**System Design**

**Document**

*Piece of Eden Rentals*

**Client**

John Winder

**Team 4**

Stephen Masterson

Rachel Cohen

Nicholas Keckeisen

Eric Forte

Matthew Walker

4/28/2016

*Piece of Eden Rentals*

System Design Document

**Table of Contents**

1. Introduction

1.1 Purpose of This Document

1.2 References

1. System Architecture

2.1 Architectural Design

2.2 Decomposition Description

1. Persistent Data Design

3.1 Database Descriptions

1. Requirements Matrix
2. Appendix A – Agreement Between Customer and Contractor
3. Appendix B – Team Review Sign-off
4. Appendix C – Document Contributions

**1. Introduction**

***1.1 Purpose of This Document***

The purpose of this document is to describe the design of the Piece of Eden Rentals

application. Key topics covered in this document include the high level system

architecture, lower level class designs, and the persistent data design Piece of Eden Rentals.

***1.2 References***

Throughout this document references will be made to:

Django Project (2016). The web framework for perfectionists with

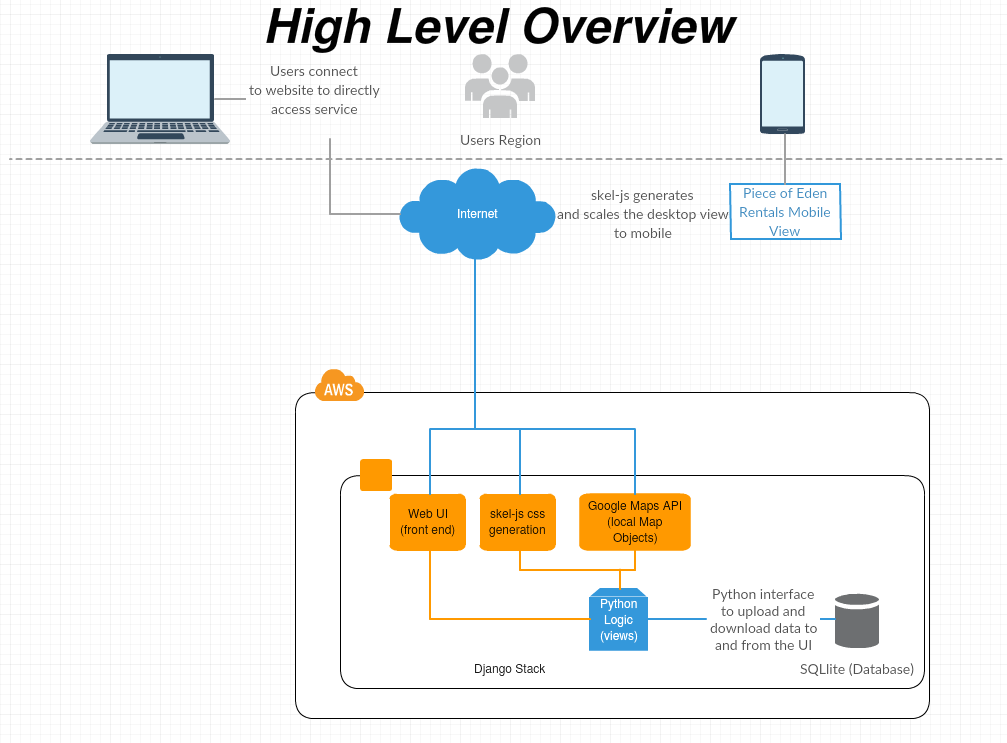
deadlines. Retrieved from “https://www.djangoproject.com/”

