System Requirements Specification

XENO

**Client**

**Sha**w**n Squire**

**Team 1**

Vesh Bhatt

Michael Bishoff

Edward LaFemina

Michael Lee

John Swank

2 March 2015

**Table of Contents**

1. Introduction
   1. Purpose of This Document
   2. References
   3. Purpose of the Product
   4. Product Scope
2. Functional Requirements
   1. Use Case 1: Search Car Database
   2. Use Case 2: Log into Website
   3. Use Case 3: View User Profile
   4. Use Case 4: Create Account
   5. Use Case 5: Add Car
   6. Use Case 6: Show/Hide/Remove Car
   7. Use Case 7: Account Creation Management
   8. Use Case 8: Suspend/Ban Account
   9. Use Case 9: Maintenance Disable/Enable Car
   10. Use Case 10: Check in Car
   11. Use Case 1l: View Maintenance Log
   12. Use Case 12: Reserve Car
   13. Use Case 13: Twitter Integration
3. Non-Functional Requirements
4. User Interface
5. Deliverables
6. Open Issues

Appendix A - Agreement Between Customer and Contractor

Appendix B - Team Review Sign-off

Appendix C - Document Contributions

Welcome to the XENO system requirements specification. XENO is an online exotic car rental service.

**1. Introduction**

**1.1 Purpose of This Document**

This document is designed to explain the features of the XENO web application, its functions, and the conditions required for operation. The intended audience is the XENO development team as well as the faculty customer, Shawn Squire.

**1.2 References**

* First Customer Meeting Notes.docx
  + Title: First Customer Meeting Notes
  + Author: Team XENO
  + Date: February 20, 2015
* XENO Product
  + Title: XENO
  + Author: Team XENO
  + Date: February 12, 2015
  + URL: http://userpages.umbc.edu/~bishoff1/xeno

**1.3 Purpose of the Product**

XENO was designed to provide exclusive clientele with the ability to reserve the use of an exotic car of their preference through an online website or Twitter.

**1.4 Product Scope**

The XENO application is comprised of thirteen use cases. Please refer to the “System Design Document” for more information as to how the use cases will be reflected in the system.

**2. Functional Requirements**

This section documents the use cases for XENO, which describe the various functionalities of the product and how a user will be able to access them.

To see how each use case will be tested, please see the “Testing Document”.

