

Validation Report

Test Plan Identifier: Wishbone-Acceptance

IEEE 829-2008 standard test plan

Project Name: **Wishbone**

**Prepared for:** Daniel Dogbey, Wishbone

**Prepared by Axiom:**

**Team Leader:** Andrew Archibald

Zeyang Hu, Ksenia Lopukhina, Svetlana Netchaeva, Oksana Shapoval, Minyi Yang

**Algonquin College Professor:** Patricia Murphy

2019-04-10

# Revision History

|  |  |  |
| --- | --- | --- |
| **Description** | **Author** | **Date** |
| Discussion of report contents during meeting | All members | March 30th |
| Initial Edit/Review of content | All members | March 30th |
| Test cases for Login Page and User Register Page | Tom | March 31st |
| Test cases for My Profile Page | Min | April 1st |
| Test cases for My Messages Page | Ksenia | April 1st |
| Test cases for My Network Page | Andrew | April 1st |
| Test cases for My Page | Oksana | April 1st |
| Test cases for Search Page | Svetlana | April 1st |
| Testing Login Page and User Register Page | Min | April 2nd |
| Testing My Profile Page | Tom | April 2nd |
| Testing My Messages Page | Svetlana | April 2nd |
| Testing My Network Page | Oksana | April 2nd |
| Testing My Page | Andrew | April 2nd |
| Testing Search Page | Ksenia | April 2nd |
| Additional Information added to document directly or emailed to Andrew/Team email | All members | April 3rd |
| Merge information | All members | April 3rd |
| Complete Document – Review | Andrew, Svetlana | April 3rd |
| Update due to fixed bugs | Andrew | April 10th |

Table of Contents

[Revision History ii](#_Toc5232559)

[Introduction 1](#_Toc5232560)

[Strategy and Schedule 1](#_Toc5232561)

[Traceability Matrix 1](#_Toc5232562)

[Features To Be Tested 1](#_Toc5232563)

[Items Pass/Fail Criteria and Unresolved Bugs 7](#_Toc5232564)

[Responsibilities 1](#_Toc5232565)

[Approvals 13](#_Toc5232566)

[Appendices 14](#_Toc5232567)

[Appendix A – Schedule Gantt Chart 14](#_Toc5232568)

# Introduction

The purpose of this report is to provide records of entire site validation. It documents summary of all tests, expected results and actual outcomes. The report contains testing strategy, traceability matrix, tested features, list of unresolved bugs, and testing responsibilities of each member of Axiom.

# Strategy and Schedule

Testing of Wishbone project was performed at different levels of development process, namely unit testing, integration testing, system testing, and acceptance testing.

**Unit Testing**. The team tested each web page in isolation. Each team member was responsible for development of particular page, building database connection, and successful creating, retrieving, updating, and deleting data. Errors and issues that were detected in this phase saved time for us and allowed to fix them at earlier stage. This test phase started in 1st Sprint (February) and continued in 2nd Sprint (March).

**Integration Testing**. At this step, the groups of related pages were tested in order to detect interfacing issues or confirm that pages communicate correctly. Our main goal for integration testing was to provide connection and redirection between the following pares:

User Register Page – Login Page

My Network – My Messages

Search Page – My Page

My Profile – My Page

We were working on Integration Testing in 2nd Sprint (March) and 3rd Sprint (March).

**System Testing**. Once successful interactions between pages were confirmed, we started to test an application as a whole. These tests helped identify design issues such as different page styles, they also let us to confirm proper functionality and alignment to the client requirements. We spent the second part of the 3rd Sprint to test our application.

**Acceptance Testing**. During this testing, we determined that the application meets the client needs and is ready for production. The tests were conducted in 3rd Sprint (March – April).

If developers make some changes to fix some issues or add new features, they retest an application again to ensure that it works. This type of test is called regression testing.

Our team implemented regression testing during the whole development process. The application building required a lot of code modifications, adding new functionalities, and error fixing.

# Traceability Matrix

The Table I lists functional requirements and identifies associated test cases.

*Table I.* Functional Requirements and Associated test cases.

|  |  |  |
| --- | --- | --- |
| **Functional**  **Requirement** | **Functional Requirement Narrative** | **Test Cases** |
| SWF1 | Messaging function (provide artists and entertainers a method of internally communicating with each other) | T47 – T57 |
| SWF2 | “MyNetworking” function | T36, T38 – T44, T46 |
| SWF3 | Allow a user to view other users’ profiles | T45 |
| SWF4 | Allow user to create their personal profile | T25 – T35 |
| SWF5 | Allow user to invite another person to the platform | T37 |
| SWF6 | Allow user to connect to/find other artists | T1 – T14 |
| SWF7 | User Feed | T15 – T24 |
| SWF8 | Log in page. Allow user to log in using other social media credentials | T58 – T70 |

# Features to be Tested

The Table II illustrates the test cases that were created to test the features associated with functional requirements. The tests with invalid data highlighted in purple, and failed tests highlighted in red.

