**Project Part - 1**

**CSI 5380 Systems and Architectures for Electronic Commerce**

**Seedy Music Club**

|  |
| --- |
| **Professor** |
| **Prof. Hussein Al Osman** |

|  |  |
| --- | --- |
| **Group Members** | **Student Number** |
| Karim Sultan | 603598 |
| Daniel Hong | 6876936 |
| Ahmad Traboulsi | 7989359 |
| Saad Rabiei | 8934521 |
| Abisola Sanni | 8968593 |

Table of Contents

[Introduction 4](#_Toc497833094)

[Project features 4](#_Toc497833095)

[Project Software and Versions 5](#_Toc497833096)

[Project Structure 7](#_Toc497833097)

[Project Architecture 10](#_Toc497833098)

[DATABASE MODEL 11](#_Toc497833099)

[Use Case Diagram 14](#_Toc497833100)

[Testing 16](#_Toc497833101)

# Introduction

Seedy Music Club is a client-server web application that provides the user a wide range of CDs to buy from. The application using the Model-View-Controller and allows the user to browse through selections of CDs according to genres, listen to samples of the CDs, easily purchase CDs, manage accounts and personalize the application.

# Project features

A Github repository was used for version management and team collaboration throughout the project. The project was secured during the development process to prevent unauthorized users from making changes to the system. All components of the application are set up to run over SSL through HTTPS to protect the integrity of the data being sent over the network. The project uses the REST web service to call the database via hibernate. The project has three major web services; account, catalog, and order. Account web service has sub-controllers; create account, detail account and login account working with the order web service to manage users’ information in the database. The catalog web service works with the view model controller, session controller, and servlet to provide users with CD information via the browse.jsp. The order web service keeps track of users’ orders and payment information and relates with account web service to prevent unauthorized transactions.

Seedy Music Club order web service

1. Payment processing.
2. Order processing.
3. Ensures users are logged in before they can attempt to purchase.

Seedy Music Club Catalog web service

1. Display all CDs with information.
2. Display CDs according to search keywords.
3. Display CDs according to genre selected.
4. Provide option to add to cart.
5. Cart allows users to delete and add multiple CDs.

Seedy Music Club account Catalog

1. Manage users’ registration details.
2. Monitors registered user login information.
3. Manages session.

# 

# Project Software and Versions

The purpose of this document is to list the most recent, stable generally available (GA) software versions to be used in our project.

Components List

The OS is irrelevant as all packages are cross-platform. For example, the production environment will use Fedora v26 Linux while my personal dev platform will be on Windows 10.

NOTE: Only stable versions will be considered (no alpha/beta/pre-release). In this case, I suggest the latest stable builds as of September 15, 2017, and that we lock on those, even should a newer version be introduced during the project duration.

All software packages are using the 64-bit version.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Layer** | **Component** | **Version** | **Notes** | **Link** |
| Web Tier, Business Tier | Apache Tomcat | 8.5.20 | Web Server and  Application Server; supports JSP and Servlets | https://tomcat.apa  che.org/whichversi  on.html |
| Data Tier | MySQL Community Server | 5.7.19 | Latest GA version; supports Fedora 26 and Win X | https://dev.mysql.  com/downloads/my  sql/ |
| Data Access | Access JDBC: MySQL Connector/J Driver | 5.1.44 | Developed by MySQL devs; needs to be installed into  TomCat | https://dev.mysql.  com/downloads/co  nnector/j/ |
| Integrated Development  Environment (IDE) | Eclipse Oxygen | Auto-updates to  latest version | If you have it  already, then  update. If not,  download Eclipse  Oxygen for latest | https://www.eclips  e.org/downloads/ |
| OS for Deployment  Environment | Fedora Server | 26 | 26 Fedora is a variant of Red Hat Linux and extremely stable | https://getfedora.o  rg/en/server/downl  oad/ |
| OS for Development  Environment | Whatever you are  Using | \* | Personal choice | N/A |
| Java | JVM/JRE: OpenJDK | 1.8.0 Update 144  (64-bit) | The most recent  the free version of Java JDK | JDK  http://openjdk.java  .net/install/index.h  tml (Linux)  https://java.com/e  n/download/  (Oracle binaries for  Windows, Mac) |
| JSP | Apache Commons Taglib | 1.2.5 | Just place in  project’s WEB-INF  lib/ directory.  There are 4 .jar  files. | http://tomcat.apac  he.org/download-t  aglibs.cgi#Standar  d-1.2.5 |
| Data | Hibernate | 4.3.11 NOTE: Will  upgrade to  5.12.1 for Part 3 | Place in  WEB-INF/lib/  directory. Include  C3P0. | https://sourceforge  .net/projects/hiber  nate/files/hibernate  4/4.3.11.Final/ |

