System Requirements Specification

MusicBug

Client
Christopher Nguyen

Team 4
Tommy Brickwedde
Jeff Clouse
Michael Cohen
Joe DiMarino
Sean Ehrig
Kevin Wiggins

3/1/2012

MusicBug
System Requirements Specification

Table of Contents

1. Introduction

Welcome to the MusicBug system requirements specification. MusicBug is a social music web application, which can be thought of as a mixture of iTunes and Facebook.

1.1 Purpose of This Document

This document is designed to explain the features of the MusicBug web application, it's functions, and the conditions required for operation. The intended audience is the MusicBug development team as well as the faculty customer, Christopher Nguyen.

1.2 Purpose of the Product

MusicBug was designed to provide personal information analysis for marketing and advertising. Music sells, and eventual integration of a radio-style application with which to play music will gather a large audience. Our client will be able to collect information on users, analyze the information, and sell targeted advertising to generate revenue. In exchange for information gathering users will be able to use the service free of charge.

1.3 Product Scope

The MusicBug web application consists of eight use cases including: searching the music database, viewing detailed music meta-data, favoriting, rating, and reviewing music, logging in, registering, and viewing profiles. Please refer to figures 1-9 below for further understanding of actions available to users. Additional Facebook integration will be added in later spirals.

Figure 1. High level use case context diagram overview

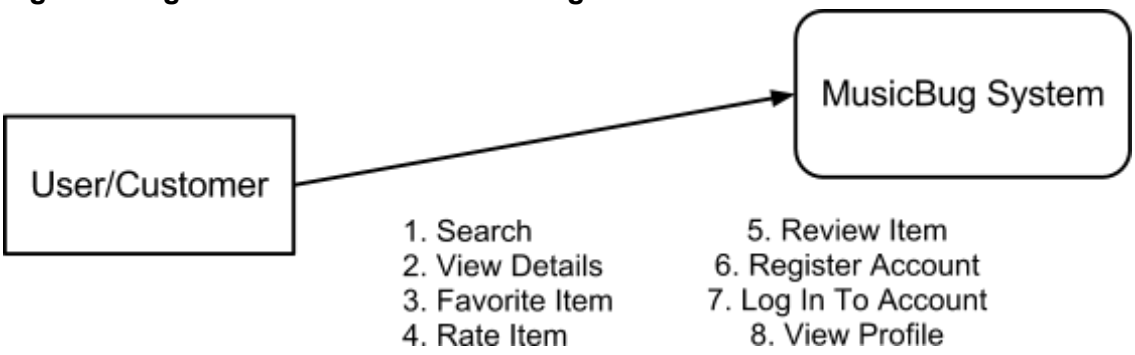


Figure 2: Use case 1, Search the database

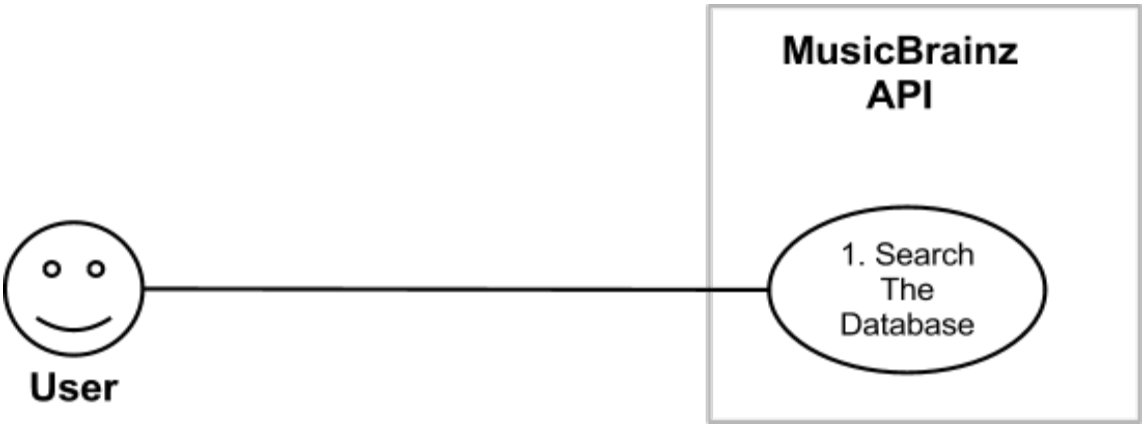


Figure 3: Use case 2, View detailed artist, album and track information

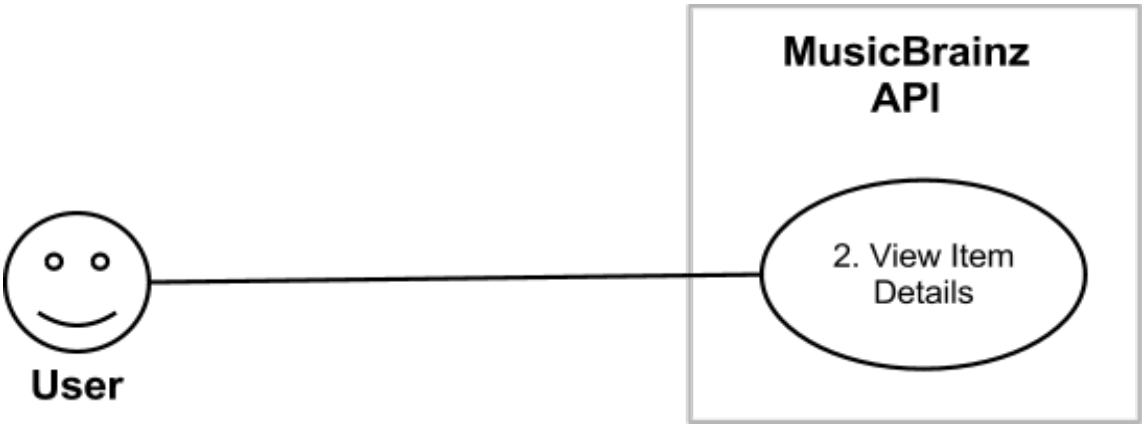


Figure 4: Use case 3, Favorite an item

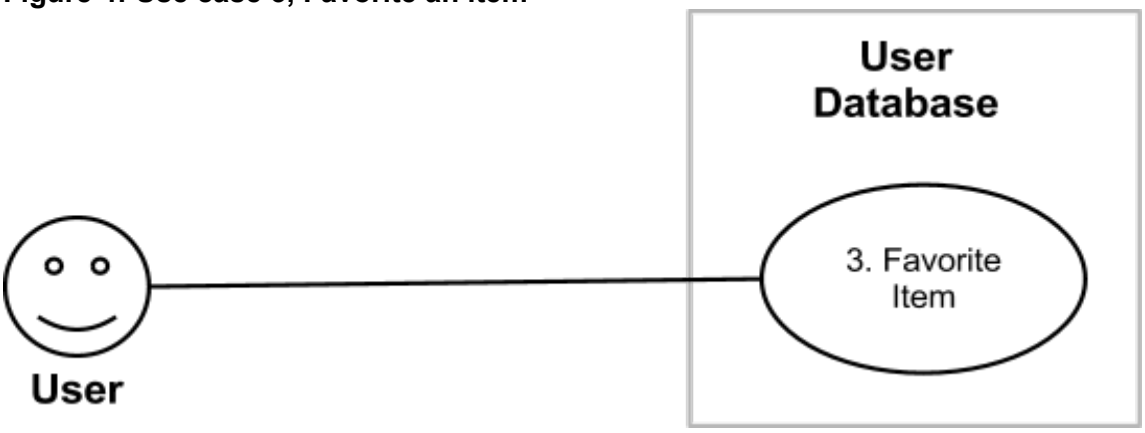


Figure 5: Use case 4, Rate an Item

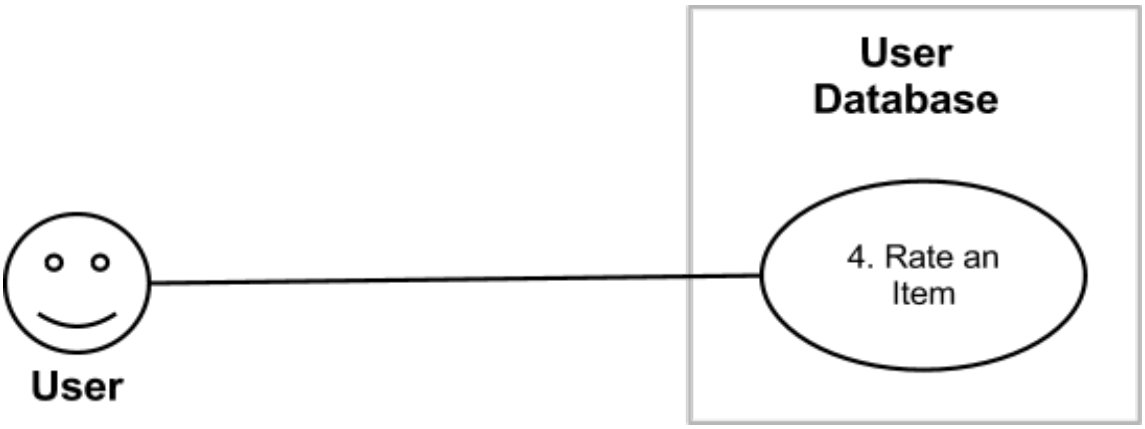


Figure 6: Use case 5, Review an Item

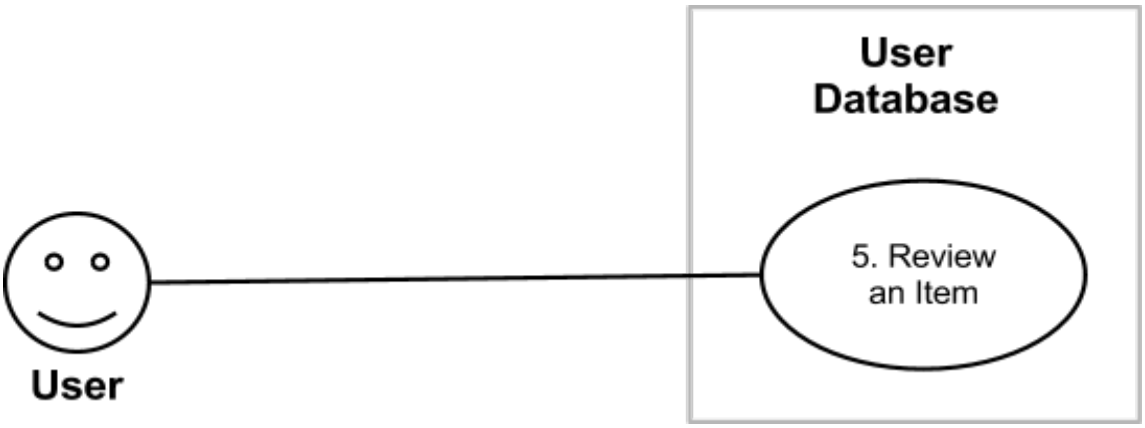