|  |  |  |
| --- | --- | --- |
| **Number** | 1 | |
| **Name** | Search Car Database | |
| **Summary** | User can enter search terms and find matching cars | |
| **Priority** | 2 (1 is highest, 5 is lowest) | |
| **Preconditions** | There are cars in the database; User is logged in | |
| **Postconditions** | A list of cars with attributes matching the search terms is displayed | |
| **Primary Actor** | The user | |
| **Secondary Actors** | Database querying mechanism | |
| **Trigger** | Button click or ‘Enter’ key pressed | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | User enters search terms |
|  | 2 | Trigger is activated |
|  | 3 | System displays matching cars from database in an orderable list |
| **Extensions** | **Step** | **Branching Action** |
|  | 3a | No matching cars in database: Display message |
| **Open Issues** | What attributes can a user search on? What form fields should there be to allow user to enter better search terms? Have a simple and advanced search? Which attributes should the list display? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 2 | |
| **Name** | Log into Website | |
| **Summary** | Allow user to log into website | |
| **Priority** | 1 (1 is highest, 5 is lowest) | |
| **Preconditions** | None | |
| **Postconditions** | User credentials are verified and site is personalized for user | |
| **Primary Actor** | User | |
| **Secondary Actors** | Credential Validation System | |
| **Trigger** | Button click or ‘Enter’ key pressed | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | User enters username and password |
|  | 2 | Trigger is activated |
|  | 3 | Username and password sent to server for validation |
|  | 4 | User’s profile and preferences are loaded into site |
| **Extensions** | **Step** | **Branching Action** |
|  | 4a | Invalid credentials: Display message and ask for credentials again |
| **Open Issues** | How should the logon screen look? Should there be a ‘Remember me’ option? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 3 | |
| **Name** | View User Profile | |
| **Summary** | Allow user to see his or her profile and personal information | |
| **Priority** | 3 (1 is highest, 5 is lowest) | |
| **Preconditions** | User is logged in; User has entered some profile information | |
| **Postconditions** | User’s personal information is displayed clearly | |
| **Primary Actor** | User | |
| **Secondary Actors** | User validation system; Profile retrieving system | |
| **Trigger** | User Icon click | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Trigger is activated |
|  | 2 | User is validated and profile information displayed |
| **Extensions** | **Step** | **Branching Action** |
|  | 2a | No profile information on record: Prompt user to enter information |
| **Open Issues** | Should we let users edit this information once it is entered? What can other users, admins, and maintenance see? Can user change what others can see (except admin)? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 4 | |
| **Name** | Create Account | |
| **Summary** | Allow someone to create an account | |
| **Priority** | 1 (1 is highest, 5 is lowest) | |
| **Preconditions** | Person does not have an account already | |
| **Postconditions** | An account with the correct type is added | |
| **Primary Actor** | Person creating account | |
| **Secondary Actors** | Email Verification and Information Storage Systems | |
| **Trigger** | Button click or ‘Enter’ key pressed | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Person clicks ‘Create New Account’ |
|  | 2 | Person enters all applicable information |
|  | 3 | Email is sent to validate email address |
|  | 4 | Email is validated and account created |
|  | 5 | User is directed to logon screen |
| **Extensions** | **Step** | **Branching Action** |
|  | 3a | For creating admin account: Need to verify with the existing admin account if the Person is supposed to get admin privileges |
|  | 3b | For creating maintenance account: Need to verify with existing admins if the Person is supposed to get maintenance privileges. |
|  | 4a | Admin reviews account and accepts or deletes account |
| **Open Issues** | How to tell user their account was not accepted by admins? What information is required and what is optional? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 5 | |
| **Name** | Add Car | |
| **Summary** | Add a car to the database | |
| **Priority** | 1 (1 is highest, 5 is lowest) | |
| **Preconditions** | Must be logged in on an admin account | |
| **Postconditions** | Car is added to database and made available to rent | |
| **Primary Actor** | Admin | |
| **Secondary Actors** | Information Storage System | |
| **Trigger** | Button click or ‘Enter’ key pressed | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Admin navigates to ‘Add Car’ page |
|  | 2 | Enter any applicable details |
|  | 3 | Data sent to database and stored |
| **Extensions** | **Step** | **Branching Action** |
|  | 1a | Not logged in as admin: Prompt for login |
| **Open Issues** | What details should there be about the car? Which are required? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 6 | |
| **Name** | Show/Hide/Remove Car | |
| **Summary** | Remove, hide, un-hide a car in the system | |
| **Priority** | 4 (1 is highest, 5 is lowest) | |
| **Preconditions** | There is a car in the system; Must be logged in on an admin account | |
| **Postconditions** | The car is removed from system or hidden/shown from ordinary users’ view | |
| **Primary Actor** | Admin | |
| **Secondary Actors** | Database Query and Information Storage Systems | |
| **Trigger** | Button click | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Admin navigates to ‘Hide/Remove’ page |
|  | 2 | Admin searches for car |
|  | 3 | Admin selects either ‘Remove’, ‘Hide’, or ‘Show’ |
|  | 4 | Database entry for car updates |
| **Extensions** | **Step** | **Branching Action** |
|  | 1a | Not logged in as admin: Prompt for login |
|  | 2a | No car matching search terms: Display message |
|  | 3a | Car is hidden and ‘hide’ is selected: Display unobtrusive message |
|  | 3b | Car is visible and ‘show’ is selected: Display unobtrusive message |
