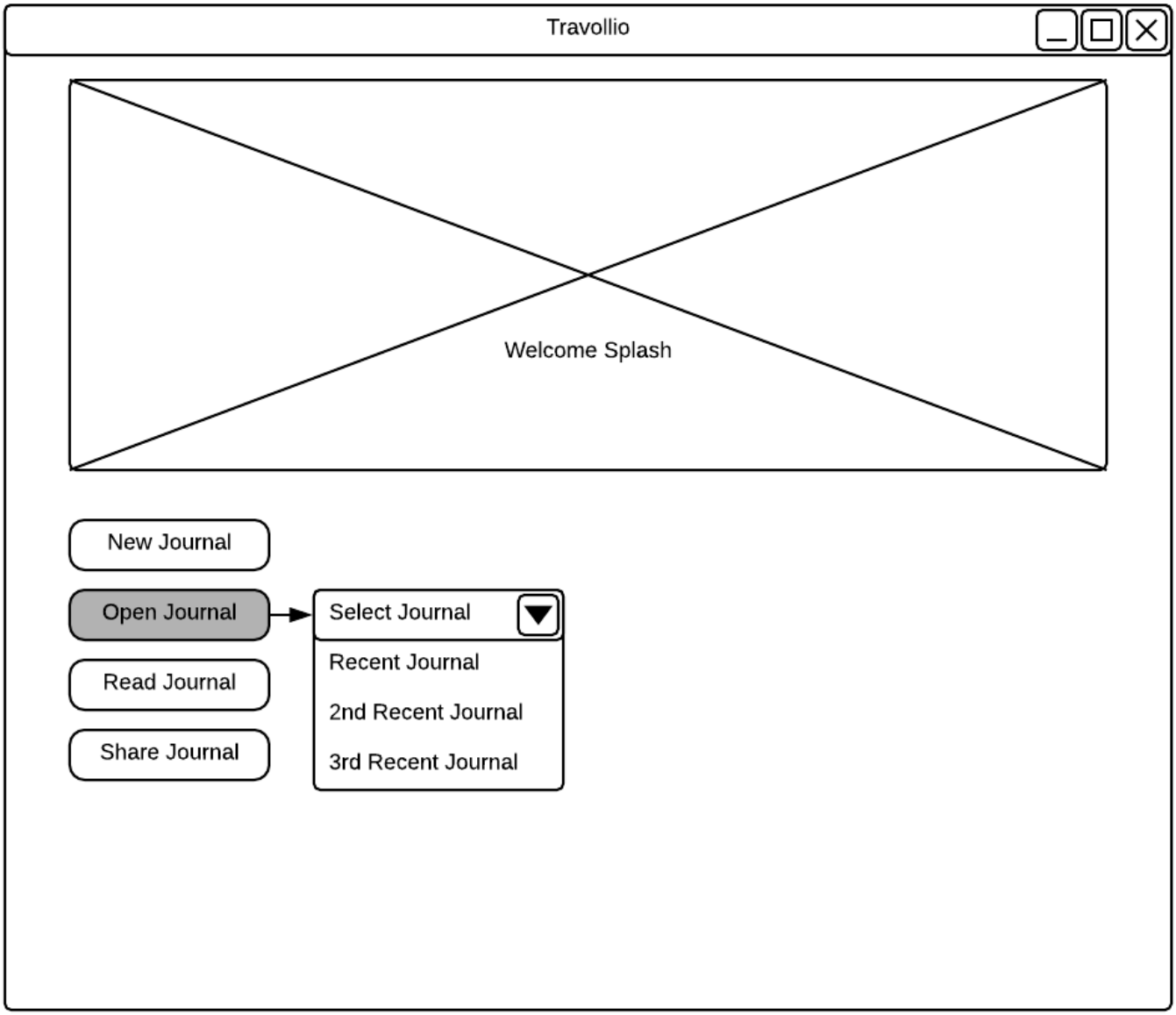
This section will layout the general concepts of the user interface for the various aspects of Travollio. Layouts included are the layout of the main application, the website, and the terminal interfaces for both outside companies and Wanderer's Tools for accessing information on the database. Travollio's main application layout will include the home page, the read Journal page, the new Journal page, and the entry edit page. Travollio's website layout will include the create account page. Travollio's terminal layout will include the Wanderer's Tools terminal request screen, and the outside company terminal request screen.

