

# Project: Music.ly Capstone Project Application

## Team Members:

1. David Grove
2. Karthiga Gujuluva Ravichandran
3. Laura Hernandez
4. Matthew Hernandez

Created on Date: 03/05/2021

---

## Prototype:

The Music.ly Application consists of two modes (One for User and other for Admin) with functionalities in each mode. The user interaction will be through a webpage of link ([Welcome Page](#)) which leads to Welcome Page where the user or admin can register, or login or Anonymous user can just view the products. User has privileges to view, order products or add products to the cart and checkout and receive a receipt. As it is an Instruments Store, the Product table does not need to have category changes as instruments don't change category.

Admin has privileges to view user details and edit, delete, and add products and change the order status. Web Security is implemented in this project. Passwords are encrypted and page is opened only when the privilege is given to the user or admin. The final Application will be developed in various sprints based on this prototype. The prototype explains the functionalities and format of the final project.

## Sprint Plan:

Task Manager Application project is divided into **4 sprints**. Each sprint consists of **1 week** of time. The tasks achieved in the sprints are given below:

## Sprint 1:

Time allocated for this sprint is **40 hours**. The tasks involved in this sprint are:

## Tasks:

1. Development of Welcome screen, Login screen, Register Screen.
2. Welcome screen should have 4 options, Home, Products, Login and Register.

3. Login screen has 2 inputs namely username and password and a submit button to proceed to Welcome page.
4. The username and password should be required fields and can accept all characters.
5. Login button searches if user already exists in the database and proceeds to login success screen with Web Security module.
6. Database is created for the application and fed with initial data.
7. Linking all screens and the database through application. Properties file.
8. JSP is used to create screens and Service, model, Repository, Entity, Security Configuration files are used to get data from DB, validate and display on the webpage.
9. User, Product, Order, Role tables are created to store all data and they are fed with initial data.
10. Role table is fed with initial data for registration to proceed.
11. Development of registration screen that has name, username, password, email, phone, address, apartment no, city, country, state, zip code, and a register button.
12. Register button leads to Successful registration screen. All are required fields.
13. Success screen has a button that leads to Login Page.
14. Error messages are thrown in login screen if incorrect data is given in the fields.
15. Save method is created to save the user data to table.
16. Find user functionality checks if user already exists in table.
17. Appropriate messages are thrown to the user if save happens or fails.

## **Sprint 2:**

Time allocated for this sprint is **40 hours**. The tasks involved in this sprint are:

### **Tasks:**

1. User Functionality is handled in this sprint. The Welcome Page for user after login will have 5 options, Home, Products, Cart, Logout, Welcome: (Username).
2. Logout makes the user to logout and it leads to Welcome page of Anonymous user.
3. Products option leads to Products page where all the products are loaded along with product data.
4. It also has a search input box with a search button. Any word written in the search input is searched in the columns of name, category, condition, and appropriate results are shown in the table.

5. The table has a view detail link that leads to View Detailed information for each individual product.
6. Price column shows unit price of individual products.
7. Navigation bar and Footer are provided for better access to screens.
8. When View Detail is clicked for a product, the View Details of Product page appears.
9. This page has the details of the product along with the image.
10. It also has 2 buttons, Back and Order button.
11. Back button leads to Products page.
12. When Order button is clicked, the product is added to the cart table with Cart status.
13. An alert popup comes that confirms that product is been added to the cart table.
14. It then leads to the View Products page with all products.

## **Sprint 3:**

Time allocated for this sprint is **40 hours**. The tasks involved in this sprint are:

## **Tasks:**

1. Cart functionality is implemented in this sprint. Cart page has Clear cart button, Check Out button, Continue Shopping button.
2. Clear cart removes all the products that were added to the Order table with Cart status.
3. Continue Shopping leads to the View Products page.
4. There is a Remove button link in each of the products added to the cart.
5. The Remove link deletes the product from the database and removes it from the cart table.
6. Check out Button leads to Order Receipt Page.
7. The functionality involved here is calculating the price of each individual items and the Grand Total price of the cart of that user.
8. The quantity of the items is checked and calculated with unit prices to compute the final grand total.
9. The Receipt has the Shipping address, Shipping date, the products ordered and their quantity and prices and Grand total price.
10. This page also has 3 buttons namely, Back, Submit Order and Cancel buttons. Cancel and Back buttons lead to the cart page.
11. Submit Order button leads to the Order Confirmed Page.
12. Submit order changes the status of each order of the user that has Cart status to Ordered status.