**2. System Architecture**

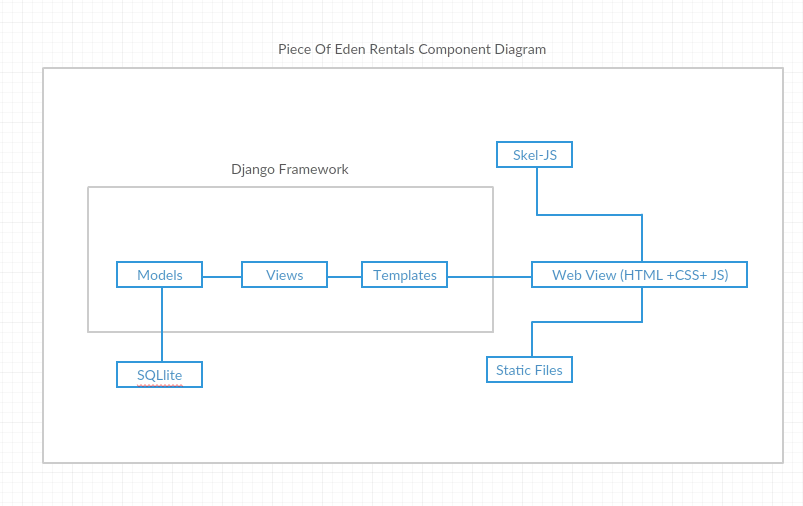
***2.1 Architectural Design***

The Piece of Eden Rentals application can be built and deployed on any machine with the Python 2.7 and the required packaged listed in requirements.txt. The core components are built using the Django framework and a SQLlite database within AWS or any other deployment platform. Initially the product was going to be built with Mongo DB, but in effort to improve usability for our customer we changed to SQLlite to make deployment an easier process.

Figure 1.1



As shown in figure 1.1, users can connect to the Piece of Eden Rental service directly through the web interface on either a desktop web interface, or a mobile friendly interface, both at PieceOfEdenRentals.com. Initially, The APP was going to use a Python REST API integrated into the Django framework to communicate with the database to obtain new information to display in its UI. However, this feature was dropped and moved to a stretch goal as the customer would rather have the development team spend time on making the product easier to deploy and the UI more appealing. Also, by using the skel-js library a mobile view can be generated dynamically from existing desktop views. The web application uses a direct connection to the database through the Django python interface. The database is also integrated directly with the Django python interface and both interactions are described in the figure below.

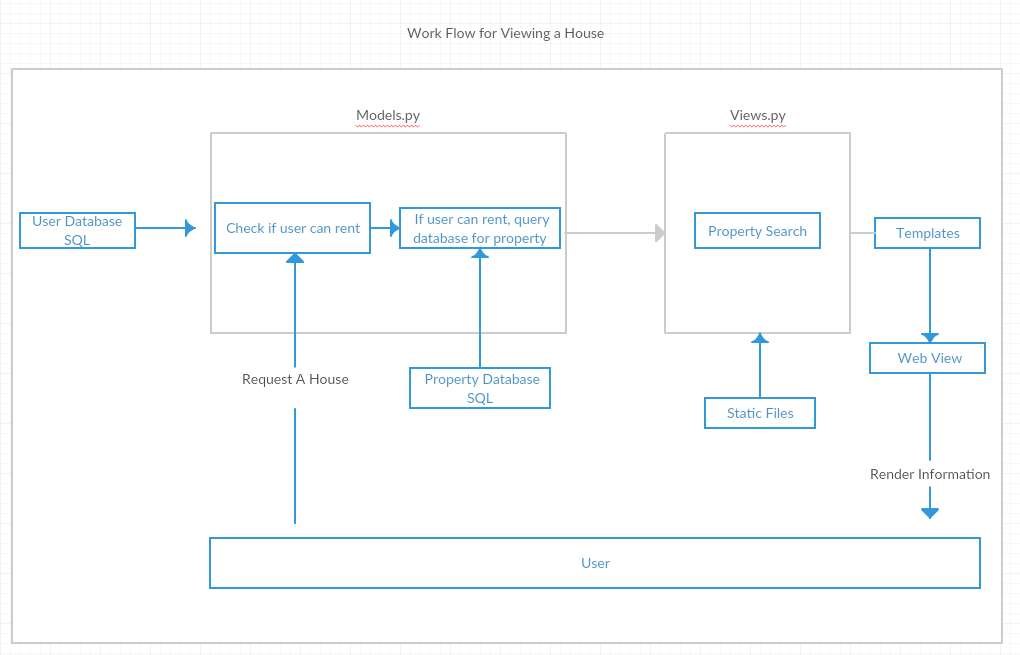


Django Stack Component Diagram (figure 1.2)