*Table II.* Test Cases.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case #** | **Description of Test Case** | **Pass Criteria** | **Tester** | **Pass/Fail** |
| **Search Page** | | | | |
| T1 | User enters **KseniaLopukhina** in search bar and clicks on Search button/or presses Enter. | Under **People were found**, Ksenia’s name appears with the link to her profile. | Ksenia  Lopukhina | PASS |
| T2 | User enters **Ksenia** in search bar and clicks on Search button. | Under **People were found**,  Ksenia’s name appears with the link to her profile. | Ksenia  Lopukhina | PASS |
| T3 | User enters **Ksenia1** in search bar and clicks on Search button. | The error message **Name cannot contain numbers** appears. | Ksenia  Lopukhina | PASS |
| T4 | User enters **Li** name in search bar and clicks on Search button. | Under **People were found**,the message **No results found** appears | Ksenia  Lopukhina | PASS |
| T5 | User enters **L** in search bar and clicks on Search button. | The error message **Name cannot contain one letter** appears. | Ksenia  Lopukhina | PASS |
| T6 | User selects **Ontario** in **Location** category and clicks on Search button. | Under **People were found**, the following names appear  **Andrew Archibald**  **Mike Smith**  **Oksana Shapoval**  **Svetlana Netchaeva**  **Zeyang Hu**  **Minyi Yang**  **Test test** | Ksenia  Lopukhina | PASS |
| T7 | User selects **Ontario** in **Location** category and **party** in **Interest** category and clicks on Search button. | Under **People were found**, the message **No results found** appears. | Ksenia  Lopukhina | PASS |
| T8 | User selects **Ontario** in **Location** category and **concert** in **Interest** category and clicks on Search button. | Under **People were found**, the following names appear  **Oksana** **Shapoval**  **Svetlana Netchaeva** | Ksenia  Lopukhina | PASS |
| T9 | User selects **Ontario** in **Location** category, **concert** in **Interest** category, **dancer** in **Artist type** category and clicks on Search button. | Under **People were found**, the following name appears  **Oksana** **Shapoval** | Ksenia  Lopukhina | PASS |
| T10 | User selects **musician** in **Artist type** category and clicks on Search button. | Under **People were found**, the following name appears  **Svetlana Netchaeva** | Ksenia  Lopukhina | PASS |
| T11 | User selects **event** in **Interest** category and clicks on Search button. | Under **People were found**,the following names appear  **Andrew Archibald**  **Zeyang Hu** | Ksenia  Lopukhina | PASS |
| T12 | User selects **festival** and **party** in **Interest** category and **Ontario** in **Location** category and clicks on Search button. | Under **People were found**, the following names appear  **Mike Smith**  **Minyi Yang**  **Test test** | Ksenia  Lopukhina | PASS |
| T13 | User does not enter any name in search bar, does not select any options in categories and clicks on Search button. | The error message **You have to choose at least one option** appears. | Ksenia  Lopukhina | PASS |
| T14 | User enters **Oksana** **Shapoval** in search bar and clicks on Search button. Under **People were found**,Oksana’s name appears. User clicks on View Page button. | The application redirects the user to Oksana’s page. | Ksenia  Lopukhina | PASS |
| **User Page** | | | | |
| T15 | User cannot create an empty post. | Error message: **Please fill out this field** | Andrew  Archibald | PASS |
| T16 | After login to user home page, user sees posts of people in descending order. | Posts are in descending order (time). | Andrew  Archibald | PASS |
| T17 | Post created by user has time of creation. | Posts are with time  (ex. **2019-04-02 21:15:17**) | Andrew  Archibald | PASS |
| T18 | Along with a time, post has user first name and last name. | Posts have user’s first name and last name. | Andrew  Archibald | PASS |
| T19 | User receives no result if he/she searches by not existing word or name among posts. | Empty post’s field. | Andrew  Archibald | PASS |
| T20 | User is able to search feed by creation time. | Shows feed for date | Andrew  Archibald | PASS |
| T21 | User is not limited in writing posts. He/she is able to write numbers or not existing words. | User writes a post with numbers or **hlsdjlsk**, and we can see it in the feed’s field. | Andrew  Archibald | PASS |
| T22 | User is able to search by number among feeds. | Filtering feeds by certain number. | Andrew  Archibald | PASS |
| T23 | User is able to find posts of particular user. | Search by name, see posts of the certain user. | Andrew  Archibald | PASS |
| T24 | User is able to find a feed with specific name in it. | Filter by key word, user sees all posts with this word. | Andrew  Archibald | PASS |
| **My Profile Page** | | | | |
| T25 | User does not input value. | Error message  **Please enter value** | Zeyang Hu | PASS |
| T26 | Input **a**,**123**, **#@**  in first name  or last name field. | Error message  **Please enter valid value** | Zeyang Hu | PASS |
| T27 | Input **minyi@hotmail.com**  or **abc@hotmail.com** | Correct email address  format. Data inserted  successfully. | Zeyang Hu | PASS |
| T28 | Input **@hotmail.com** | Error message  **Please enter valid value** | Zeyang Hu | PASS |
| T29 | Input **aabc123.com** | Error message  **Please enter valid value** | Zeyang Hu | PASS |
| T30 | Input **123**, **abc** in share url. | Error message  **Please enter valid value** | Zeyang Hu | PASS |
| T31 | Input **@**, **?**, **!** into work title  or description. | Error message  **Please enter valid value** | Zeyang Hu | PASS |
| T32 | Input **/**, **@**, **#** in experience  time. | Error message  **Please enter valid value** | Zeyang Hu | PASS |
| T33 | Input **abc** in phone number  field. | Error message  **Please enter valid value** | Zeyang Hu | PASS |
| T34 | Input **613-000** in  phone number field. | Error message  **Please enter valid value** | Zeyang Hu | PASS |
| T35 | Input 6130000000 in  phone number field. | Correct phone number  format. Data inserted  successfully. | Zeyang Hu | PASS |
| **My Network** | | | | |
| T36 | **My Network** page allows users to see their connections on the page. | **My Network** page shows connected users, suggested users, pending outgoing requests and pending incoming requests. | Oksana Shapoval | PASS |
| T37 | Inviting a user adds them to **My Network** as pending. | Adding suggested user adds user to **My Network** as pending user. | Oksana Shapoval | PASS |
| T38 | Cancelling pending request. | If a user clicks cancel on their pending user request, the user is removed from **My Network** | Oksana Shapoval | PASS |