## **Sprint 4:**

Time allocated for this sprint is **40 hours**. The tasks involved in this sprint are:

### **Tasks:**

1. Administrator functionalities are done in this sprint.
2. Welcome page of Administrator has 5 links namely, Customers, Logout, Orders, Products, and home.
3. Logout makes the admin to logout and leads to home page.
4. Customers link leads to Customers Page that has all the details of all the customers.
5. Products link leads to the Products page that has all the details of all the products available in store.
6. Each product has Delete and Update buttons.
7. Delete button click deleted the product from the store table.
8. Update product button leads to Update product page where admin can change the product name, category, condition, price, and image and click submit button.
9. This updates the product with new details and takes to Welcome page of Admin.
10. Admin can also add new product to the store by clicking Add new Product button and it leads a Add Product page.
11. Admin can give all details of new product and save to the store by clicking submit button.
12. It then leads to All Products Page.
13. Orders link leads to Orders Page which has all the orders with Ordered status.
14. Each order has an update button which leads to Update Order Page.
15. Update Order Page has the information of that order and admin can edit the status column alone to change the Order status to Shipped or whatever status.
16. The submit button in the update order page leads to the Products page of admin.
17. Password encryption is done to store the password credentials using MyUserDetailsService methods.
18. Logger is used to log all the tasks done by the user.

### **To - Do List:**

1. Make the application mobile responsive.
2. Admin should be able to delete customer.

3. Custom exception handling should be done.
4. Performance improvement can be done.
5. Bug Fixing and Junit Testing should be done.

## Stack Used:

- Backend: Java8, Spring Data JPA, Hibernate, Spring Web Security, MySQL Database, Spring Boot
- Frontend: Bootstrap, CSS, JSP, JS

## Queries Used:

- Insert into roles(role\_id,role) values(1,"ROLE\_USER");
- Insert into roles(role\_id,role) values(2,"ROLE\_ADMIN");
- Insert into store  
(product\_id,product\_category,product\_condition,product\_image,product\_name,product\_price) values(1,"A","new","i1","A",1);
- Insert into store  
(product\_id,product\_category,product\_condition,product\_image,product\_name,product\_price) values(2,"B","new","i2","B",2);
- Insert into store  
(product\_id,product\_category,product\_condition,product\_image,product\_name,product\_price) values(3,"C","new","i3","C",3);

## Music.ly Capstone Project Application - Link

[Karthiga-web/Capstone \(github.com\)](https://github.com/Karthiga-web/Capstone) is the repository link for my Music.ly Capstone Project Application in GitHub.

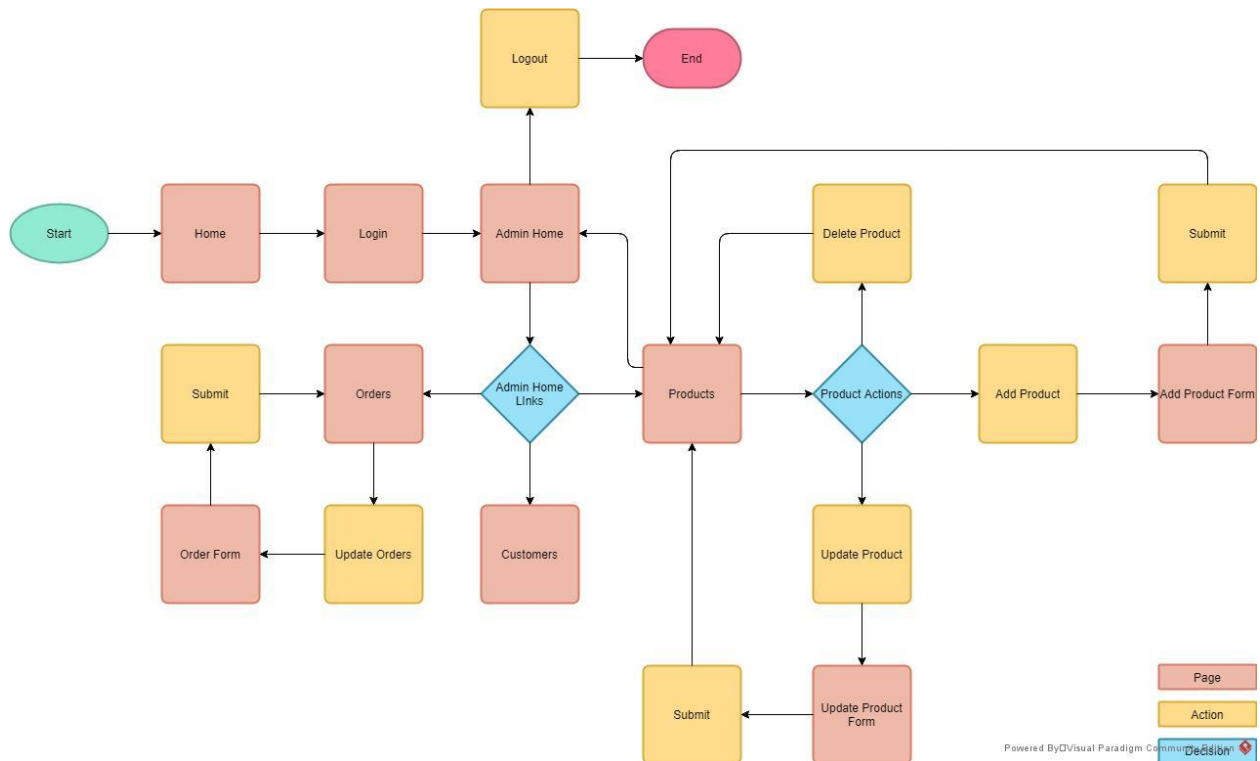
## Algorithm:

