

# **System Requirements Specification**

Piece Of Eden Rentals

4/14/2016

Customer: John Winder

Team 3

Stephen Masterson

Rachel Cohen

Nicholas Keckeisen

Eric Forte

Matthew Walker

## **Table of Contents**

- 1. Introduction
  - 1.1 Purpose of This Document
  - 1.2 Purpose of the Product
  - 1.3 Product Scope
- 2. Functional Requirements
  - 2.1 Use Case 1
  - 2.2 Use Case 2
  - 2.3 Use Case 3
  - 2.4 Use Case 4
  - 2.5 Use Case 5
  - 2.6 Use Case 6
  - 2.7 Use Case 7
  - 2.8 Use Case 8
- 3. Use Case Tests
  - 3.1 Use Case 1 Test - Searching the Database
  - 3.2 Use Case 2 Test - Viewing Detailed Music Meta-data
  - 3.3 Use Case 3 Test - Favorite An Item
  - 3.4 Use Case 4 Test - Rating An Item
  - 3.5 Use Case 5 Test - Reviewing An Item
  - 3.6 Use Case 6 Test - Registering to the Database
  - 3.7 Use Case 7 Test - Logging in to the Database
  - 3.8 Use Case 8 Test - Viewing User Profile
- 4. Non-Functional Requirements
- 5. User Interface
- 6. Deliverables
- 7. Open Issues
- 8. Appendix A – Agreement Between Customer and Contractor
- 9. Appendix B – Team Review Sign-off
- 10. Appendix C – Document Contributions

### **1. Introduction**

Welcome to the Piece of Eden Rentals System Requirements Specifications. Piece of Eden Rentals is a beachfront property rental web application, which allows users to put their beachfront properties up for rent, choose properties to rent, and rate other properties.

#### **1.1 Purpose of This Document**

This document describes how to use the Piece of Eden Rentals web application, as well as the functions and features of the product. The audience for this document is the development team (Team 3) and the customer (John Winder).

#### **1.2 Purpose of the Product**

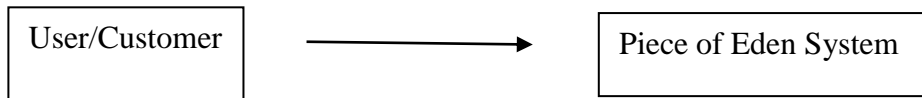
Piece of Eden was designed to provide a service to rent and list beachfront properties all across the world. Users of the product will be able to create a user profile, list a property, search for a

property, and rent a property for a period of time.

### 1.3 Product Scope

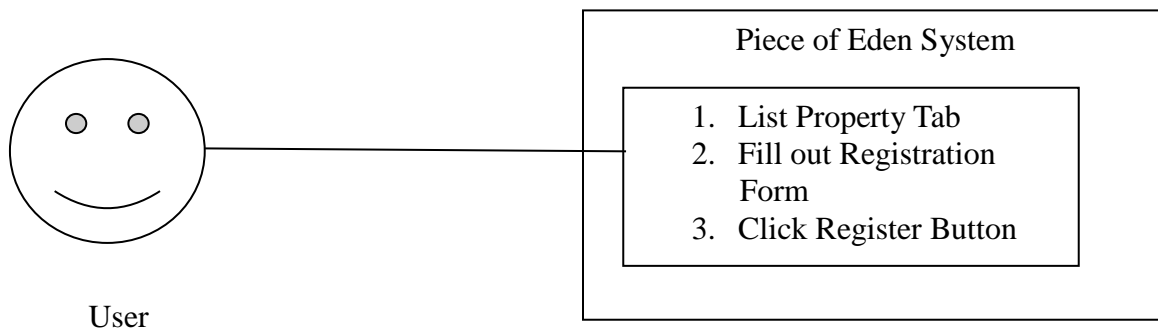
The Piece of Eden web application consists of eight use cases including: registering a property, clicking on a property pin, rating a property, logging in to a user account, renting a property, selecting a user avatar, viewing a user page, visit a webpage, and create a user.