Travollio's UI is based on the principles of three clicks to desired outcome, and familiar symbols. We want users to not become frustrated with menu navigation, and having each action be at most three clicks away can help cut back on this frustration. Familiar symbols will also cut back on user frustration with trying to understand exactly what the tool they are selecting will accomplish.

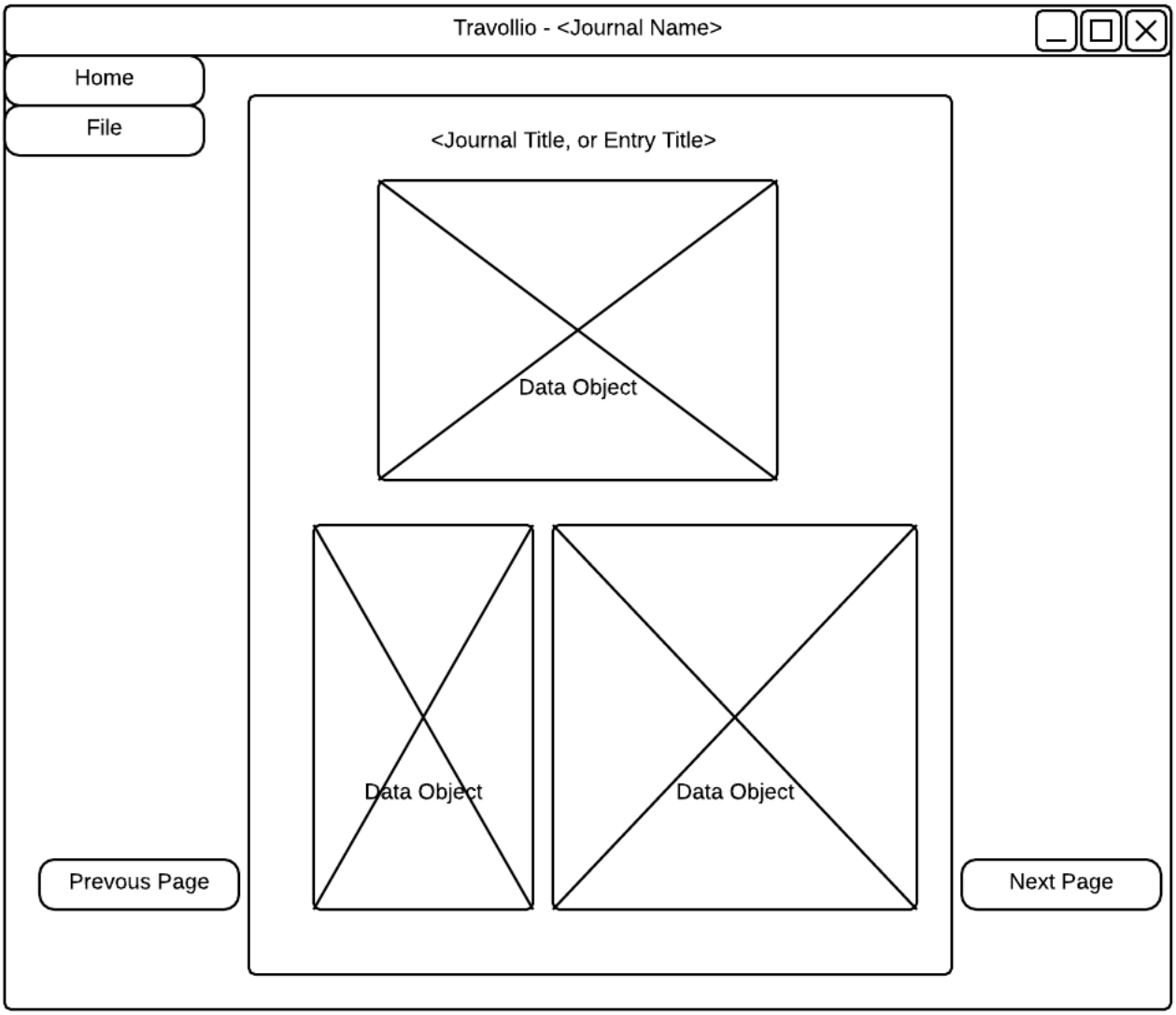
4.2 Window Navigation Diagram



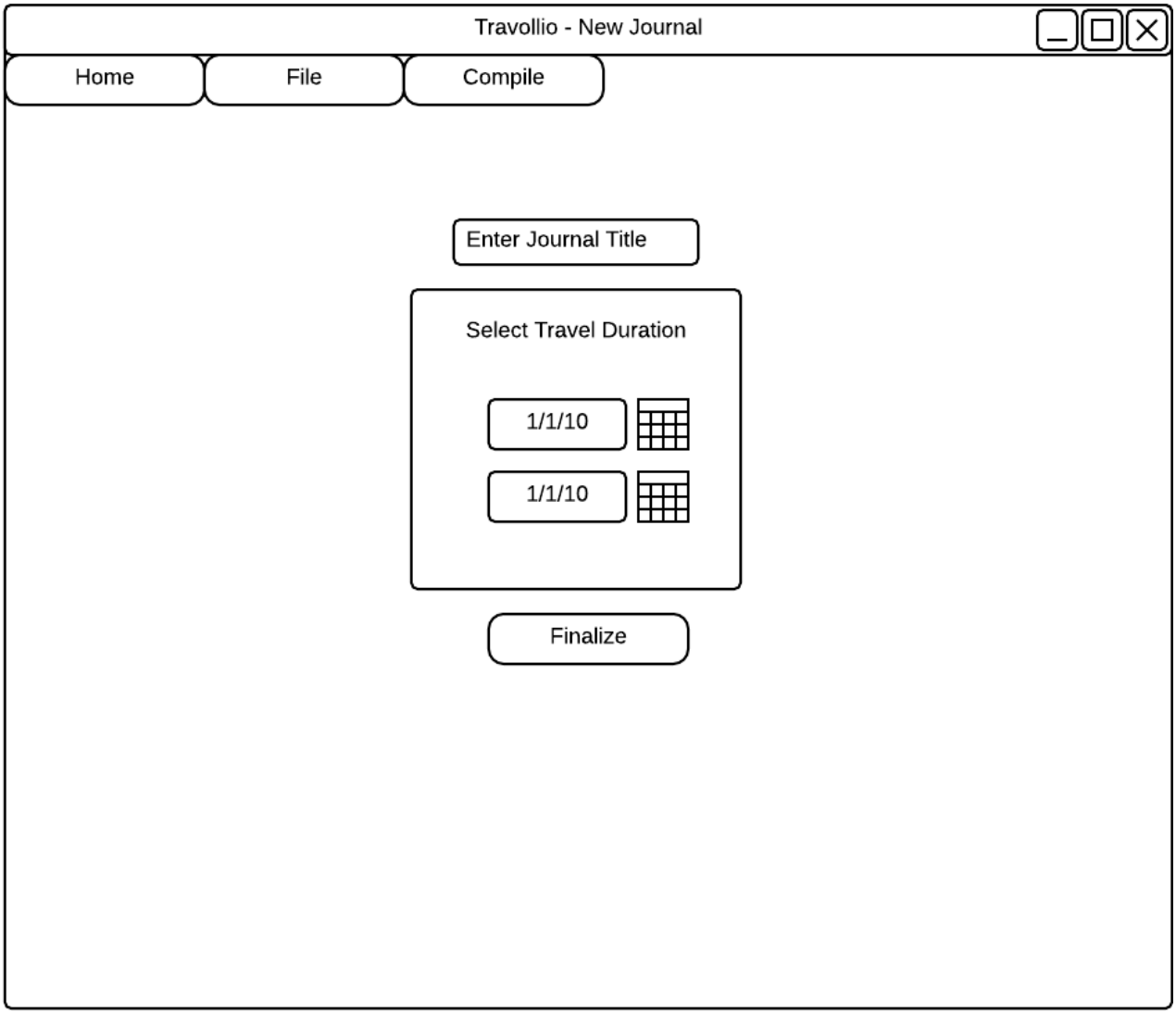