1. The project opens with Welcome Page made with Bootstrap and options to login, register and view products.
2. The Anonymous user can view products, product details but cannot order a product.
3. Registration page is where the user enters all the details and clicks submit button and the user is registered, and it leads to Successful registration page.
4. The Home button in Successful registration page leads to Login Page where the user is requested to login with username and password.
5. Invalid username and password and validated by Web Security and appropriate error messages are thrown.
6. Successful login leads to Welcome Page for User and now the user can view products, product details and order them or add to cart.

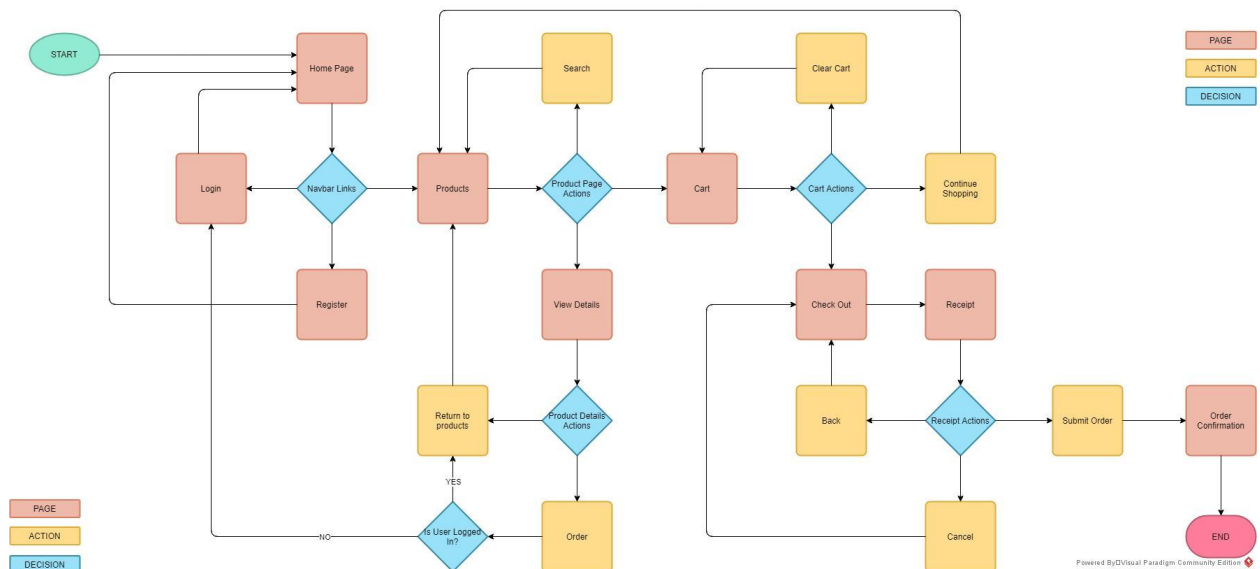
7. The User views the products and adds them to cart. He can add as many products needed and any number of products he may require.
8. The cart page has all the products he had added from previous sessions and current session.
9. The user has ability to remove products from cart page with remove option and he can also clear the cart.
10. The user can check out from cart page if he wants to check out.
11. The Submit Order page shows all the details of all the products added by the user to the cart.
12. The price, total price and quantities are shown in the receipt.
13. The user can now either click Back or Cancel to go back to the cart page or Submit Order to Submit the order.
14. The submit order leads to Order successfully ordered page.
15. This page again leads to Welcome Page for users to make another purchase of products.
16. Admin has ability to change the status of the order to Shipped or any status.
17. Admin Welcome page has options like Products, Customers, Orders and Logout.
18. Admin can view customer details by clicking the Customer option or Logout by clicking logout.
19. Admin can view the Orders by clicking orders option and edit each order by clicking the update button on the order and change the status of the order.
20. Admin can view the product details by clicking products option.
21. The products page of admin has all details of each individual product and has options to edit and delete the product.
22. Delete button deletes the product from the store and again shows all the products.
23. Update button leads to update page and the admin can change product details like cost, name, category or add new product.
24. The admin can add a new product by uploading image, product details and save to the store.
25. Bootstrap, CSS, JSP are used to design the Frontend.