Figure 7: Use case 6, Register with the system

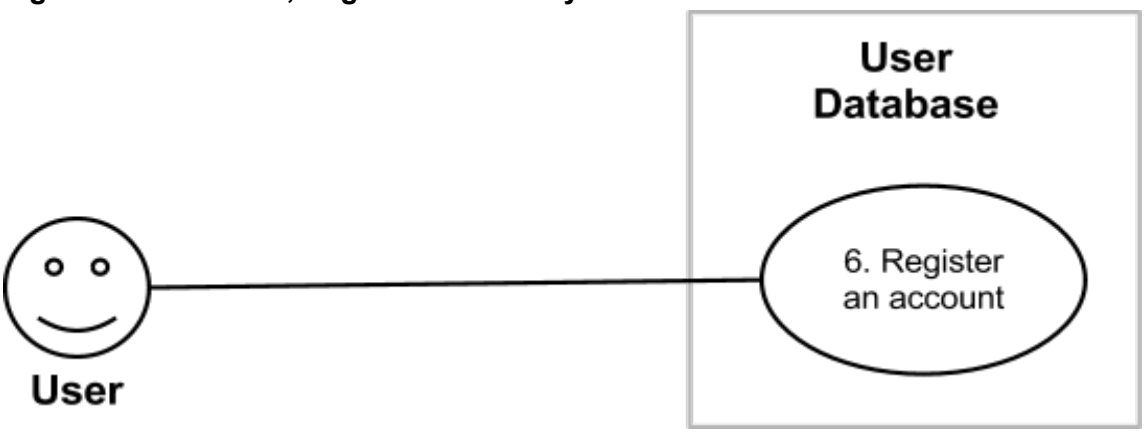


Figure 8: Use case 7, Log in to the Account

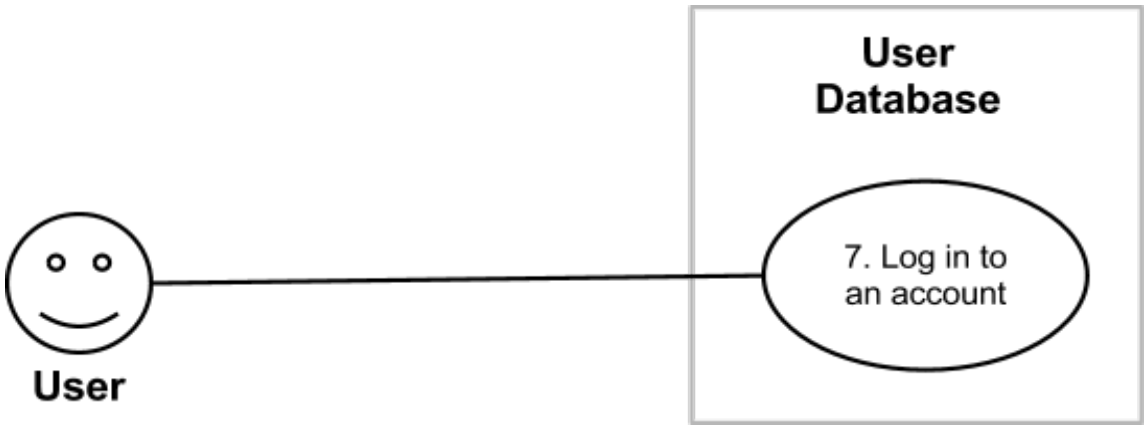
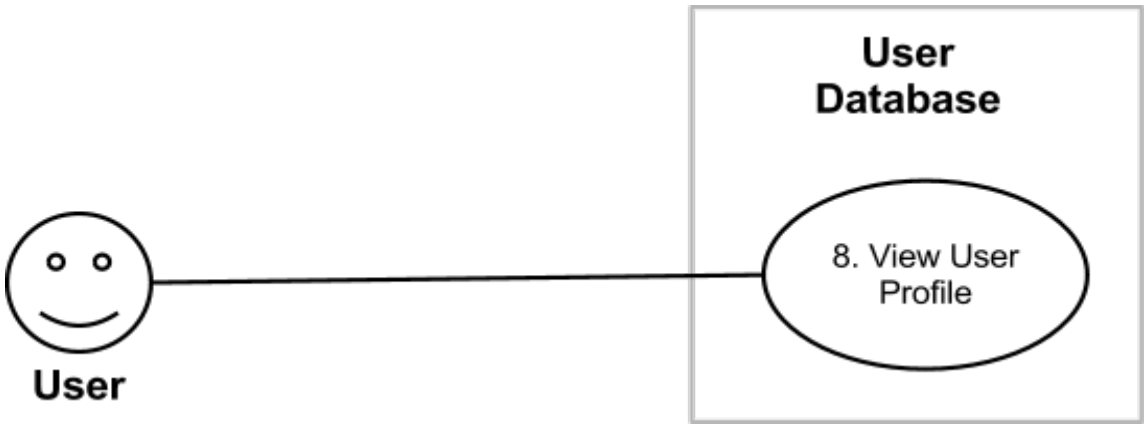


Figure 9: Use case 8, Viewing User Profile



2. Functional Requirements

Please refer to the top-level use case diagram, figure 1, and context diagram, figure 1, above for an overview of each use case.

2.1 Use Case 1

Number	1	
Name	Search the Database	
Summary	User enters criteria to search the database	
Priority	5	
Preconditions	The criteria the user enters is in the database	
Postconditions	A list of relevant items in the database are displayed in alphabetical order by search type.	