4.3 Forms: Screen/UI Design



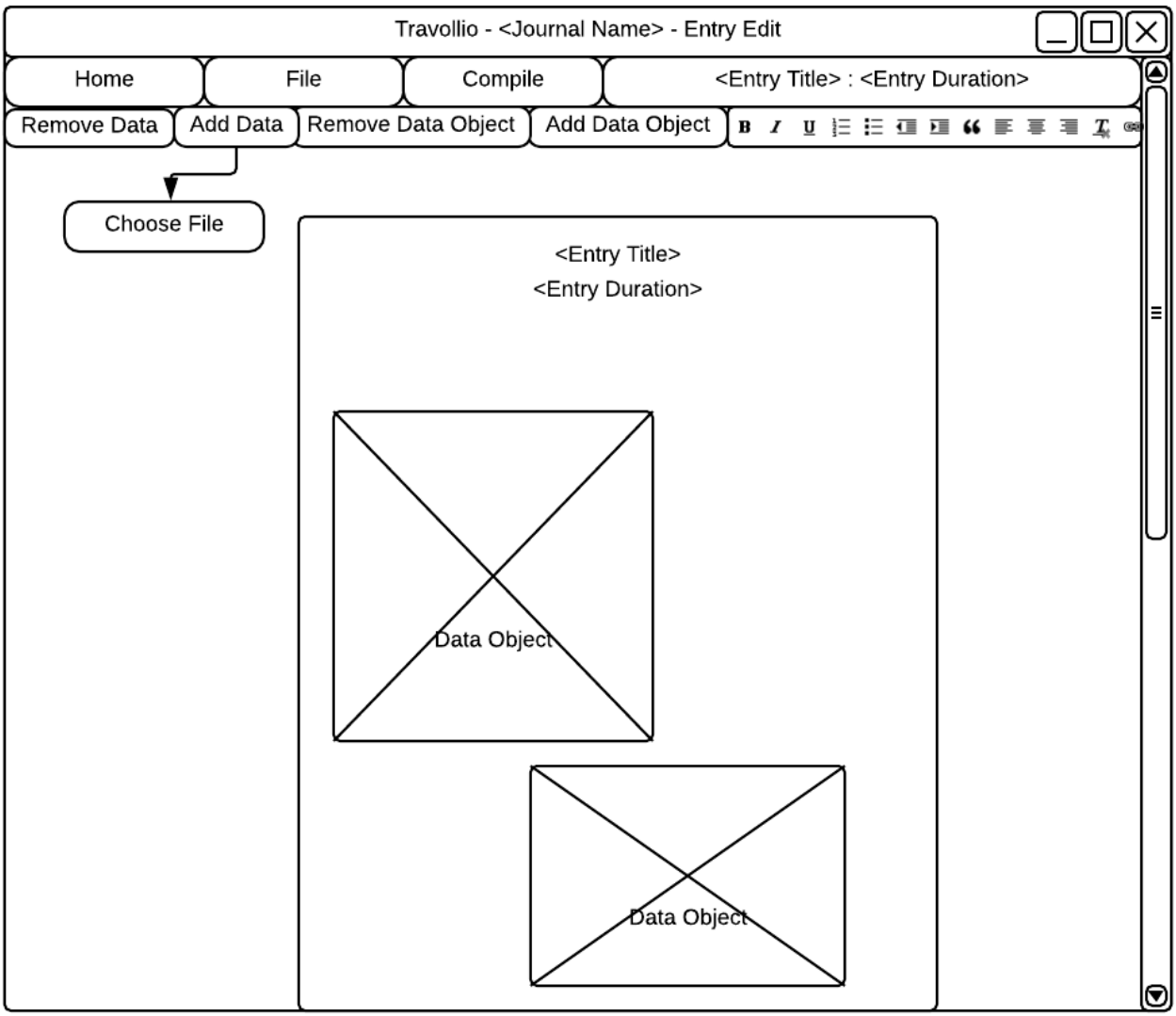
Travollio Application Home Page



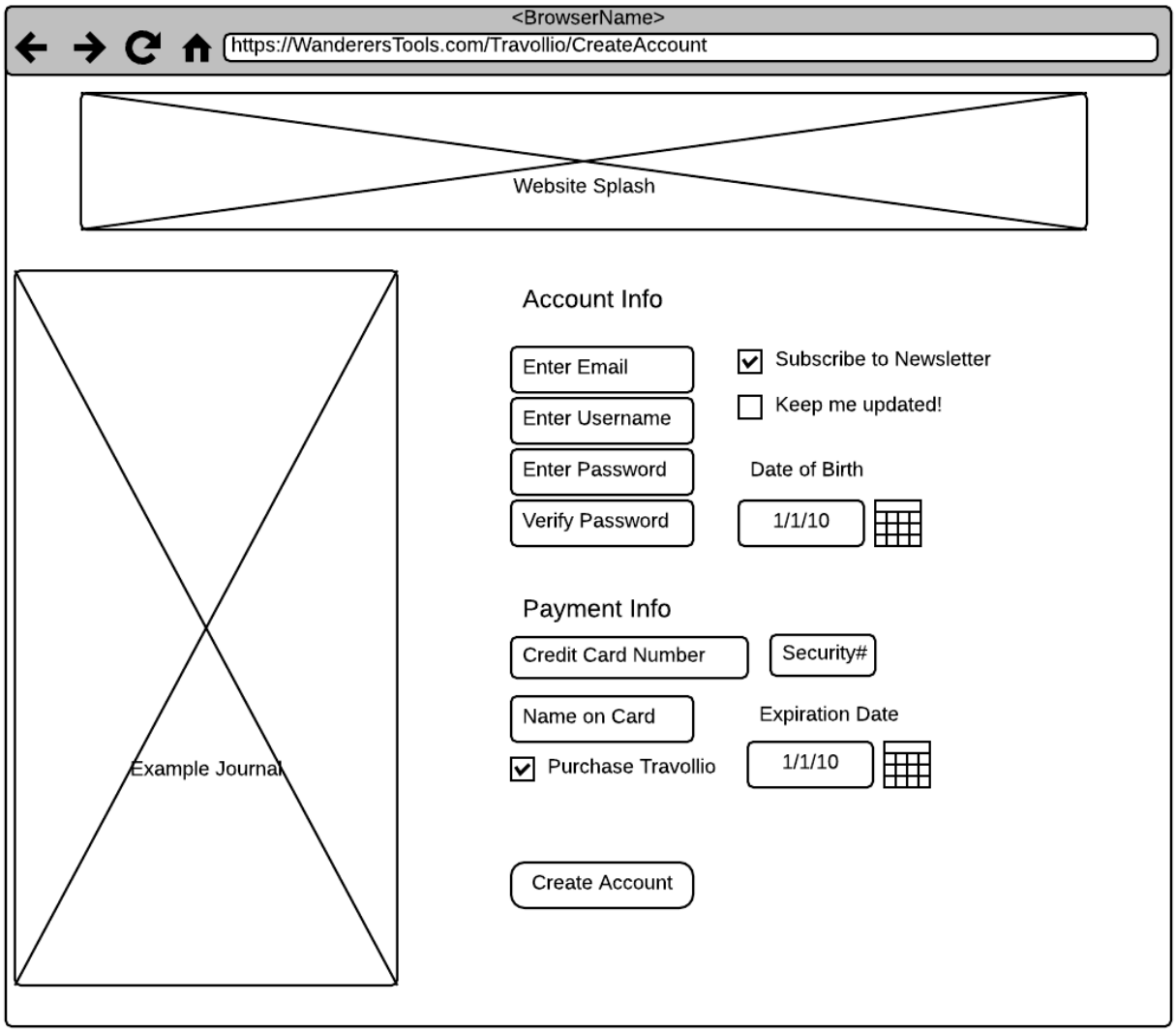
Travollio Application Read Journal Page



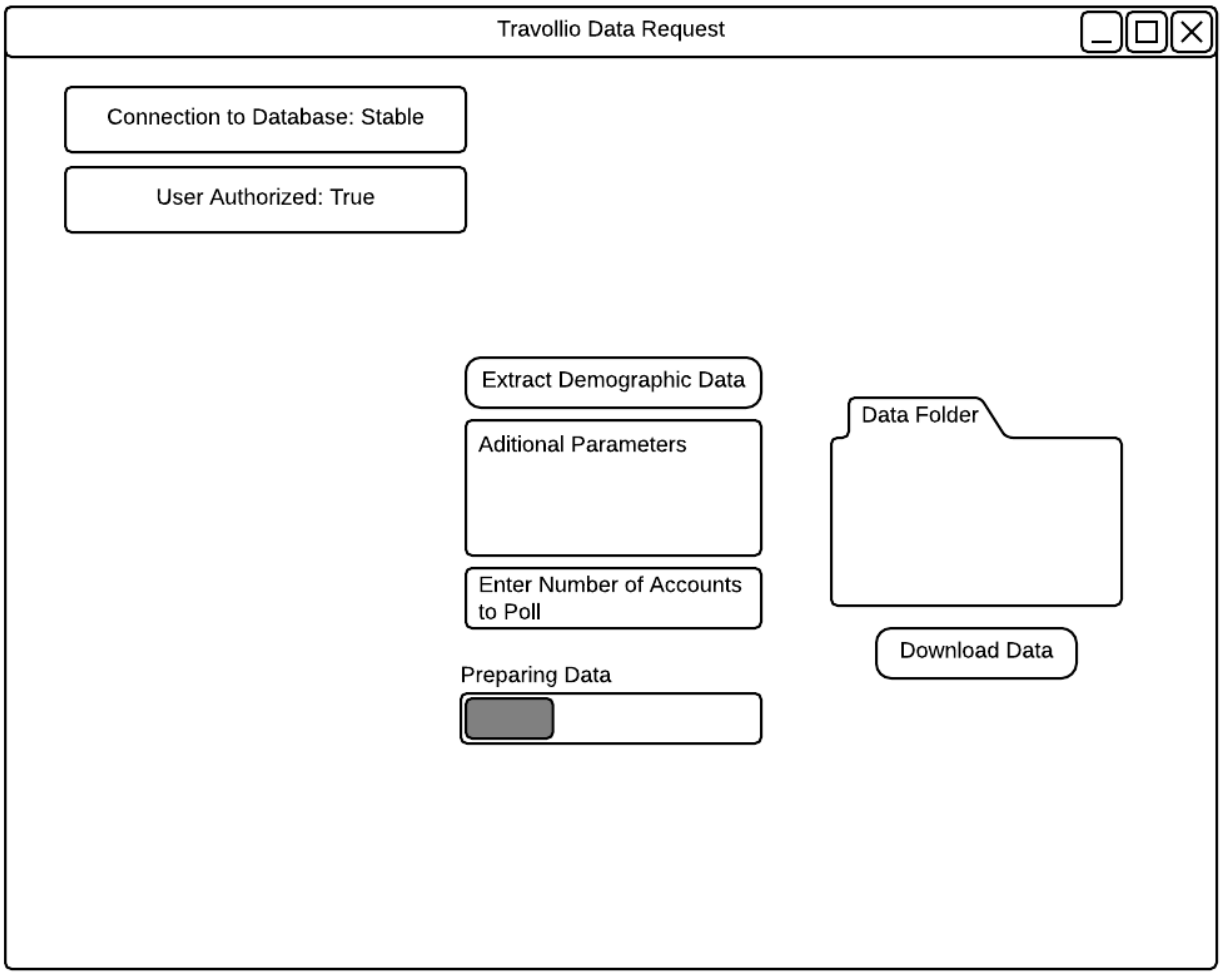