## **Flowcharts:**

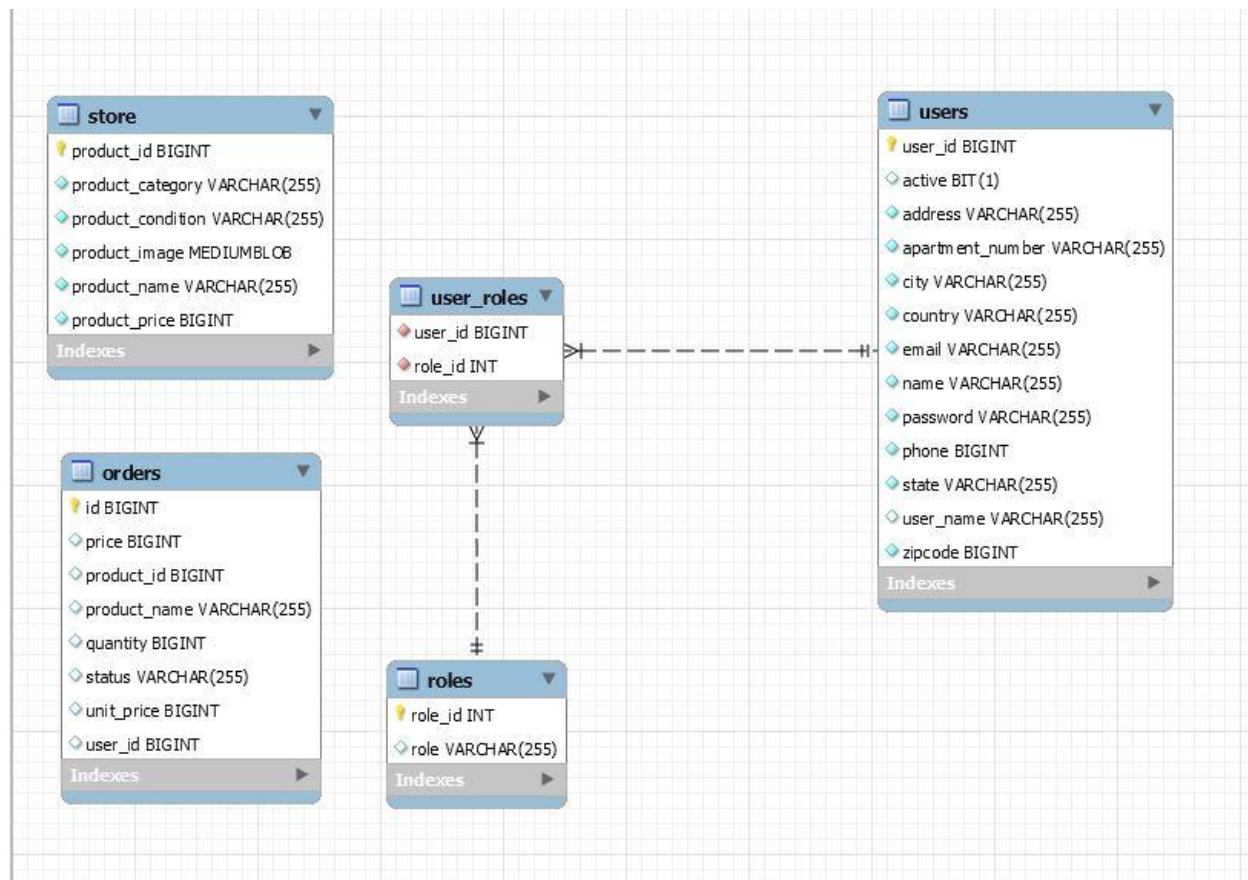
### **Admin Flowchart:**



## User Flowchart:

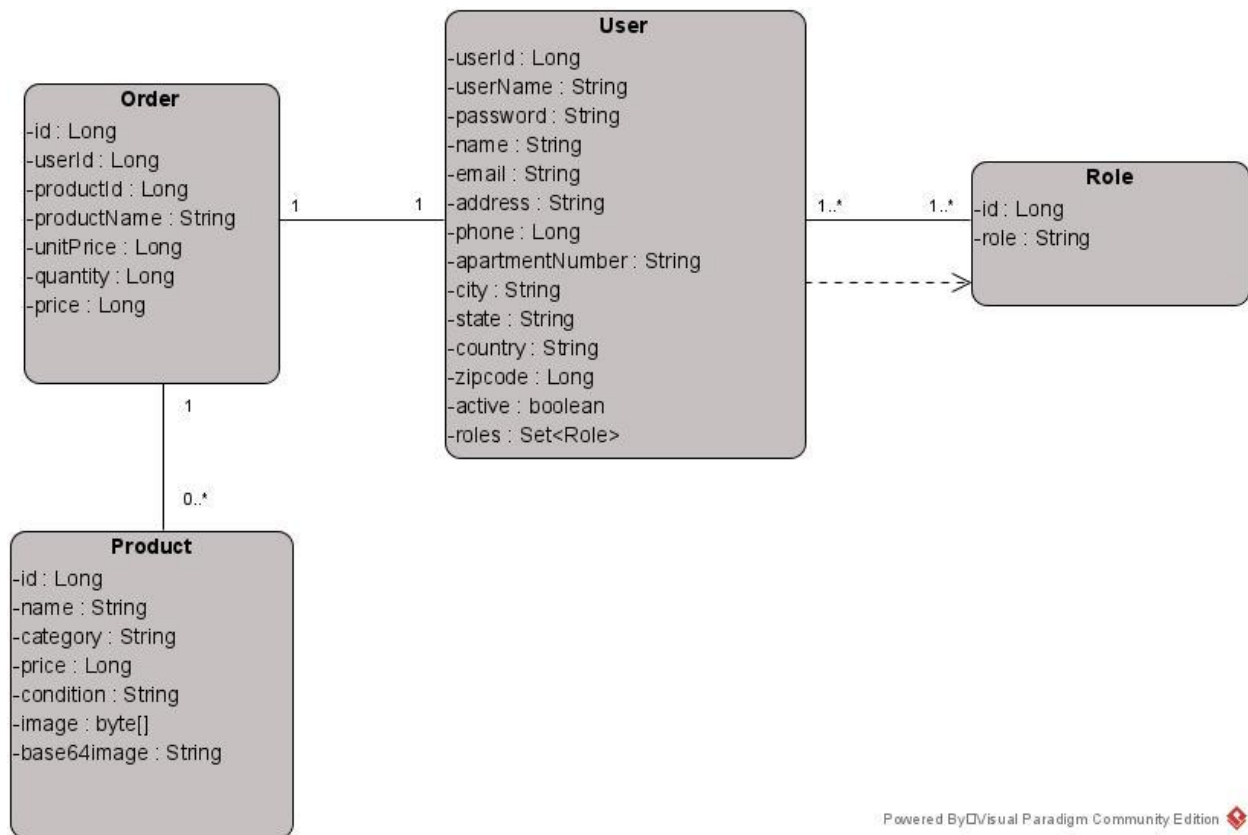


## Entity Relationship Diagram:



**Class Diagram:**





## Java Concepts Involved in the project:

1. Classes
2. Packages
3. Spring Boot Framework
4. Logger – slf4f Logger and Logger Factory
5. Web Security
6. Password Encryption
7. Service-Model-Controller-Repository-JSP
8. Multipart File Image Handling
9. Bootstrap
10. CSS, JSP
11. JPA repository

## Conclusion:

Music.ly Application does not crash for incorrect inputs or any discrepancies caused by the user or admin. It has better performance as it uses Service, repository, Controller, jsp and logger to log data. It encrypts the password and uses Web security to provide access to pages for users.