**Figure 1. High level use case context diagram overview**

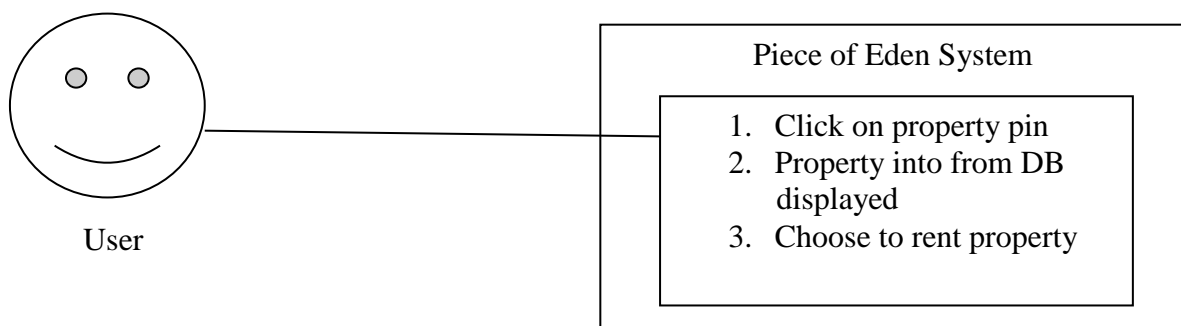


1. Register a Property
2. Click on a Property Pin
3. Rate a Property
4. Logging in to user account
5. Renting a Property
6. Select a user avatar
7. View User Page
8. Visit a webpage
9. Create a user

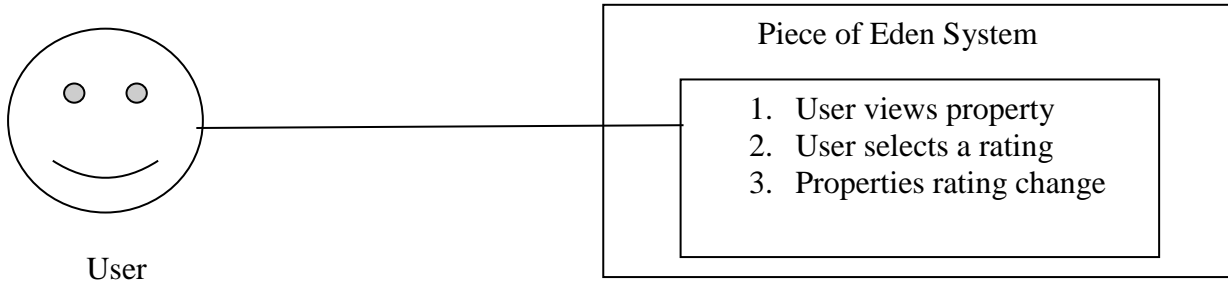
**Figure 2. Use Case 1, Register a Property**



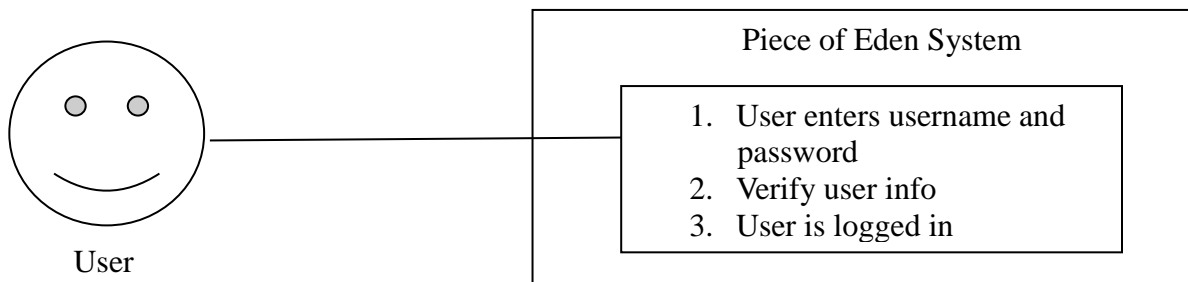
**Figure 3. Use Case 2, Click on a Property Pin**



