Specification Document for
Capstone Music Store

Group 1: Pratik Acharya, Paul Aguilar, Alex Schultz, Gabriel Villar

Simplilearn Capstone Project

April 9, 2021

GitHub: https://github.com/pauleaguilar/capstone-music-store

Table of Contents

Application Overview 2

Running the program 3

Navigating the Screens 4-6

Sprints 7-10

Application ERD 11

Core Concepts and Future 12

Application Overview

The Music Store application is web-based and is both desktop and mobile-friendly. The application functions as a digital store for users to purchase music products. All of the details respective to each page can be examined in detail within the “Navigating the Screens” section. From a broad perspective, users can browse the product catalog by scrolling or entering a search, edit their shopping cart, and check out. Managers have the capabilities to manage categories and manage products. Administrators have the same features as managers, as well as the abilities to add administrators and managers and manage the list of all users, regardless of their security privileges.

Running the Program

1. In MySQL Workbench, run ‘create database “shoponline”;’
2. Run the provided SQL
3. For the icons to function, you will need to create an account at fontawesome.com
    a. Once an account is created, they will provide you with a snippet of code to plug into your code. Copy the code
    b. Navigate to the fragments.html file and open it with Notepad or your text editor of choice
    c. Within the bounds of the <head> tag, you will see a tag that says <script>; delete this whole line of code and paste the line of code you were provided. Now your icons should function!
4. Clone the project folder into Eclipse
5. Edit application.properties with your own SQL credentials as needed
6. Build the Maven project
7. Run the project as a "Maven Build..." and enter "spring-boot:run" in goals
8. Give the program a minute to boot
9. Enter "localhost:8090" in your browser of choice

Navigating the Screens

NOTE: Screenshots for Management users have been omitted as their functionality is essentially the same as admin, except with limited tabular accessibility. Screenshots of Management functionality would be redundant. Please reference admin and management screenshots for further details.

- Home.png
    - This serves as the landing screen for users accessing the site
    - Users are redirected here by clicking the Home button in the top left corner
    - Here, users can browse all of the products and narrow them by category
- HomeFilter.png
    - This is the home screen with the Instruments filter selected
    - As shown, this limits the product view to the instruments category
- HomeMobile.png
    - When the screen is shrunk to a mobile-friendly screen size, the navigation bar is condensed to a hamburger menu
    - In this image, the screen is shown with the hamburger menu icon selected
- HomeSearchFAIL.png
    - When a user attempts selecting search with any empty field, they are prompted with this notification
- Login.png
    - The default login screen
    - Users selecting the shopping cart from the home screen without being logged in are redirected here
    - Logged in users selecting Log Out are redirected here
- LoginFAIL.png
    - An error message if a user enters invalid credentials within the login screen
- Forbidden.png
    - Users attempting to access pages outside their scope of authority are met with this 403 Forbidden error
- Search.png
    - When the user enters a search, they are navigated to this search results page

    - In this example, I searched “Taylor Swift”
    - This search utilizes the iTunes API and searches against the iTunes registry
- SearchSelection.png
    - Selecting a song will open a snippet of the selected song in a new tab to listen to
- CartEmpty.png
    - When a user logs in for the first time, their shopping cart will be empty as shown
- Cart.png
    - When a user selects Add to Cart, the item is added to their cart and they are redirected here
    - In this example, I selected adding a guitar to my cart. Selecting Remove will result in CartEmpty.png
- Order.png
    - Once a user selects Check Out within their cart, they are navigated to this success screen and their cart is emptied
    - Navigating to Cart from here will result in CartEmpty.png
- HomeAdmin.png
    - The home screen layout for an admin user
    - Note that the admin now has access to Admin and Management tabs at the top of the screen
    - Managers would only see a Management tab
- Admin.png
    - The default screen for the Admin tab
- AdminCategory.png
    - The screen for Manage Category from Admin.png
    - Here, administrators can create, update, or delete categories
- AdminCategoryADD.png
    - For proof of concept for AdminCategory.png, a category named Vehicles was added
    - Selecting Delete will result in AdminCategory.png
- AdminCategoryUPDATE.png
    - Editing the new category from AdminCategoryADD.png will lead us to this page