Travollio Application New Journal Page



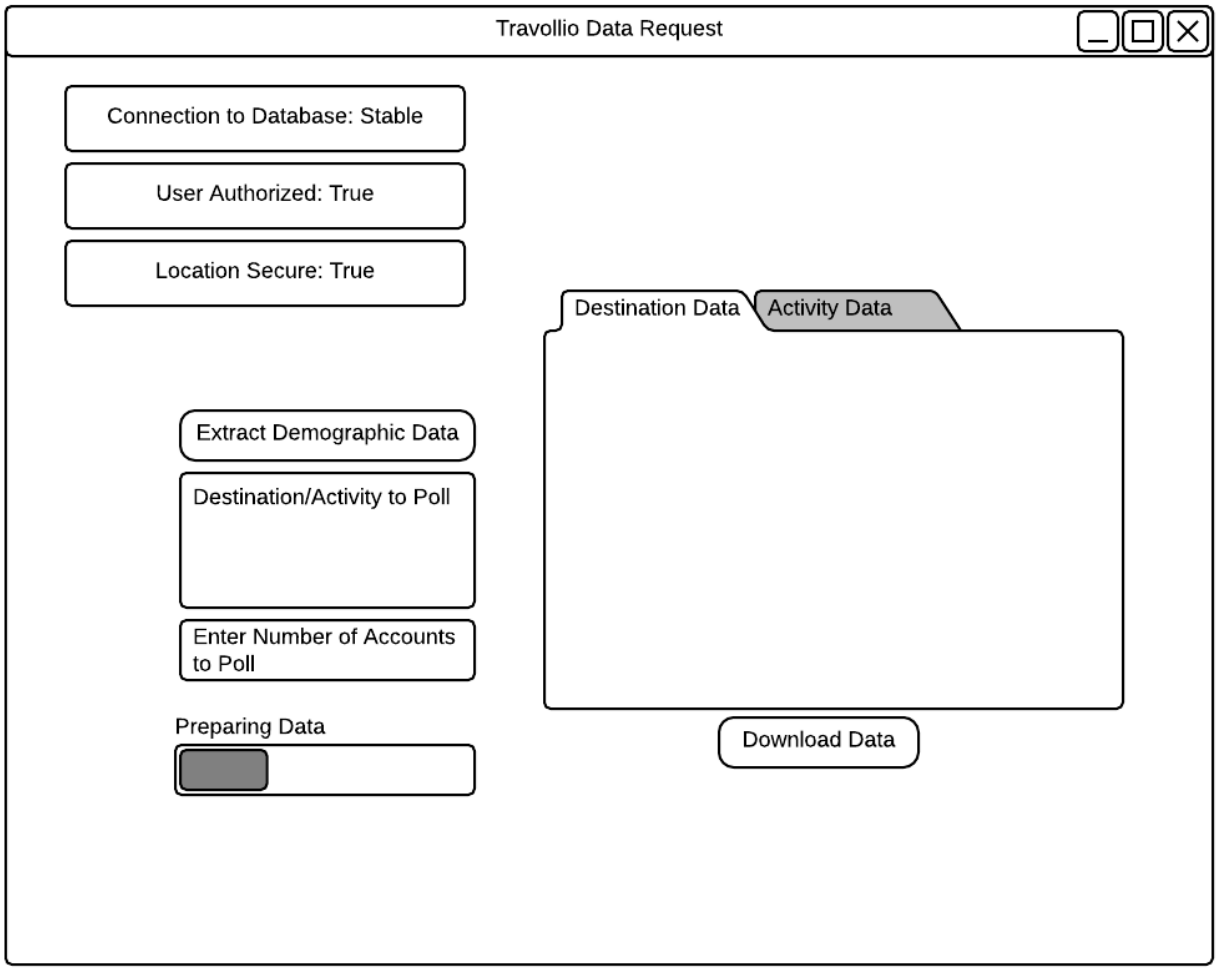
Travollio Application Edit Entry Page



Travollio Website Create Account Page



Travollio Wanderer's Tools Data Request Screen



Travollio Outside Companies Data Request Screen

Appendices

Bibliography

William S. Pfieffer. “A Pocket Guide to Technical Communication”. Upper Saddle River, NJ: Prentice Hall, 2011. Print.

Dennis Wixom and Tegarden “System Analysis and Design: An Object Oriented Approach With UML, 5th Edition”. Hoboken, NJ: Wiley, 2010. Print.

All Charts and Diagrams created with LucidChart https://www.lucidchart.com

Documents Consulted:

Cuisine by Car DESIGN – 2.pdf

Cuisine by Car DESIGN – 1.pdf

5K Foods SPEC – 1.pdf

5K Foods SPEC – 2.pdf

5K Foods SPEC – 3.pdf