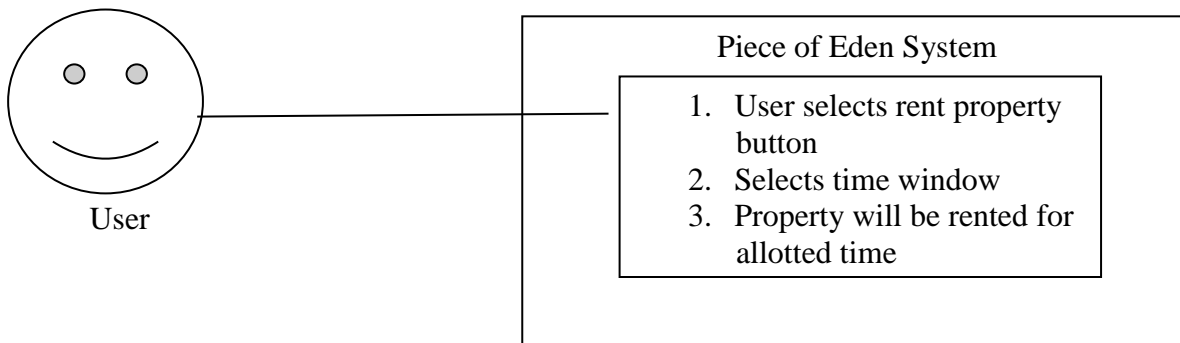
**Figure 4. Use Case 3, Rate a Property**



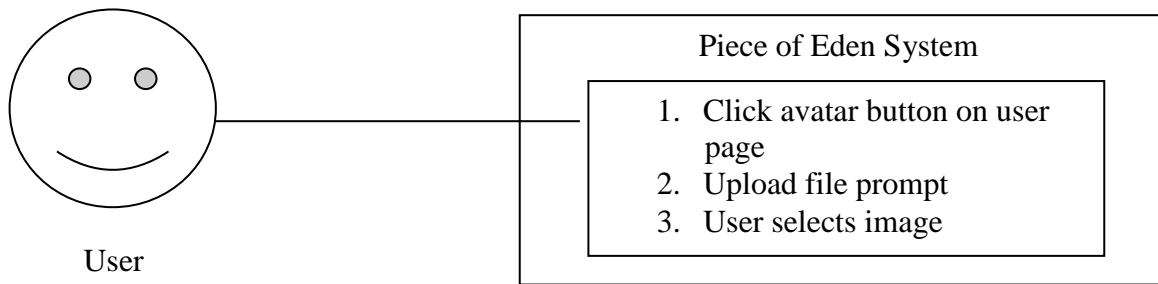
**Figure 5. Use Case 4, Logging in to a user account**



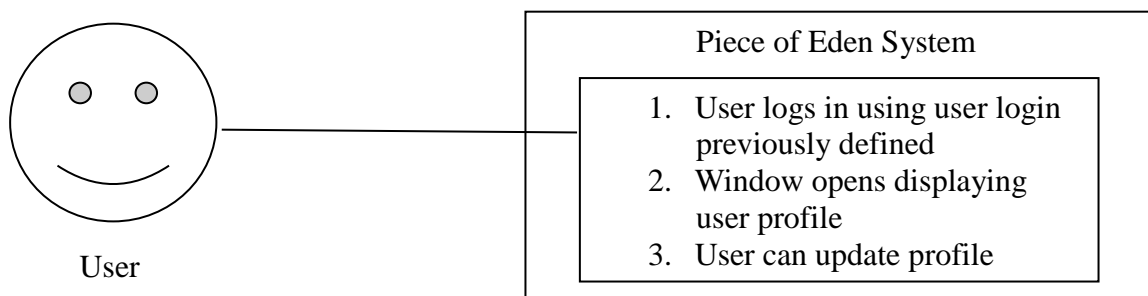
**Figure 6. Use Case 5, Renting a property**