| **Open Issues** | Should a reason for removal or hiding be required? Should car be totally erased from system or just marked as permanently unavailable? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 7 | |
| **Name** | Account Creation Management | |
| **Summary** | Allow Admin to approve the creation of a User or Maintenance Account | |
| **Priority** | 2 (1 is highest, 5 is lowest) | |
| **Preconditions** | An account must be created and Admin notified | |
| **Postconditions** | The account is either approved or deleted by the Admin | |
| **Primary Actor** | Admin | |
| **Secondary Actors** | Account Management System | |
| **Trigger** | Button click | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Admin navigates to ‘Account Management’ page |
|  | 2 | Admin selects ‘Approve’ or ‘Remove’ from list of newly created accounts |
|  | 3 | Notification sent to account holder |
| **Extensions** | **Step** | **Branching Action** |
|  | 1a | Not logged in as admin: Prompt for login |
|  | 2a | Account rejected: Email sent to account holder’s email address and account is rendered inactive |
| **Open Issues** | Should a reason for rejection be required? On what conditions should an account be rejected or approved? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 8 | |
| **Name** | Suspend/Ban Account | |
| **Summary** | Admin can either suspend an account for a period of time or ban it completely | |
| **Priority** | 5 (1 is highest, 5 is lowest) | |
| **Preconditions** | An account exists | |
| **Postconditions** | The account is inactive for a period of time or permanently | |
| **Primary Actor** | Admin | |
| **Secondary Actors** | Account Management System | |
| **Trigger** | Button click | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Admin navigates to ‘Account Management’ page |
|  | 2 | Admin selects ‘Suspend’ or ‘Ban’ from list of accounts |
|  | 3 | Notification sent to account holder |
| **Extensions** | **Step** | **Branching Action** |
|  | 1a | Not logged in as admin: Prompt for login |
|  | 2a | Account banned: Notification sent to account holder and a permanent mark on account is made to indicate it is disabled |
|  | 2b | Account suspended: Notification sent to account holder and a temporary flag is set to indicate account disabled until a certain time |
| **Open Issues** | Should a reason for suspense or banning be required? How long does a suspension last? What constitutes a suspension or ban? Can account holders appeal decision? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 9 | |
| **Name** | Maintenance Disable/Enable Car | |
| **Summary** | Maintenance can disable a car if it needs to be worked on or enable it when it is ready | |
| **Priority** | 5 (1 is highest, 5 is lowest) | |
| **Preconditions** | Car is in database | |
| **Postconditions** | Car is marked as disable for maintenance and cannot be reserved; car enabled and can be reserved | |
| **Primary Actor** | Maintenance | |
| **Secondary Actors** | Information Storage System | |
| **Trigger** | Button click | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Maintenance selects a car |
|  | 2 | Trigger is activated |
|  | 3 | Car appears with a maintenance flag or the flag is cleared |
| **Extensions** | **Step** | **Branching Action** |
|  | 3a | Car is already enabled or disabled and maintenance tries to enable or disable: Display message |
| **Open Issues** | Should there be a limit on how long car is disabled? What happens if someone reserved the car for a future date and it is still disabled then? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 10 | |
| **Name** | Check in Car | |
| **Summary** | Maintenance or Admin can check in a car when it has been verified returned. | |
| **Priority** | 1 (1 is highest, 5 is lowest) | |
| **Preconditions** | Car has been checked out | |
| **Postconditions** | Car is either checked in and available or maintenance disabled | |
| **Primary Actor** | Maintenance | |
| **Secondary Actors** | Information Storage System | |
| **Trigger** | Button Click | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Maintenance looks at list of returned cars |
|  | 2 | Trigger activated |
|  | 3 | Car appears as ready to be checked out again |
| **Extensions** | **Step** | **Branching Action** |
|  | 2a | Car needs to be put in maintenance: Follow maintenance disable procedure |
| **Open Issues** | How long between car being returned and auto-check in if maintenance doesn’t get to it? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 11 | |
| **Name** | View Maintenance Log | |
| **Summary** | Allow Admin and Maintenance to see the maintenance logs for each car | |
| **Priority** | 5 (1 is highest, 5 is lowest) | |
| **Preconditions** | Car is in database | |
| **Postconditions** | Maintenance log has been viewed | |
| **Primary Actor** | Admin or Maintenance | |
| **Secondary Actors** | Database Query System | |
| **Trigger** | Button click | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Admin or Maintenance navigates to ‘Maintenance Log’ page |
|  | 2 | Select a car |
|  | 3 | Selection is sent to database and |
| **Extensions** | **Step** | **Branching Action** |
|  | 1a | Not logged in as admin or maintenance: prompt for login |
| **Open Issues** | What details should be displayed? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 12 | |
| **Name** | Reserve Car | |
| **Summary** | Allow a user to reserve a car for a day and check it out | |
| **Priority** | 1 (1 is highest, 5 is lowest) | |
| **Preconditions** | User is logged in; Cars are available | |
| **Postconditions** | Car is marked as reserved for that day and | |
| **Primary Actor** | User | |
| **Secondary Actors** | Calendar and Information Storage Systems | |
| **Trigger** | Button click | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | User selects car |
|  | 2 | User browses for and selects date on calendar |
|  | 3 | Reservation is made |
| **Extensions** | **Step** | **Branching Action** |
|  | 3a | Car is already reserved: Display message and prompt for a near by date |
| **Open Issues** | Should there be any kind of queue? Does it cost anything to make a reservation? | |

