Single Sign On Project

Software Requirements Specification

1.3

12/2/2020

**Prepared for**

Team Single Sign On Authentication

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 25/12/2019 | Version 1 | Banumathy Sankaran | Initial Draft |
| 30/12/2019 | Version 1.1 | Banumathy Sankaran | Added section 5.Change management process |
| 5/02/2020 | Version 1.2 | Banumathy Sankaran | Added section 3.7 Design constraints |
| 15/02/2020 | Version 1.3 | Banumathy Sankaran | Typo;s Fixed |

**Document Approval**

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |
| --- | --- | --- |
| **Printed Name** | **Title** | **Date** |
| LQA | Lead Software Engg. | 18/02/2020 |
| TM | Lead QA Engineer | 18/02/2020 |
| PO | Project Sponsor | 20/02/2020 |

CONTENTS:

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

2. Specific Requirements

2.1 External Interface Requirements

2.1.1 User Interfaces

2.1.2 Hardware Interfaces

2.1.3 Software Interfaces

2.1.4 Communications Interfaces

3.1 Front End Details

3.2 Technical Requirements

[3.3 Possible technical validations in BDD format](https://docs.google.com/document/d/1rPW5DV82VJT6vtA1VDSrfxaCBuAduxW0zb1yfTh_VMk/edit?pli=1#heading=h.mjp0getbhfnb)

[3.4 Classes / Objects](https://docs.google.com/document/d/1rPW5DV82VJT6vtA1VDSrfxaCBuAduxW0zb1yfTh_VMk/edit?pli=1#heading=h.jhazfwhdo0tq)

[3.5 Non-Functional Requirements](https://docs.google.com/document/d/1rPW5DV82VJT6vtA1VDSrfxaCBuAduxW0zb1yfTh_VMk/edit?pli=1#heading=h.fyd165ozedci)

3.6 Inverse Requirements

3.7 Design Constraints

3.8 Logical Database Requirements

3.9 Other Requirements

4. Analysis Models

5. Change Management Process

1. **Introduction**

The Single Sign On Project aims at creating and testing the new Login Page from the end user’s perspective that verifies the basic authentication for the customers. This release leverages all the tests as part of the acceptance criteria which are in turn derived into Feature Files and mapped to the corresponding user stories.

**1.1 Purpose**

*The Purpose of this document is to outline the requirements for the Real Estate website to be developed by Team Albuquerque for Sportsbet.com.au. This document will be used by all stakeholders including developers and testers.*

**1.2 Scope**

*The scope of this project is limited to the testing of the features described in the succeeding sections of this document.*

*Non-functional testing like stress, security and performance is beyond scope of this project.*

*Automation testing and API Testing are in scope and need to be tested*

*The testing carried out across SIT, UAT and pre-production environments.*

*The login site will be only compatible with Chrome version 27 and above*

**1.3 References**

*Nil*

**2. Specific Requirements**

The Single Sign On will have 1 role-Customer

Following features/modules will be available to customer

|  |
| --- |
| **Customer** |
| Sign In To Real Estate Appln |
| New Customer SignUp |
| Sign In from other networking Sites |
| Strong password Protection |
| Option to view password |
| Session time out if the user is inactive for more than 10 minutes |
| Successful logout |

**Description of the modules**

|  |  |  |
| --- | --- | --- |
| **Feature Name** | **Applicable**  **Roles** | **Description** |
| Sign In To Real Estate Application | Customer | The customers should be able to login to the Real Estate application as the first and foremost step. |
| New User account sign up | Customer | The customer should be able to create an account successfully based on the authentication standards and policies |
| Strong password protection | Customer | A customer is constantly prompted for strong password and is forced to change password after every 45 days |
| View password option | Customer | This is to enable the users from the account getting locked for incorrect password attempts |
| Session Time out after 10 mins | Customer | The signed in user will automatically be signed out of the page in 10 minutes of inactivity |
| Successful log out | Customer | The end user logs out successfully |

**2.1 External Integration Requirements**

**2.1.1 User Integration**

None

**2.1.2 Hardware Integration**

None

**2.1.3 Software Integration**

Yes with third party networking sites such as facebook,Apple; requires integration testing

**3.1 Front End Details**

This section describes the Front end of Real Estate Single Sign On.