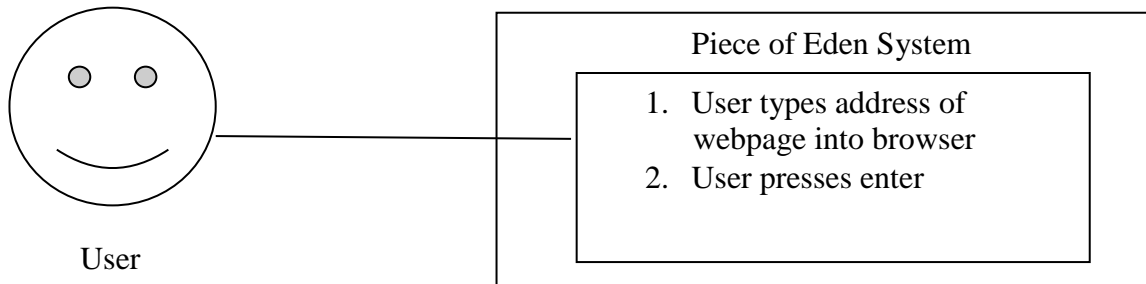
**Figure 7. Use Case 6, Select a user avatar**



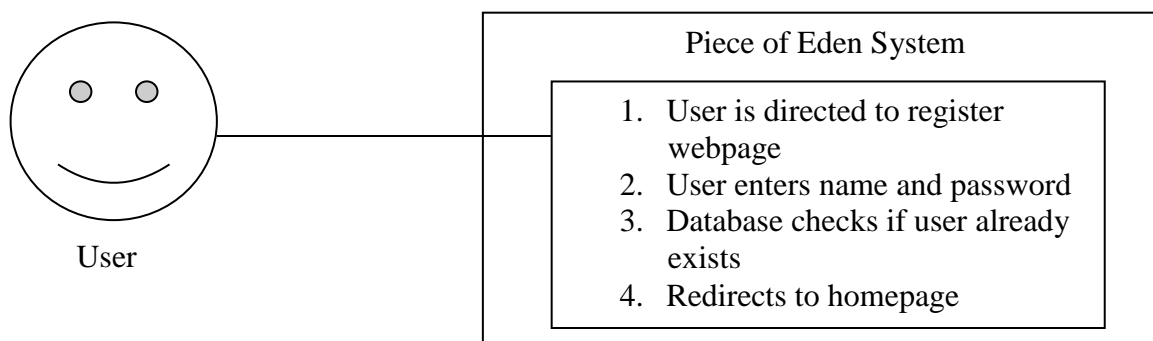
**Figure 8. Use Case 7, View a user page**



**Figure 9. Use Case 8, Visit a webpage**



**Figure 10. Use Case 9, Create a user**



## 2. Functional Requirements

Description of use cases and required functions of the software.

### 2.1 Use case 1

<b>Number</b>	1	
<b>Name</b>	Register Property	
<b>Summary</b>	Register a property for rent. Can include name, location, description and pictures	
<b>Priority</b>	5	
<b>Preconditions</b>	User has an account	
<b>Postconditions</b>	The property will be viewable and rentable by other users	
<b>Primary Actor</b>	Property Owner	
<b>Secondary Actors</b>	Google API, Property Database	
<b>Trigger</b>	Selecting the “Register property” button	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User selects the “Register property” button
	2	Using Google Maps API the user can select location
	3	User is prompted to enter a Name, Description and Pictures of the property
	4	User selects the “Register” button
	5	The property is added to the database and should be viewable to other users
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	2a	User Select “Cancel” button
	2b	User is returned to the home screen
	4a	User Select “Cancel” button
	4b	User is returned to the home screen
	3a	User does not upload pictures
	3b	Continues normally with no pictures for the property
<b>Open Issues</b>		

### 2.2 Use case 2

<b>Number</b>	2	
<b>Name</b>	Click on property pin	
<b>Summary</b>	User clicks on the property’s google map pin	
<b>Priority</b>	5	
<b>Preconditions</b>	A property is registered and displayed on the map	
<b>Postconditions</b>	Window displays (image of house, house style (modern, cottage, etc), availability windows, price, star rating for property	
<b>Primary Actor</b>	User	
<b>Secondary Actors</b>	Property Database	
<b>Trigger</b>	Clicking on property pin	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User clicks on property pin