Primary Actor	User	
Secondary Actors	MusicBrainz API	
Trigger	User clicks the search button	
Main Scenario	**Step**	**Action**
	1	User enters criteria to search for in the database (artist, track, album)
	2	System checks the database, and displays a list of relevant items categorized by search type (artist, track, album).
Extensions	**Step**	**Branching Action**
	1a	Criteria not in the database **:** Display a message
Open Issues	What meta-data will be searched? Will all meta-data be searched or only artist, track, and album?	

2.2 Use Case 2

Number	2	
Name	View details of an item	
Summary	User views details of specific data	
Priority	5	
Preconditions	A list of items in the database is displayed in browse or search result format.	
Postconditions	The details of the item specified by the user are displayed	
Primary Actor	User	
Secondary Actors	MusicBrainz API	
Trigger	User clicks an item to view	
Main Scenario	**Step**	**Action**
	1	Detailed information about that specific data is presented
Extensions	**Step**	**Branching Action**
	1a	User clicks to view detailed information on data contained within these details. User is directed to use case #2.
Open Issues	What specific meta-data is displayed?	

2.3 Use Case 3

Number	3	
Name	Favorite an item	
Summary	User adds an item to their list of favorite items	
Priority	3	
Preconditions	User is looking at a specific item	
Postconditions	The item is in the user's favorite list	
Primary Actor	User	
Secondary Actors	User Database	
Trigger	User clicks the favorite button	
Main Scenario	**Step**	**Action**
	1	The item is added to the user's list of favorites

2.4 Use Case 4

Number	4	
Name	Rate an item	
Summary	The user rates an item on a numerical scale	
Priority	3	
Preconditions	User is viewing the details of an item	
Postconditions	The item has a rating that is tied to the user	
Primary Actor	User	
Secondary Actors	User Database	
Trigger	Clicking the rate button on either the search or details page	
Main Scenario	**Step**	**Action**
	1	User chooses a rating to give to the item
	2	The rating is added to the User Database

2.5 Use Case 5

Number	5	
Name	Review an item	

Summary	The user writes a review of an item	
Priority	3	
Preconditions	User is viewing the details of an item	
Postconditions	The user's review is shown on the item page	
Primary Actor	User	
Secondary Actors	User Database	
Trigger	Clicking the review button on either the search or details page	
Main Scenario	**Step**	**Action**
	1	User types a review of the item
	2	The review is added to the User Database

2.6 Use Case 6

Number	6	
Name	Register an account	
Summary	The User creates an account in the database	
Priority	5	
Preconditions	The account details are not already in the database	
Postconditions	The account is added to the database	
Primary Actor	User	
Secondary Actors	User Database	
Trigger	The user clicks the register button	
Main Scenario	**Step**	**Action**
	1	The user enters the information required to make a new account
	2	The system creates a new account with the information given by the user

2.7 Use Case 7

Number	7	
Name	Log in to an account	
Summary	The user logs in to the site	
Priority	5	

Preconditions	The user already has an account in the system	
Postconditions	The user is logged in to the site	
Primary Actor	User	
Secondary Actors	User Database	
Trigger	The user clicks the log in button	
Main Scenario	**Step**	**Action**
	1	The user enters their account info into the login form
	2	The user is logged in to the system

2.8 Use Case 8

Number	8	
Name	View User Profile	
Summary	The user is taken to a page displaying their profile information	
Priority	4	
Preconditions	The user is logged in to their account	
Postconditions	The user is at their profile page	
Primary Actor	User	
Secondary Actors	User Database	
Trigger	The user clicks the profile button	
Main Scenario	**Step**	**Action**
	1	The system displays the user's information

3. Use Case Tests

Use case testing is a vital part of product validation and verification. The final product should both work how originally intended and be bug free. These tests outline how to test to the requirements during both system and release testing.

3.1 Use Case 1 Test - Searching the Database

Search the music database for artist, album or track. Confirm that the search result displays the

same information that was searched, and that no data is being hidden. See Testing Report for further details and results.

3.2 Use Case 2 Test - Viewing Detailed Music Meta-data

Search or browse for a specific artist, track or album. Switch between viewing artist, then album with track information. See Testing Report for further details and results.

3.3 Use Case 3 Test - Favorite An Item

Add an item to the user's favorites. Check the favorites page to confirm that the item was added. See Testing Report for further details and results.