| T39 | If I send a request, other user can see and accept the request. | Sending a request updates other user, allows them to accept request. | Oksana Shapoval | PASS |
| T40 | Accepting an incoming request adds user to **My Network** | User that accepted request gets the user they accepted in their **My Network** page. | Oksana Shapoval | PASS |
| T41 | If other users accept your request to add, that user appears on sender’s **My Network** | User that accepted request shows up in sender’s **My Network** page. | Oksana Shapoval | PASS |
| T42 | Clicking **My Network** loads logged in users **My Network** page. | **My Network** on navigation bar loads users **My Network** page. | Oksana Shapoval | PASS |
| T43 | Deny an incoming request. | Allow user receiving a request to deny the request. | Oksana Shapoval | PASS |
| T44 | Displays correctly on multiple browser. | Displays correctly on different browsers. | Oksana Shapoval | PASS |
| T45 | Redirect to user profile. | Clicking a user’s profile button loads their profile. | Oksana Shapoval | PASS |
| T46 | Redirect to Messages. | Clicking Message user loads the Messenger to send messages to the user. | Oksana Shapoval | PASS |
| **Chat** | | | | |
| T47 | Pre-condition:  User1 is connected to User2. User1 never wrote messages to User2.  User1 clicks on **Message** button under User2 profile on **My Network** page. | Chat window is opened. No messages are displayed. In the text view **To** full name of the User2 is displayed. | Svetlana  Netchaeva | PASS |
| T48 | Pre-condition:  User1 is connected to User2. User1 has sent messages to User2 before.  User1 clicks on **Message** button under User2 profile on **My Network** page. | Chat window is opened. All the messages between User1 and User2 displayed with the option to scroll through them. In the text view **To** full name of the User2 is displayed. | Svetlana  Netchaeva | PASS |
| T49 | User1 types some text in message window and presses **Send Message**. | Message is sent. Message is displayed in green color on the right side of the messages text area. | Svetlana  Netchaeva | PASS |
| T50 | User1 types some text in message window and presses Enter. | Message is sent. Message is displayed in green color on the right side of the messages text area. | Svetlana  Netchaeva | PASS |
| T51 | User1 does not type any text in message window and presses Enter. | No message is sent. No message is displayed in green color on the right side of the messages text area. | Svetlana  Netchaeva | PASS |
| T52 | User1 does not type any text in message window and presses **Send Message** button. | No message is sent. No message is displayed in green color on the right side of the messages text area. | Svetlana  Netchaeva | PASS |
| T53 | User1 does not type numeric value and presses **Send** **Message** button or Enter. | Message is sent. Message is displayed in green color on the right side of the messages text area. | Svetlana  Netchaeva | PASS |
| T54 | Pre-condition:  User2 is connected to User1. User2 has sent messages to User1 before.  User2 clicks on **Message** button under User1 profile on **My Network** page. | Chat window is opened. All the messages between User1 and User2 displayed with the option to scroll through them. Messages from User1 are displayed on the right in gray, messages from User2 are displayed on the left in green. In the text view **To** full name of the User1 is displayed. | Svetlana  Netchaeva | PASS |
| T55 | Not authorized user types chat url. | 404 page is displayed. | Svetlana  Netchaeva | PASS |
| T56 | Authorized user types chat url with the receiver ID that does not exist. | 404 page is displayed. | Svetlana  Netchaeva | PASS |
| T57 | User1 sends message to User2. | Message area of the User2 displays the message from User1 in 3 seconds. | Svetlana  Netchaeva | PASS |
| **User Register Page** | | | | |
| T58 | Leave first name field empty when registering. | New user cannot be registered. Alert appears that first name field cannot be empty. | Minyi  Yang | PASS |
| T59 | Leave last name field empty when registering. | New user cannot be registered. Alert appears that last name fields cannot be empty. | Minyi  Yang | PASS |
| T60 | Leave the user email field empty when registering. | New user cannot be registered. Alert appears that email field cannot be empty. | Minyi  Yang | PASS |
| T61 | Leave password and field empty when registering. | New user cannot be registered. Alert appears and tell user that password field cannot be empty. | Minyi  Yang | PASS |
| T62 | Leave confirming-password field empty when registering. | New user cannot be registered. Alert appears and tell user that confirming-password field cannot be empty. | Minyi  Yang | PASS |
| T63 | Type in a wrong format which does not contain special character “**@**” or “**.**” in email field. | New user cannot be registered. Alert appears and tell user that email format did not pass validation | Minyi  Yang | PASS |
| T64 | When registering, user does not type in the same string characters in password field and confirming-password. | New user cannot be registered. Alert appears and tell user that password and confirming password are not matched. | Minyi  Yang | PASS |
| T65 | When registering, the password user typed in is too short or too long that is not within range between 8 to 12 characters. Or, user did not type in at least one capital letter. | New user cannot be registered. Alert appears and tell user that password is not within a valid length. | Minyi  Yang | PASS |
| T66 | When registering, user typed in first name, last name, email (with valid format), password and same password character in confirm password field. | New user will be registered and a notice is shown that the user has been registered successfully. | Minyi  Yang | PASS |
| **Login Page** | | | | |
| T67 | When logging in, user does not provide email address in the field. | Alert will show that user will not be able to login, and email is required for logging in. | Minyi  Yang | PASS |
| T68 | When logging in, user does not provide password in the field. | Alert will show that user will not be able to login, and password is required for logging in. | Minyi  Yang | PASS |
| T69 | When logging in, user does not provide correct password. | User will not be logged in because of the wrong password. | Minyi  Yang | PASS |
| T70 | When logging in, user provide valid user email address and valid password. | User will be logged in and redirected to **My Page**. | Minyi  Yang | PASS |