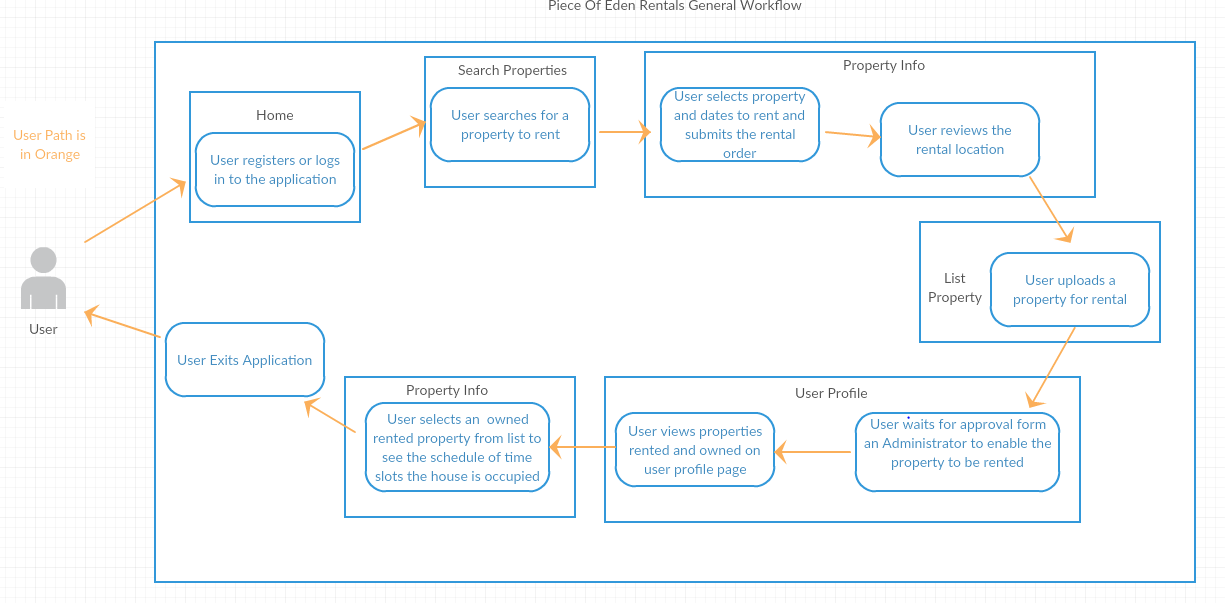
The Django stack uses core components such as Models, Views, and Templates which follows a Model View Controller architecture(MVC). The model layer is used for managing the database, consisting of various classes the correspond to tables in the SQLlite database. These models are used to handle both user data and property data providing flexibility in their access. The views layer is used to manage, abstract, and display standard (HTML, CSS, PHP, and JS) web templates which make up the template layer. The views act as the controller in Django’s MVC as they manage the data flow to and from the user. Additionally, they contain python logic that can manipulate Templates or Models dynamically per user request. What is displayed to the user is the content fetched from the Models and put into templates that are then modified as the situation demands. This allows for features such as browsing, viewing details about a property, and other which will be more fully described later in the document.

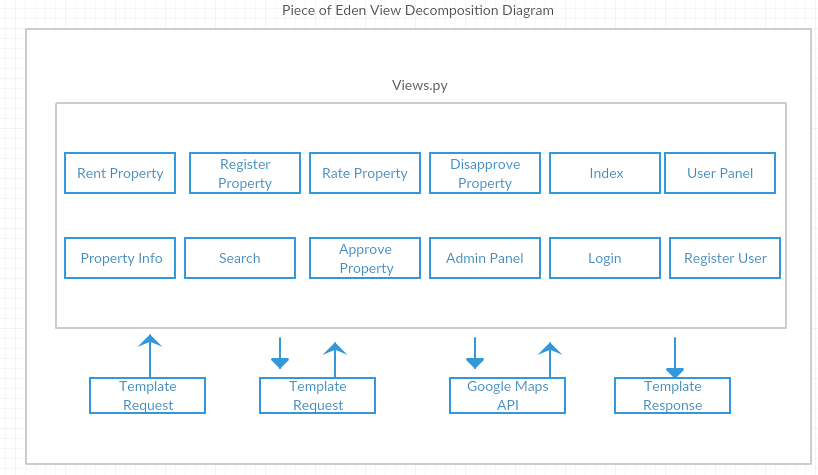
***2.2 Decomposition Description***

The front end web UI will be separated into 4 main pages, a homepage, a user page, a rental submission page (list property), and a rental browsing page. Greater detail as to the design of these pages is on the User Interface Design document; however, the UI will utilize a hybrid of object-oriented and procedural architecture. Each property itself is a python object that gets stored in the database with specific variables such as location of the house, picture links, rating of the house,etc… Each specific page displaying a house will be procedurally generated from the ID of the house the user requested. The generic page will act like a container that gets filled with the responses from the database as to the specific house the user selected. This work flow is shown below:



The models in the application are python classes which are used to generate and manage the SQL database tables. All property fields are publicly accessible; however, user data is only available to other users or the administrator. The models and views are the main structure to each part of the general workflow for using the application. A full user workflow is shown in the figure below. It is important to note that none of the actions the user takes are required, rather this workflow is to exhaustively show actions a user may take or is likely to take. The orange path denotes the users path, the rounded boxes denote the action the user takes, and the cornered boxes denote the view the action takes place in.