    - The field is prepopulated with the category name to be changed
- AdminCategoryUPDATE1.png
    - After editing the category name from AdminCategoryUPDATE.png to Vroom Vroom, this change can be seen in this screenshot
    - Selecting Delete will result in AdminCategory.png
- AdminProduct.png
    - The first half of the screen for Manage Product from Admin.png
    - Here, products can be updated, deleted, or added
    - Deleting a product will result in this page but with the product removed
- AdminProduct1.png
    - The second half of the screen for Manage Product from Admin.png
- AdminProductADD.png
    - After adding a fake product, it is appended to the end of the product registry as shown
    - Updating the product will lead to the same screenshot but with any changes appended
- AdminAdd.png
    - The screen for Add Admin/Manager from Admin.png
    - This is also the screen for when a user creates a new account
    - Creating a new user will redirect the admin to the home screen
- AdminList.png
    - The screen for User/Manager/Admin List from Admin.png
    - This displays all known users and their respective information
    - This includes the test data entered within AdminAdd.png
- Management.png
    - The screen for displaying management abilities
    - Manage Category is the same as AdminCategory.png
    - Manage Product is the same as AdminProduct.png
    - If a manager is logged in, they would not be able to see the Admin tab within the navigation bar

Sprints

Sprint 1 – April 5th, 2021

Our goal in the first Sprint was to discuss the project's needs and goals. We decided that the most efficient way to start the project was by implementing the registration option for the page. We discussed if we wanted to use a framework for the front-end or if we wanted to just use html/css. We agreed to use Bootstrap for the front-end design of the page. The page header displays the title of the web store and has a navigation bar that allows easy access for the user.

User Action 1	
Name	Start application
Trigger	Getting the index page
Action	Open homepage that allows the user to browse the product catalog
Notes	Must allow user to continue to next page

User Action 2	
Name	Allow user registration
Trigger	User registers
Action	After the user does the input, the website will return "registration completed/successful" and the respective database is updated
Notes	Allow the user to return to the homepage if desired

Sprint 2 – April 6th, 2021

This sprint is focused on creating the login page for the user and admin. The main goal is to program the functionality that allows the user to successfully login. After a successful login they will have the option to return the homepage. We decided to use a loginController using Spring, that will communicate to the tomcat server with GET and POST requests.

User Action 3	
Name	Show login page for the admin
Trigger	User enters admin credentials on login page
Action	User is logged in with admin privileges
Notes	Navigate user to home screen

User Action 4	
Name	Allow admin to add users/admins to database
Trigger	Admin attempts adding a new user/admin to database
Action	Admin is prompted with a successful or denied outcome
Notes	

Sprint 3 – April 7th, 2021

This sprint is focused on creating the products, adding the ability to add pictures to the products, and a category. The main goal is to program the functionality that allows the admin to successfully add a product. We decided to use a Service using Spring JPA Repository, that will use CRUD operations for the functionality of the admin duties.