Other Tools

Here is a partial list of other tools that aren’t necessary but may help:

● Putty (www.putty.org) for SSH (there’s also a newer variant, Kitty, that is good)

● Jetty (embedded application server in Eclipse for testing/debugging servlets in the IDE)

● MySQL Workbench ( dev.mysql.com ) - visualization and administration tools for MySQL

● Notepad (LOL, just had to mention this basic one; can’t live without it!)

● Github for code sharing; Eclipse has support for this; I will start a repository once we reach the next part

● MS Visio (free from dreamspark ); for UML and logic / flow visuals

● MS Project (free from dreamspark ); project plans and team tasks

# 

# Project Structure

Seedy Music Club

1. Deployment Descriptor- contains context parameters, error pages, filter mappings, filters, listeners, references, servlet mappings, servlets and welcome pages
2. JAX-WS Web Services
3. Java resources

* Src
* Club.seedymusic.controller- a collection of the controllers calling the web services

1. AccountCreateControllerServlet- create an account for a new user in the database
2. AccountDetailControllerServlet- compares login information with registration information of the user using ID
3. AccountLoginControllerServlet- Verifies login information
4. CatalogServlet.java- gets CDs information from the web service according to the request of the user
5. CatalogViewModel.java- sends CD information to the view according to the user request through the session controller.
6. OrderController.java- manages the shopping cart and payment information
7. PaginationItem.java-
8. SessionController.java

* Club.seedymusic.ecom

1. shoppingCart.java- manages the shopping cart by allowing users to add and delete CDs and calculate orders and payment

* club.seedymusic.exceptions- throws exceptions when certain criteria are not met and informs the user

1. FailedLoginException.java
2. UserAlreadyExistsException.java
3. UserDoesNotExistException.java

* Club.seedymusic.jpa.bean

1. Account.java
2. Cd.java
3. Order.java
4. OrderItem.java

* Club.seedymusic.jpa.dao

1. AccountDAO.java
2. CdDAO.java
3. OrderDAO.java

* Club.seedymusic.jpa.OrderController

1. OrderControllerServlet.java

* Club.seedymusic.test

1. TestDAO.java
2. TestDAOAccount.java
3. TestDAOGet.java
4. TestDAOList.java
5. TestServlet.java

* Club.seedymusic.util

1. ConfigurationManager.java

* Club.seedymusic.webservice

1. CatalogServicce.java
2. OrderWS.java
3. UsernameNotFoundException.java

* Club.seedymusic.wrapper

1. Dao.configurtioin.properties
2. Hibernate.cfg.xml
3. Hibernate.cfg.xml.example

* Libraries- contains the jar files that are needed in running this application in the following subfolders
* Apache Tomcat v8.5
* JRE System Library
* Referenced Libraries
* Web App Libraries

1. JavaScript Resources

* ECMAScript Built-In Library
* ECMA 3 Browser Support Library
* SeedyMusicClub

1. Referenced Libraries
2. Docs

* CatalogWebServiceTests by Ahmad Traboulsi
* Design data layer by Karim Sultan
* Design use cases by Karim Sultan
* Genesis genre weightings by Karim Sultan
* Architecture by Karim Sultan
* Project cookbook by Karim Sultan
* Project software version by Karim Sultan