The views which serve the data to the web templates, can be accessed arbitrarily and generate responses based upon the input the user requested as displayed in the previous diagrams. Available views and their interaction with the rest of the application are shown below

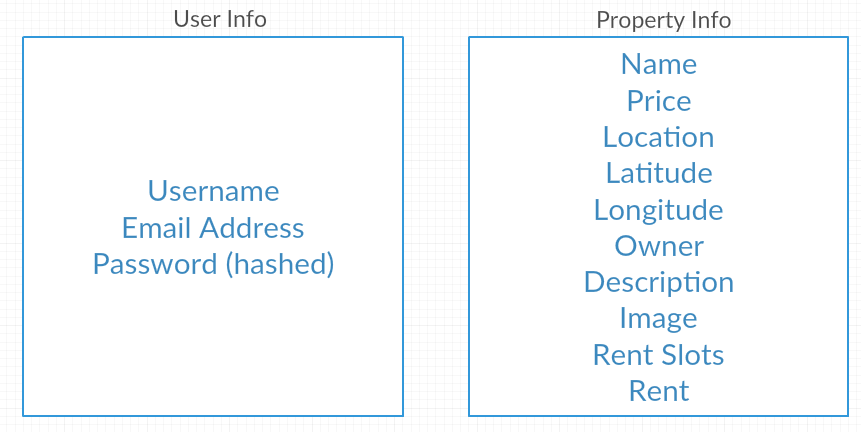
The Python backend will maintain the logic to properly assign the procedurally generated elements of the application to the specific variables that should contain the information to display it properly. Some of this backend also dynamically generates and manages JavaScript for better user interaction. For example, the Python login generates a list of event objects that are inserted into a JavaScript Array of calendar events to display what times a user has booked a given property. Also, the Google Maps API uses JavaScript, handled by python logic, to dynamically update the input form for creating a property, allowing the user to click on the map to select a location to use as the coordinates. Furthermore, the information in the database is updated as necessary using the Django SQLlite Engine’s direct integration tools directly from the python code

**3. Persistent Data Design**

***3.1 Database Descriptions***

The database itself is separated into two parts, a property database which is publicly viewable, and a user database which has security restrictions. The user database information will not be available unless the user is logged in. Furthermore, the user’s themselves are only able to edit their own information. However, the admin user can edit both the user and the property database. An entry-relationship schema is used for both databases and both are run using SQLlite.

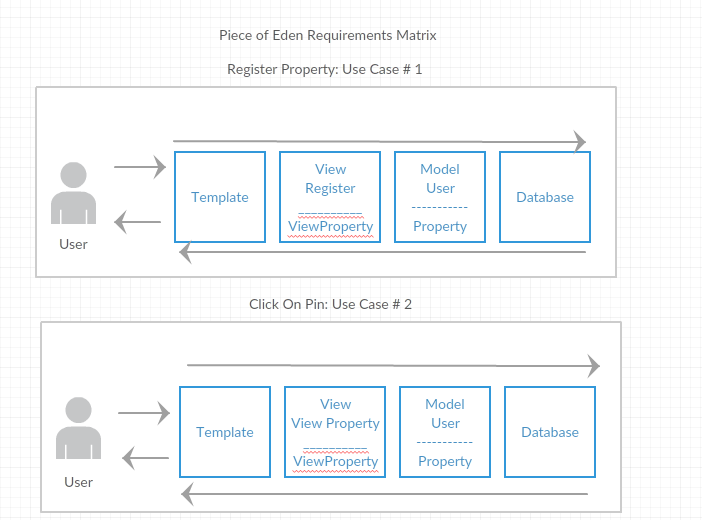
Entry and modification of the database is directly done through python scripting. Django has built in functions for using model abstraction classes to generate the database tables. Access to these tables is then handled through python classes which have specific access abilities to each table. Modifying and reading specific tables requires a unique token to be passed to each user of the site in order to prevent cross site and/or unwanted data manipulation. Furthermore, the python classes allow for direct input validation in an environment where incoming arbitrary data or code does not have the ability to be executed or inserted unless proven valid. Through this the data integrity is maintained throughout the site. The schema diagrams for the user and property database are shown below.