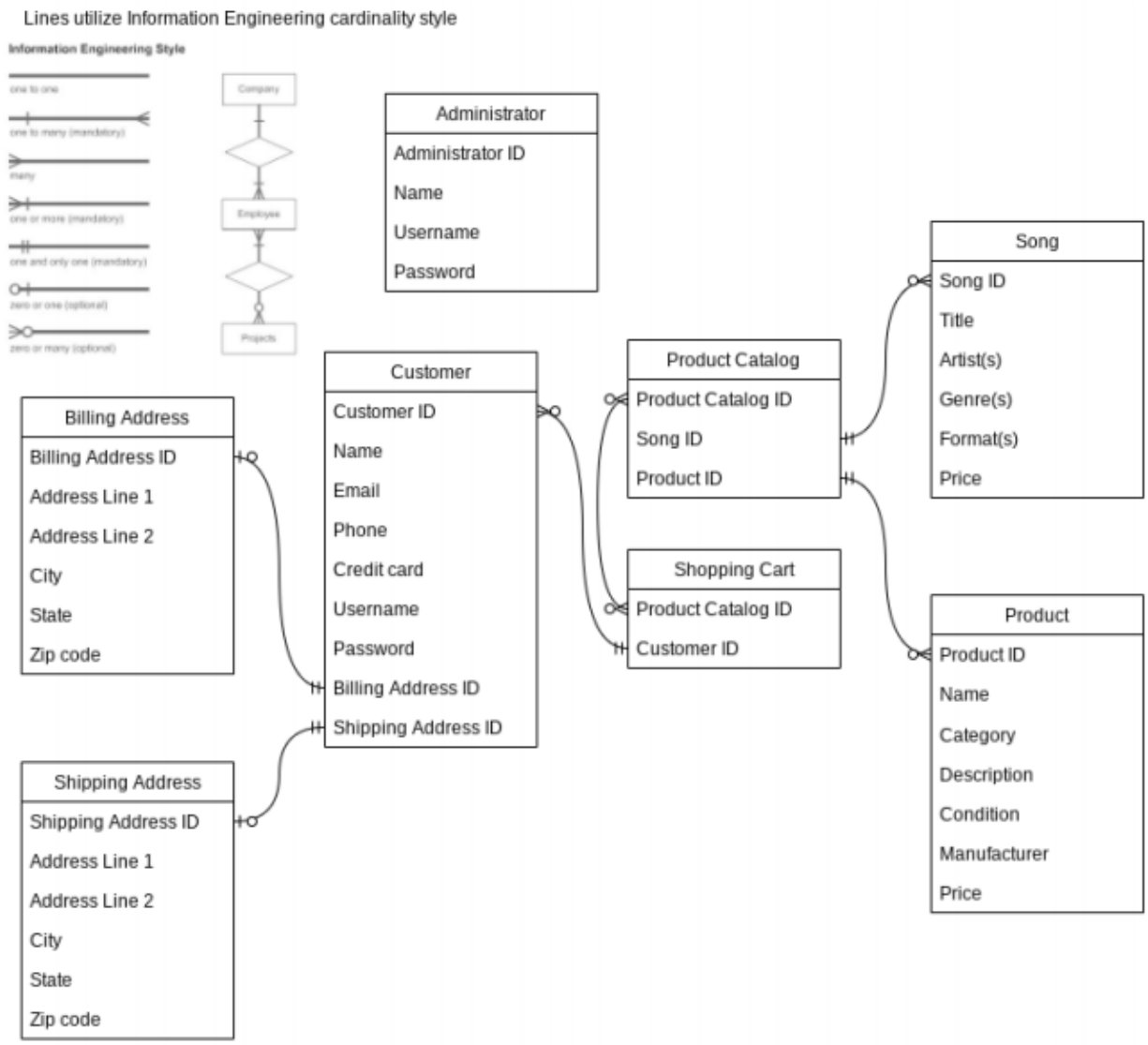
User Action 5	
Name	Create addProduct and addCategory pages
Trigger	Admin selects quit within the add product or add category pages
Action	Any changes are saved and the admin is notified
Notes	Once the user selects quit, this implies they have finished making changes to the products and categories

Sprint 4 – April 8th, 2021

This sprint is focused on creating the add to cart and checkout features of the website. We need to design the cart that shows all of the products that will be added and that the total amount adds up correctly, so that the user can checkout successfully. We decided to use a Service using Spring JPA Repository that will use CRUD operations for adding, deleting a product in the cart.

User Action 6	
Name	Create addtoCart and Checkout pages
Trigger	The user has at least one item in their cart and selects checkout
Action	The user receives a confirmation page that their order has been placed
Notes	

Application ERD



Core Concepts and Future

On the front-end, Bootstrap is used to help make a clean user interface and experience both graphically and functionally, which is complemented with icons provided by FontAwesome. Bootstrap easily enables us to employ a mobile-first methodology to designing the front-end of our website while harnessing its extensive toolkit capabilities. Moving to the back-end, to ensure saved data persists between sessions and users, a MySQL database connection is achieved by using Hibernate and its methods. The SQL consists of multiple tables collaborating under one database. As for application security, this is achieved by use of Spring Security within the Spring Boot Framework. This security validates all logins against the aforementioned SQL database and grants each user their respective privileges. In turn, this prevents unauthorized users from accessing webpages outside their scope of privilege. This security extends itself from the front-end to the back-end by encoding each user's password. If any user gains access to the database, each user's password is saved as an encoded assortment of various characters. With this in mind, please note that if you would like to manually add test users to the database, this cannot be done within the SQL database as each password needs to be encoded first. For searching songs, the API for iTunes is utilized. Search results and compared against the iTunes music registry and the respective results and presented.

To improve the application, features can be added to the application among subjective sundry graphical changes. When logging in, a "forgot password" functionality could be implemented, which would require the application to send an email to the user with some form of further instructions. With this implemented, verification of newly created accounts via email could be easily added as well. Going further from these changes would delve more into the subjective realm of graphic design and UI/UX design rather than back-end core functionalities. For example, pop-ups prompting the user to sign up and save money on their next purchase are not a key feature for the website to function as intended. The sky is the limit for what features can be implemented, but as for improving the core functionality of the program, password recovery and account validation both via email should suffice.