1. Scripts

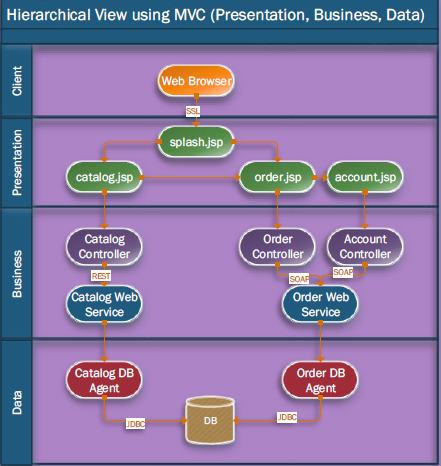
* Genesis contains assets, clean.bat, clean.sh, genesis.py, README.txt
* SQL- contains the script for creating an instance of the database; make\_seedy\_db.sql

1. WebContent

* Assets
* Covers- cover pictures of the CDs
* CSS- CSS file for styling the page; spash.css
* Images- images used in styling the page; background, icons, logo.
* Samples- audio samples of the CDs
* META-INF contains MANIFEST.MF
* Test – contains the .jsp files to display the results of the tests
* TestDAO.jsp
* TestDAOAccount.jsp
* TestDAOGet.jsp
* TestDAOList.jp
* Test.jsp
* WEB-INF- contains .jsp files the users use in interacting with the application
* Catalog.jsp
* confirmOrder.jsp
* index.jsp
* login.jsp
* loginfailure.jsp
* orderstatus.jsp
* register.jsp
* shoppingcart.jsp
* userrdoesnotexist.jsp
* classpath

# 

# Project Architecture



# 

# DATABASE MODEL

The database application used for this project is MySQL and was accessed through hibernate.