	2	Window opens and displays the property information from the database
	3	User can choose to rent property, or close window
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	3a	If the user chooses to rent the property, will be taken to a renting property window
	3b	If the user chooses to close the window, they will be returned to the map
<b>Open Issues</b>		

### 2.3 Use case 3

<b>Number</b>	3	
<b>Name</b>	Rate property	
<b>Summary</b>	Rate a property. The property's rating will move up or down depending on the rating	
<b>Priority</b>	4	
<b>Preconditions</b>	User is logged in, property is viewable	
<b>Postconditions</b>	The property's rating changes	
<b>Primary Actor</b>	User	
<b>Secondary Actors</b>	Rating database	
<b>Trigger</b>	User selecting a rating for the property	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User views a property
	2	User selects a rating +1/-1
	3	Properties rating changes accordingly
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	3a	If the user selected +1 adds one to the properties rating
	3b	If the user selected -1 subtracts one to the properties rating
<b>Open Issues</b>		

### 2.4 Use case 4

<b>Number</b>	4	
<b>Name</b>	Login into user account	
<b>Summary</b>	The user logs into his/her account	
<b>Priority</b>	5	
<b>Preconditions</b>	User has created an account	
<b>Postconditions</b>	The user is logged into their account	
<b>Primary Actor</b>	User	
<b>Secondary Actors</b>	User database	
<b>Trigger</b>	The user clicks on the log in button	

<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User enters their username and password
	2	The user's information is verified
	3	The user is logged in
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	2a	The user's login information is incorrect
	2b	The user is returned to the login page
<b>Open Issues</b>		

### 2.5 Use case 5

<b>Number</b>	5	
<b>Name</b>	Renting a property	
<b>Summary</b>	Renting a property. The property should become "rented" by the user, and not be available for other users to rent	
<b>Priority</b>	5	
<b>Preconditions</b>	User has an account, the property is rentable	
<b>Postconditions</b>	The property will be rented by the user	
<b>Primary Actor</b>	User	
<b>Secondary Actors</b>	Google API,Property Database	
<b>Trigger</b>	Selecting the "Rent property" button	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User selects the "Rent property" button
	2	The user will select when they want to rent the property
	3	The property will be rented by that user till their time slot expires
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	2a	The user can select to cancel their renting if they change their mind
<b>Open Issues</b>		

### 2.7 Use case 6

<b>Number</b>	6	
<b>Name</b>	Select User Avatar	
<b>Summary</b>	User receives upload prompt to select an avatar	
<b>Priority</b>	2	
<b>Preconditions</b>	A registered user has logged in	
<b>Postconditions</b>	The image file that the user selected is uploaded to the database as his/her avatar and is displayed as such in references to the user's avatar.	
<b>Primary Actor</b>	User	



<b>Secondary Actors</b>	User Database	
<b>Trigger</b>	Clicking change avatar button	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User clicks the change avatar button on the user page
	2	Prompt opens for the user to upload a file to be used as an avatar
	3	User selects image and it is cropped to fit in avatar pane
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	3a	If the user selects an invalid file, the prompt will return with invalid file and restore the user's previous avatar.
<b>Open Issues</b>		

## 2.8 Use case 7

<b>Number</b>	7	
<b>Name</b>	View User Page	
<b>Summary</b>	User views a personalized page showing rentals and settings	
<b>Priority</b>	5	
<b>Preconditions</b>	A registered user has logged in	
<b>Postconditions</b>	A personalized screen appears with user information such as username, properties rented before, etc... Also, personalization options are available at this screen.	
<b>Primary Actor</b>	User	
<b>Secondary Actors</b>	User Database	
<b>Trigger</b>	Clicking on the login button	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User logs in using User login previously defined
	2	Window opens and displays the user's profile
	3	User can choose to update information in profile
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	3a	If the user chooses to update password information a new screen will launch.
	3b	If the user chooses to update their avatar they will be prompted with a new action to do so.
	3c	If the user decides to log out then the user will be returned to the home screen.
<b>Open Issues</b>		