# Items Pass/Fail Criteria and Unresolved Bugs

~~Test 18 and 20 failed. 18 is difficult because PHP post doesn’t seem to update session data, so session data saved in the previous form isn’t accessible, a workaround will need to be found. 20 Searching by date doesn’t give any results, may be intentional, search by time doesn’t make sense, but day may be useful.~~

~~Tests T26, T30, T31, T32 have a problem. The column in database were set as String. so the system will accept special characters. Therefore, we need to filter data to prevent special characters to be inserted.~~

~~Test T44 failed. The pages are not shown correctly on multiple browser.~~

Issues resolved.

# Responsibilities

Every team member participated in testing and was assigned to test a page different his/her own. To check team members’ test responsibilities, see the Table II.

# Approvals

|  | **Signature** | **Date**  (hand-written) |
| --- | --- | --- |
| **Client name:** <insert client name>  Wishbone |  |  |
| **Project Leader:** Andrew Archibald  Project group: Axiom |  |  |
| **Team member:** Zeyang Hu  Project group: Axiom |  |  |
| **Team member:** Ksenia Lopukhina  Project group: Axiom |  |  |
| **Team member:** Svetlana Netchaeva  Project group: Axiom |  |  |
| **Team member**: Oksana Shapoval  Project group: Axiom |  |  |
| **Team member:** Minyi Yang  Project group: Axiom |  |  |

# Appendices

## Appendix A – Schedule Gantt Chart

The Appendix A was provided as an example by Patricia Murphy (January 2019) Software Development Project. It was used here with no modifications.