3.4 Use Case 4 Test - Rating An Item

Give an item a rating. Check that the item has a rating on the details page. See Testing Report for further details and results.

3.5 Use Case 5 Test - Reviewing An Item

Write a review for an item. Confirm that a review is added to the details page of the item. See Testing Report for further details and results.

3.6 Use Case 6 Test - Registering to the Database

Register a new account for the database. Verify that the information the user entered at registration is correct. See Testing Report for further details and results.

3.7 Use Case 7 Test - Logging in to the Database

Log in to the database using an already created account. Verify that the user has access to features only usable by members. See Testing Report for further details and results.

3.8 Use Case 8 Test - Viewing User Profile

Log in to an account with appropriate information. Navigate to the profile page, and check that the user's favorites, ratings, etc. are correctly displayed. See Testing Report for further details and results.

4. Non-Functional Requirements

Some things are a necessity when creating a web application. Below outlines the criteria used to judge the system as a whole.

#	Item	Priority 1(lowest) to 5 (highest)

1	The database shall be secure	5
2	Reliable environment	3
3	Application should be easily navigated	4
4	The system shall be written in Python	2
5	The system shall use the Django framework	2
6	The system shall use Model view controller architecture	2
7	Application is highly responsive	3
8	Application should be extensible	3
9	Application is easy to maintain	3
10	All code should be well documented	2
11	Application should be portable	1

5. User Interface

See “User Interface Design Document for Social Music Network”

6. Deliverables

Deliverables include:

- Systems Requirement Specification
- System Design Document
- User Interface Design Document
- Copies of all Biweekly Status Reports

A CD (or electronic copy in a ZIP file) containing the following:

- Systems Requirement Specification
- System Design Document
- User Interface Design Document
- User Manual
- All source code
- The executable program
- Any other software required for installation and execution of the delivered program.

7. Open Issues

Issues that have been raised and do not yet have a conclusion will be addresses later in the development process. Please refer to the table below.

Issues	**Scheduled for**
Data security	Spiral 3
Create rating algorithm to determine artist rating	Spiral 3
Show album artwork in the details pages	Spiral 3
Non-functional requirements difficult to verify	Ongoing

8. Appendix A – Agreement Between Customer and Contractor

The customer agrees to a *Music Social Network* system with searching, browsing and detailed meta-data capabilities. Use cases are included in the functional requirements section above of the behavior between the system and user. Additional features will be provided in further development spirals.

When and if future changes to this document occur a drafted new document will be created. Both a hard and electric copy of both versions will be presented to the client for review. Upon approval, the draft will be finalized and signed off by both parties.

Client

Name _____ Date _____
Print

Name _____ Date _____
Signature

Team

Name _____ Date _____
Print

Name _____ Date _____
Signature

Name _____ Date _____
Print

Name _____ Date _____
Signature

Name _____ Date _____
Print

Name _____ Date _____
Signature

Name _____ Date _____
Print

Name _____ Date _____
Signature

Name _____ Date _____
Print

Name _____ Date _____
Signature

Name _____ Date _____
Print

Name _____ Date _____
Signature

9. Appendix B – Team Review Sign-off

This document has been collaboratively written by all members the team. Additionally, all team members have reviewed this document and agree on both the content and the format. Any disagreements or concerns are addressed in team comments below.

Team

Name _____ Date _____
Print

Name _____ Date _____
Signature

Comments
_____
_____

Name _____ Date _____
Print

Name _____ Date _____
Signature

Comments
_____
_____

Name _____ Date _____
Print

Name _____ Date _____
Signature

Comments
_____
_____

Name _____ Date

_____
Print
Name _____ Date
_____
Signature
Comments
_____
_____
__

Name _____ Date
_____
Print
Name _____ Date
_____
Signature
Comments
_____
_____
__

Name _____ Date
_____
Print
Name _____ Date
_____
Signature
Comments
_____
_____
__

10. Appendix C – Document Contributions

Jeff Clouse took the lead on the writing of this document with 70% of the writing. He began the writing of the use cases and other functional requirements, the non-functional requirements, and document introduction. Michael Cohen contributed by creating the diagrams, appendices, adding to non-functional requirements, document formatting as well as creating the UML. Michael was a key player in review of content. Michael contributed to 30% of this document.