## 2.9 Use case 8

<b>Number</b>	9	
<b>Name</b>	Visit the Homepage	
<b>Summary</b>	View the landing page (index.html)	
<b>Priority</b>	5	
<b>Preconditions</b>	none	

<b>Postconditions</b>	The website's homepage will display in browser	
<b>Primary Actor</b>	Any user	
<b>Secondary Actors</b>	Google API, django, mongodb,	
<b>Trigger</b>	Selecting the "Register property" button	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User types address of webpage into browser
	2	User presses enter
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
		None
<b>Open Issues</b>	Website is currently locally hosted	

## 2.10 Use case 9

<b>Number</b>	9	
<b>Name</b>	Register new user account	
<b>Summary</b>	The user creates an account with a password	
<b>Priority</b>	5	
<b>Preconditions</b>	None	
<b>Postconditions</b>	The User's account has been created	
<b>Primary Actor</b>	User	
<b>Secondary Actors</b>	User database	
<b>Trigger</b>	The user clicks on the "Register" button	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User is directed to Register webpage
	2	User enters their username and password
	3	Database checks if user exists, adds user
	4	Redirects to Home
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	3a	The user exists in the database
	3b	The user is returned to register page, error message displayed
<b>Open Issues</b>		

## 3. Functional Requirements

### 3.1 Use case 1 Test – Registering a Property

Register a property with a name, location, description and picture. See if these items appear in the property database and is viewable to other users

### 3.2 Use case 2 Test – Click on property pin

Click on a pin and see that the image of house, house style (modern, cottage, etc), availability windows, price, star rating for the property.

### 3.3 Use case 3 Test – Rate property

Rate a property up or down and see that the rating registers with the ratings database.

### 3.4 Use case 4 Test – Login into user account

Login into an existing account. Make sure the database verifies the account so it can only be accessed with the correct password.

### 3.5 Use case 5 Test – Rent a property

Rent a property. Verify the database changes for that property, and that the property is not rentable by other users.

### 3.7 Use case 6 Test – Select User Avatar

Upload an image as an avatar. Verify the image has changed and is viewable to other users.

### 3.8 Use case 7 Test – View User Page

Connect to a user's page and view their information. Verify the information is correct on the database and viewable on the application.

### 3.9 Use case 8 Test – Visit Homepage

User types address of webpage into browser and presses enter to visit the homepage.

### 3.10 Use case 9 Test – Register new account

The user is directed to the register webpage where they enter their username and password. The database checks to see if the user already exists. If not, the user is added and redirected to home.

#### 4. Non-Functional Requirements

Below outlines criteria used to judge the system as a whole (priority 1 = lowest, 5 = highest).

#	Requirement	Priority
1	Uses the google maps API	5
2	Users can create accounts and login	5
3	Properties can be viewed, rented and rated	4
4	Users can put properties up for renting	4
5	Users can upload avatars	2
6	System should use the model view controller	3
7	Application should be extensible	3
8	Application should be maintainable	3
9	Application should be portable	1
10	Code should be well documented	2

#### 5. User Interface

See “User Interface Design Document for Piece of Eden Rentals”

#### 6. Deliverables

Deliverables include:

- Systems Requirements Specifications
- System Design Document
- User Interface Design Document
- Bi-Weekly Status reports

A Zip File of

- Systems Requirements Specifications
- System Design Document
- User Interface Design Document
- User Manual
- All source code
- The executable program

#### 7. Open Issues

Issues that have not been resolved as of the last commit of this document.

Issue	Scheduled for
-Nonfunctional requirements	Ongoing
-More robust rental system	Spiral 3
-Duplicate checking in DB	Spiral 3
-Allow user to delete content once posting	Spiral 3

## 8. Customer Agreement

The customer agrees to the capabilities of “Piece of Eden rentals” specified in the use cases and test cases outlined in this document.

When and if changes occur, a new draft will be printed and signed by the customer and Team 3.

### Client

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

Print

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

Signature

### Team

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

Print

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

Signature

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

Print

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

Signature

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