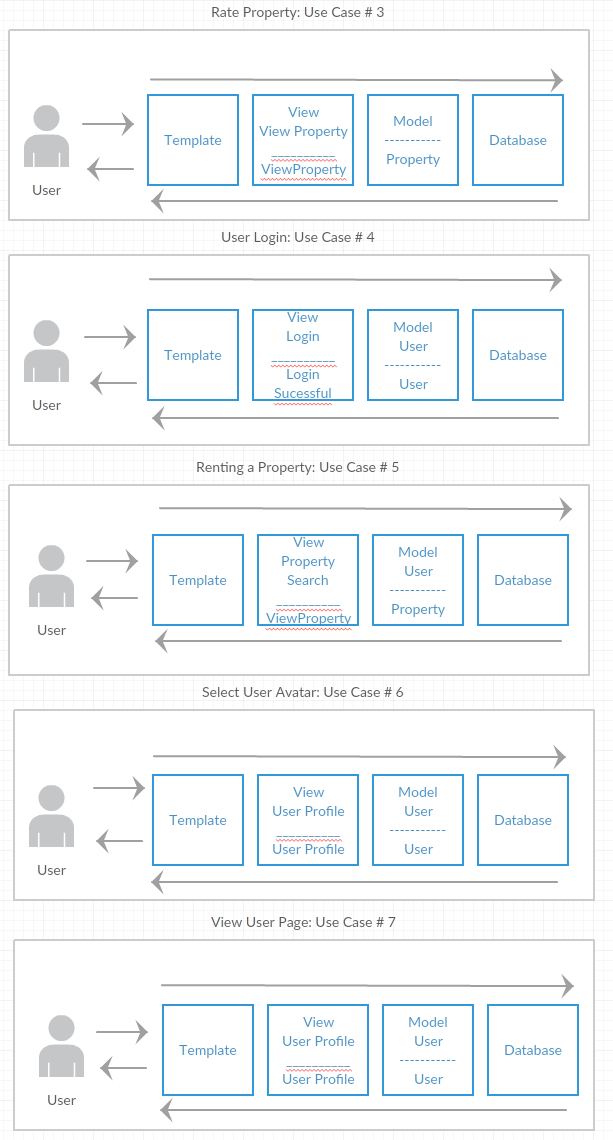


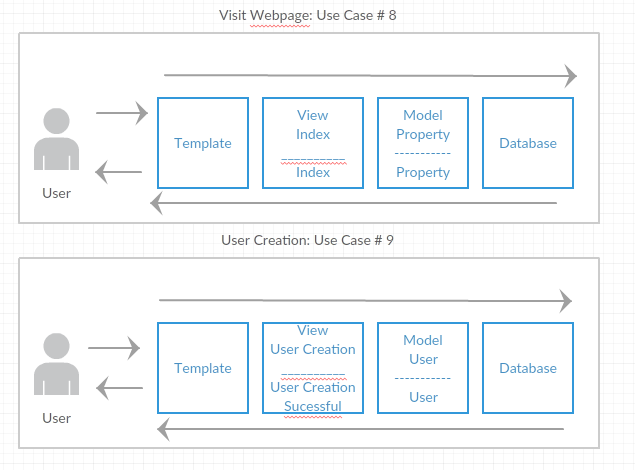
**4. Requirements Matrix**

Please refer to the System Requirements Specification for details regarding the

corresponding use cases.







**5. Appendix A – Agreement Between Customer and**

**Contractor**

The customer agrees to a *Luxury Rental Network* system with searching, browsing and

detailed meta-data capabilities. See System Requirements Specification for more

information. Additional features will be provided in further development spirals.

When and if future changes to this document occur a drafted new document will be

created. Both a hard and electronic copy of both versions will be presented to the client for

review. Upon approval, the draft will be finalized and signed off by both parties.

**Client**

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Print

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

**Team**

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Print

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Print

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Print

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Print

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Print

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Print

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

**6. Appendix B – Team Review Sign-off**

This document has been collaboratively written by all members the team.

Additionally, all team members have reviewed this document and agree on both

the content and the format. Any disagreements or concerns are addressed in

team comments below.

**Team**

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Print

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

Comments

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Print

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

Comments

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Print

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

Comments

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Print

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

Comments

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Print

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

Comments

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Print

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

Comments

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_

**7. Appendix C – Document Contributions**

Eric Forte is the owner of this document and, as of this version, is the sole contributor.