Internet Music Database (IMuDb) - Integration Test Plan  
  
Dimitar Dimitrov  
Patrick Freestone  
Greg Vannoni

*Table of Contents*

I. Introduction 3

I.I Purpose 3

I.II Scope 3

II. Integration Strategy 4

II.I Entry Criteria 4

II.II Elements to Be Integrated 4

II.III Integration Approach 4

II.IV Sequence of Integration 4

III. Test Cases 5

IV. Integration Test Environment 8

V. Program Stubs and Test Data 9

V.I User Account 9

V.II Search Data 9

V.III Front Page Dynamic Content 9

V.IV Band Page 9

VI. Schedule 10

VII. References 11

# I. Introduction

## 

## I.I Purpose

The purpose of this document is to serve as a guide for testing the components of the Internet Music Database (IMuDb) defined in the software design specification (SDS) as they are integrated.

## I.II Scope

This document covers the high-level integration test plan for IMuDb.

# II. Integration Strategy

## II.I Entry Criteria

As with any software project, integration and test should not commence before detailed design and 75% of code and unit testing is complete.  This ensures that a testable product has already been through some validation and verification before integration testing occurs.

## II.II Elements to Be Integrated

* Front Page Dynamic Content - validate that the front page dynamic content allows users to login, create an account, and are able to navigate to album pages, and band/musician pages
* Purchase Song - ensure that a user can find a song (via search or click-through) and is able to purchase the song via iTunes or Amazon.
* User Profile Creation -  A user should be able to register for an account, receive a verification email, and then log into the system with in a short amount of time.

## II.III Integration Approach

Integration will occur from a top-down approach first establishing the User Interface integration with Shared Services, then integration with User, Domain and Functional modules and then Database.  During this testing it will be important to create the software so that it can handle "canned" messages, or messages that do not require a database.  A database must be primitively setup for user accounts, but only a bare-bones configuration is recommended.

A top-down approach is favored as most of IMuDb has relatively trivial database access, but heavy User Interface needs.  Integration testing in this way is the best for find defects early on so that system users will not find them when the system is in production.

## II.IV Sequence of Integration

Provide a sequence of actions that will be performed as part of the integration(s).

1. Stand up User Interface for the Front Page Dynamic Content
2. Stand up Shared Services (this is the foundation for database access, but will returned canned information)
3. Perform integration testing between the User Interface and Shared Services (ie: click through the Front Page to a second page)
4. Stand up Domain and User modules
5. Perform integration testing again from the front page to another, this time more functionality should be present and more information should be displayed to the user
6. Perform an integration test for user account creation, then log into the system as that user
7. Stand up functional modules and perform an integration test after searching for a keyword, then click through the results page to a band/musician or album page
8. Finally, stand up the database portion of the system and perform end-to-end integration tests with a multitude of different data

# III. Test Cases

For each step defined in II.IV, provide the actual test cases that will be run to test the integration.

|  |  |
| --- | --- |
| Test Name | TC1 - Front Page Dynamic Content link to Band/Musician Displays Information |
| Description | This will test Section II.IV, Step 3 |
| Steps | Select a banner or link from the front page to be directed to a band/musician page |
| Expected Result | User is presented with a canned, barebones page for the band Perl Jam |
| Actual Results |  |

|  |  |
| --- | --- |
| Test Name | TC2 - Front Page Dynamic Content link to Band/Musician Displays Information with Domain and User modules |
| Description | This will test Section II.IV, Step 5 |
| Steps | Select a banner or link from the front page to be directed to a band/musician page |
| Expected Result | User is presented with a canned page with more information displayed than TC1 for the band Perl Jam, user should also see area for comments |
| Actual Results |  |

|  |  |
| --- | --- |
| Test Name | TC3 - User account Creation and Usage |
| Description | This will test Section II.IV, Step 6 |
| Steps | User should attempt to create an account, this should activate a canned user account and send an activation email.  The user should attempt to login. |
| Expected Result | An email is sent for the canned registered user.  The canned user account should be able to log in. |
| Actual Results |  |

|  |  |
| --- | --- |
| Test Name | TC4 - Search |
| Description | This will test Section II.IV, Step 7 |
| Steps | User should perform a basic search from the front page with the keyword "Jeremy" |
| Expected Result | Canned search results should be displayed per Section V's canned test data |
| Actual Results |  |

|  |  |
| --- | --- |
| Test Name | TC5 - Database Integration |
| Description | This will test Section II.IV, Step 8 |
| Steps | All previous tests should be performed, but because the database has been integrated, the user should not be presented with canned data from the previous steps. |
| Expected Result | User is not presented with canned data and can search, and navigate through the website (to band/musician and album pages) and will be provided different data |
| Actual Results |  |

|  |  |
| --- | --- |
| Test Name | TC6 - Contributor |
| Description | This will allow data to be added to IMuDb |
| Steps | Now that database integration is possible, allow a test user account to become a contributor.  The contributor account should be used to create an additional album page for an artist.  Once the page is created, a search should be performed with the same album name. |
| Expected Result | Contributor account is allowed to create an album page.  After the page is created, a user is able to see the album in search results. |
| Actual Results |  |

# IV. Integration Test Environment

Describe where the integration tests will occur.

Integration testing can occur from anywhere with a web-based project.  The team performing integration testing should be outside of the work network, on a relatively normal machine (2GHz Intel Core Duo CPU, 2GB RAM, Windows XP with both Internet Explorer 8 and Google Chrome).

# V. Program Stubs and Test Data

Define any program stubs or test data that will need to be created to support the tests.

## V.I User Account

Test User Account and Password = testacct : 123test

## V.II Search Data

Search query string = "Jeremy"

Search results should provide a Band/Musician = "Perl Jam" with Album = "Ten"

## V.III Front Page Dynamic Content

Front page should have a stub image for Perl Jam, when the user clicks on the banner they should

be taken to the band page

## V.IV Band Page

When a user goes to any band page, they should be always directed to the band Perl Jam.

Displayed on the page should be the following data:

Musicians:

Eddie Vedder  
Mike McCready  
Stone Gossard  
Jeff Ament  
Matt Cameron

Genres:

Alternative Rock, Grunge

Canned data for Historical Information, Albums and Awards should also be displayed

# VI. Schedule

Integration testing will occur when the team is 75% complete with Code and Unit Testing (CUT) or after one month of development, whichever comes first.

# VII. References

Software Design Specification. Team Dragon. May 9, 2010.