Below is the schema for the database

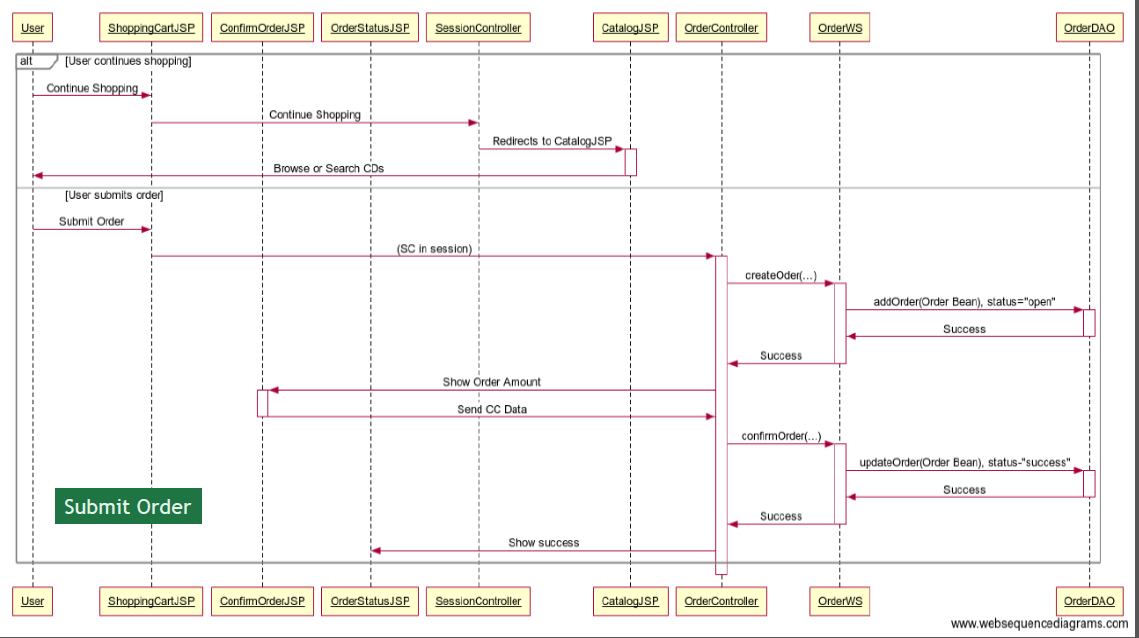


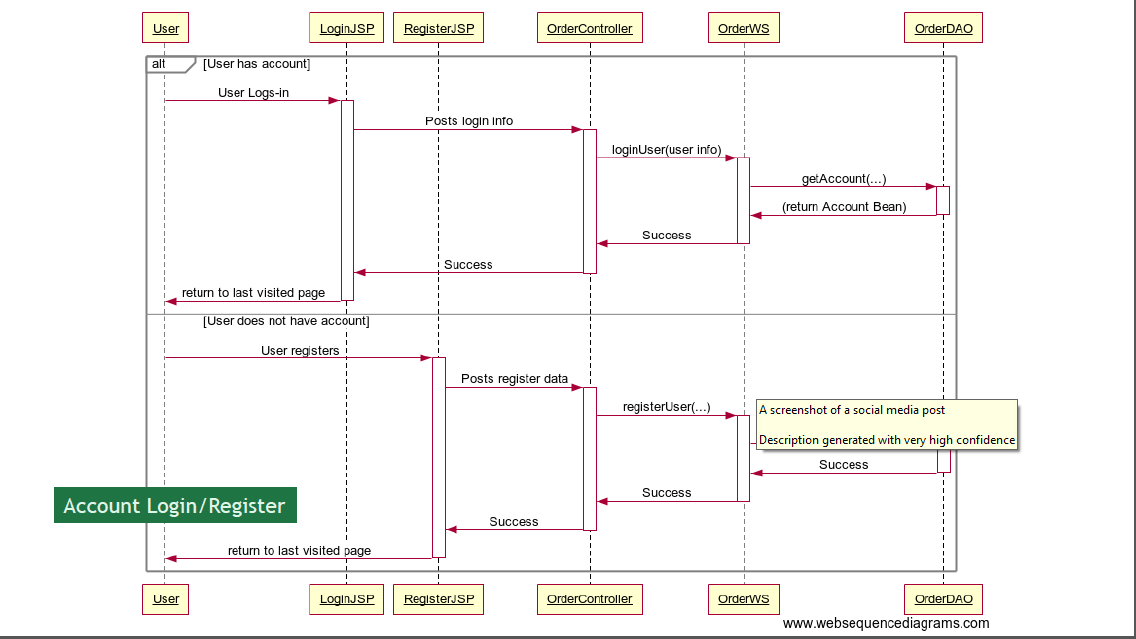
# Use Case Diagram



To browse the catalog







To buy CD(s)

# Testing

**Test Case**

**Test Case ID: Version 1.0**

**Test Environment: Locally hosted browser**

**Registration page test**

Test steps

1. Navigate to seedy music club registration page
2. Fill in the required information

* The email address must be valid i.e. contain “@” and “.com”.
* The passwords must match.

1. Click submit

Expected result: The page should redirect to the home page and automatically log in the user.

**Login testing**

**Pre-condition**- User is already registered on Seedy Music Club.

Test steps

1. Navigate to seedy music club login page.
2. Enter the email address of the registered user.
3. Enter the password of the registered user.
4. Click “Log In”.

Expected Result- The page should redirect to the previous page.

**Browse Page testing**

Test steps

1. Select a particular genre from the genre dropdown.

OR

1. Search for CDs according to name.

Expected Result- The page should load samples of CDs from that category or CDs matching the name searched. Each CD should contain a cover, sample that can be played by the user, genre, band, price and a button that allows the user to add to cart.

**Shopping cart Testing.**

**Pre-condition** – User is already properly signed in.

Test steps

1. Navigate to browse page/ my music store.
2. Add CD choice(s) to cart.

Excepted Result- The page should redirect to the user personalized shopping cart containing the selected CD.

The user should have the option to either; continue browsing the CD lists and add more CDs to cart or checkout.

The total price should increase according to the CDs selected.

The user should be able to delete CD from the cart and the total price should reflect the change.

**Payment Platform Testing**

**Pre-condition-** The user already has a selection of CD(s) in the shopping cart and is ready to check out.

The user is already logged in.

Test steps

1. The user checks out
2. Enters credit card information
3. Enters shipping address

Expected result- The user should get a message confirming successful transaction.

Note- The payment platform is hard coded to fail on every fifth transaction