|  |  |  |
| --- | --- | --- |
| **Number** | 13 | |
| **Name** | Twitter Integration | |
| **Summary** | Allow users to access basic functionality of the site through Twitter | |
| **Priority** | 3 (1 is highest, 5 is lowest) | |
| **Preconditions** | User has registered Twitter handle with XENO | |
| **Postconditions** | User has completed the desired task | |
| **Primary Actor** | User | |
| **Secondary Actors** | Twitter Integration System | |
| **Trigger** | Tweet sent | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | User sends Tweet |
|  | 2 | Twitter Integration System parses Tweet |
|  | 3 | The desired action is taken according to other use cases |
|  | 4 | The information requested, or confirmation of action is sent back to user in a Tweet |
| **Extensions** | **Step** | **Branching Action** |
|  | 1a | No Twitter handle registered: Error message and prompted to register one |
|  | 2a | Tweet is in an invalid format or missing information: Tweet error message and possible correct formats |
| **Open Issues** | How often can someone Tweet to XENO? What functionality should be Twitter-enabled? What should the format of Tweets be? | |

**3. Non-Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **Item Number** | **Item** | **Priority**  **1 (highest) to 5 (lowest)** |
| 1 | The Database shall be secure | 1 |
| 2 | Web Application should be easy to use | 1 |
| 3 | The server side system shall be written in Python | 1 |
| 4 | The system shall use the Flask library | 1 |
| 5 | The system shall use the Model-View-Controller architecture | 1 |
| 6 | The front end system shall use the latest markup and styling specifications | 2 |
| 7 | The Web Application will be compatible with all major browsers | 3 |
| 8 | All code should be well documented | 2 |
| 9 | Exotic and Foreign cars must be featured | 5 |
| 10 | Web Application must appeal to a wealthy and exclusive audience | 4 |
| 11 | Web Application must be mobile-friendly | 5 |
| 12 | Tweets should be easy to form without assistance by the user | 4 |
| 13 | Tweets sent to users should be easy to understand and with complete information | 4 |

The table below corresponds to the table above.