It also lists a few Test cases to describe the functioning of the system

Following is a list of feature wise fields

**Login and Logout**

* Email address
* Password
  + Show and Hide password buttons

**Create an Account**

* Email address field
* Password field

**Forgot Password**

* Alternate email address field
* Code recieved
* New password field
* Confirm password field

**Sign In using other networking sites**

* Google
* Facebook
* Apple

**Look And Feel of the new Login Page**

* Logo of the real estate app along with the external links
* Modal message
* Confirm delete

**3.2 Technical Requirements**

**New Account**

* Email address – Email id is required
* Email id cannot have Upper case letters
* Email id must be in the format aaa@xyz.com

**Password**

* Strong password of min 8 characters and max 24 characters
* One upper case, lower case numbers and special characters must be included
* Password cannot be repeated
* Must be changed every 45 days

**Login**

* User-ID must not be blank
* Password must not be blank

**Change Password**

* Old Password must not be blank
* New Password must not be blank
* Enter at-least one numeric value
* Enter at-least one special character
* Choose a difficult Password
* Confirm Password must not be blank
* Passwords do not Match

**3.3 Possible technical validations for Front End in BDD format**

**Background**: User is Logged In

**Given** the user is on the Login Page of the real Estate Application

**And** The emailid field and the password field is visible to the user

**Scenario: Verify that the login screen contains elements such as Username, Password**

**When** the user cofirms he is on the login page

**Then** the "Forgot your password" is visible to the user

**And** the "Sign in" button is visible to the user

**And** the "Create an account" link is available

**Scenario: Action 1 - successful login**

**When** the user enters a valid email address in the emailid field

**And** user enters a valid password in the password field

**And** user hits on the "Sign in" button

**Then** the user successfully lands on the "Home" page

**Scenario: Action 2 - verifying forgot password functionality**

**And** the user enters a valid email address in the emailid field

**And** the user clicks on "Forgot your password" link

**And** the user successfully lands on the "Forgot your password" page

**And** the user enters alternate email in the emailid field

**And** user hits on the "Reset my password" button

**And** the user successfully lands on the "Update your password" page

**And** Enters the verification code sent to his alternate mail

**When** enters the new password details on the "New password" field

**And** enters the new password details on the "Confirm new password" field

**And** user hits on the "Change Password" button

**Then** user sees a success message on the password change

**Scenario: Action 5 - verify if the user is able to login with a new password**

**When** the user enters a valid email address in the emailid field

**And** the user clicks on "Forgot your password" link

**And** successfully changes the password

**And** the user successfully lands on the "Sign in" page

**Then** the user is able to successfully login with the email id and new password

**Scenario: Verify if the user is getting alert message on entering invalid email address**

**When** user enters a valid password in the password field

**And** the user enters invalid email address in the emailid field

**Then** "Please enter a valid email address" message appears

**Scenario: Action 3 - verifying invalid login**

**Scenario Outline**: enter invalid credentials

**When** the user enters a <random valid email> in the emailid field

**And** the user enters a <random password> in the password field

**And** hits on the "Sign in" button

**Then** "Those details didn't work" message appears

**Examples**:

| random valid email | random password |

| abc@dddd.com | "SHDca\*0888" |

| testemail.fffh.com | "FludfbTT77@" |

**Scenario**: Checking if the user cannot create account with duplicate email addresses

**When** the user clicks on "Create an account" link

**And** the user successfully lands on the "Create account" page

**And** enters an already existing email address in the emailid field

**And** user enters a valid password in the password field

**And** user hits on the "Sign in" button

**Then** "You can't use this email address, because it has already been used to create an account" message appears

**Possible API validations**

* Verification of status code.   
    Code Status  
    200 OK  
    201 Created  
    202 Accepted  
    203 Non-Authoritative information  
    204 No Content  
    302 Found  
    400 Bad Request  
    401 Unauthorized  
    404 Not Found  
    500 Internal Server Error  
    502 Bad Gateway  
    503 Service Unavailable
* Response needs to test based on various inputs-details from the front end should be passed to the API of Single Sign in and the response should be either 200/202.
* Verify response time of API calls is within the timeout period of the gateway

**3.4 Design Constraints**

Many of the Real estate application users may not have adequate computer knowledge to use the site. Hence, System must be intuitive and easy to understand.

**3.5 Logical Database Requirements**

JDBC/ODBC, preferred Database such as SQL server, Oracle

**3.6 Other Requirements**

Nil