|  |  |
| --- | --- |
| **Item Number** | **Verification method for Item Completion** |
| 1 | The Testing and Implementation leader will test the XENO application against SQL injection attacks to ensure improper commands do not execute in the Database. All SQL query statements will also be reviewed on a bi-weekly basis to ensure the String substitution operator is used (“%s”) instead of the String concatenation (“+”) operator. |
| 2 | The XENO team will use its expertise in UI design to develop an initial UI Design document, present it to the customer and revise it as needed. They will then code the frontend of the application as discussed to ensure the application is intuitive to use and requires minimal training for users. |
| 3 | The Testing and Implementation leader will verify that Python is being utilized to its full potential and conforms to the language’s standards on a bi-weekly basis. |
| 4 | The Testing and Implementation leader will verify the Python Flask library is being utilized to its full potential and follows conventions on a bi-weekly basis. |
| 5 | The Testing and Implementation leader will work with other engineers on a daily basis to modularize the application into the Model-View-Controller architecture and prevent duplication of code. |
| 6 | The XENO team will refer to the documentation of markup and styling libraries utilized to confirm the functionality they provide is supported on the latest browsers. |
| 7 | The XENO team will employ the latest stable release of all applications and libraries that are certified to work across Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox. |
| 8 | The XENO team will add comments to describe the functionality of the code throughout the development of the application. The Testing and Implementation leader will also review the code on a bi-weekly basis and instruct the author(s) of the code to add or revise comments as needed. |
| 9 | The XENO team will implement a simple way to mark cars as exotic and foreign. They will then give an overview of this feature to the customer. |
| 10 | The XENO team will utilize the Internet to conduct a simple market survey to find ideal design guidelines that appeal to wealthy and exclusive audiences. These design guidelines will be showcased in the UI Design document, presented to the customer, revised as discussed, and implemented. |
| 11 | The XENO team will integrate Twitter Bootstrap into the application so it will adjust its layout, but maintain ease of use, to conform to the various resolutions encountered on mobile devices. |
| 12 | The XENO team will setup a meeting with the customer to construct a Tweet API that will feel natural to users and feel as if they are Tweeting a human to interact with the application. |
| 13 | The XENO team will setup a meeting to develop sample Tweet formats with the customer and then revise them so the Tweets are concise and to the point while still maintaining the message they are trying to get across to the user. |

**4. User Interface**

See “User Interface Design Document” for XENO.**5. Deliverables**

These items will be delivered to the customer in the specified format:

Hard Copy:

* System Requirements Specification
* System Design Document
* User Interface Design Document
* Testing Report
* Code Inspection Report

An electronic copy containing the following:

* System Requirements Specification
* System Design Document
* User Interface Design Document
* Testing Report
* Code Inspection Report
* All source code

**6. Open Issues**

These are known issues that have yet to be completely solved but are scheduled to be fixed before the specified spiral ends.

|  |  |
| --- | --- |
| **Issues** | **Scheduled For** |
| Database Design | Spiral 2 |
| What attributes a user can search on | Spiral 2 |
| Login screen visuals | Spiral 2 |
| Permissions for editing user profiles | Spiral 3 |
| Denied account notification | Spiral 3 |
| Required car details | Spiral 2 |
| Car availability implementation | Spiral 3 |
| Account suspension length | Spiral 3 |
| Twitter implementation | Spiral 2 |
| Featured car of the day logic | Spiral 2 |
| Credit System logic | Spiral 3 |

**Appendix A – Agreement Between Customer and Contractor**

The contractor, XENO, has met with customer Shawn Squire on February 13, 2015 to discuss what web application XENO will be creating. During the meeting, both parties agreed on what must be done and have informed one another about the procedure that must take place in the event there is a change to the document. In the event where there must be changes made to the document, XENO will email Shawn Squire with a draft of the modified document immediately, asking for approval.

**Customer: Shawn Squire**

Email: ssquire1@umbc.edu

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Team: XENO**

Name: Vesh Bhatt

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Michael Bishoff

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Edward LaFemina

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Michael Lee

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: John Swank

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Appendix B – Team Review Sign-off**

As of 3/2/2015, all group members of XENO have reviewed the System Requirements Specifications (SRS) document and all have agreed the content and format of this document is correct. This document has been tailored to meet the client’s necessities for an online web application which will offer an online exotic car rental service to VIP customers.

**Team: XENO**

Name: Vesh Bhatt

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Michael Bishoff

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Edward LaFemina

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Michael Lee

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: John Swank

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Appendix C – Document Contributions**

Vesh Bhatt: Contributed checks for thoroughness and consistency with other documents. Also added the verification and implementation of functional requirement goals. Estimated percent of work is about 5%.

Michael Bishoff: Verified the details of use cases and reviewed any other written text. Estimated percent of work is about 10%

Edward LaFemina: As the requirements leader, was responsible for the formatting and appearance, as well as writing section 1 and documenting the functional and nonfunctional requirements. He reviewed and edited all other sections before submission. Estimated percent of work is about 70%.

Michael Lee: Michael worked on Appendix A & B, Open Issues, and References. Estimated percent of work is about 10%.

John Swank: Also contributed checks for thoroughness as well as completeness. Estimated percent of work is about 5%.