1. **System Requirements**

Sporty Shoes is a company that manufactures and sells sports shoes. They have a walk-in store, and now, they wish to launch their e-commerce portal sportyshoes.com.

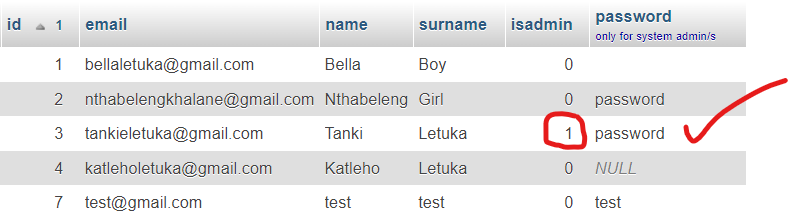
**System Specifications**

***Login / Logout***

* The system opens with a login page and immediately takes the user to the products administration on successful login. **\*\*\*PS:** ***the system is hardcoded to take the user “***[***tankieletuka@gmail.com***](mailto:tankieletuka@gmail.com)***” and password “password” as the initial detail for assignment submission but a new user can always be added to test the authentication.* \*\*\***
* On successful login, the system will display who has logged in and with an option to logout on every page visited.
* On unsuccessful authentication the system displays an error and user has an option to login again.
* On logout, the system goes back to the login page with a message showing a successful logout; user has an option to login again.

***Admin Registering***

For the purpose of this project, the admin can only be registered from the backend, directly on the database in the table, “user” as follows. Then from there, the admin can change their password to whatever the admin desires.

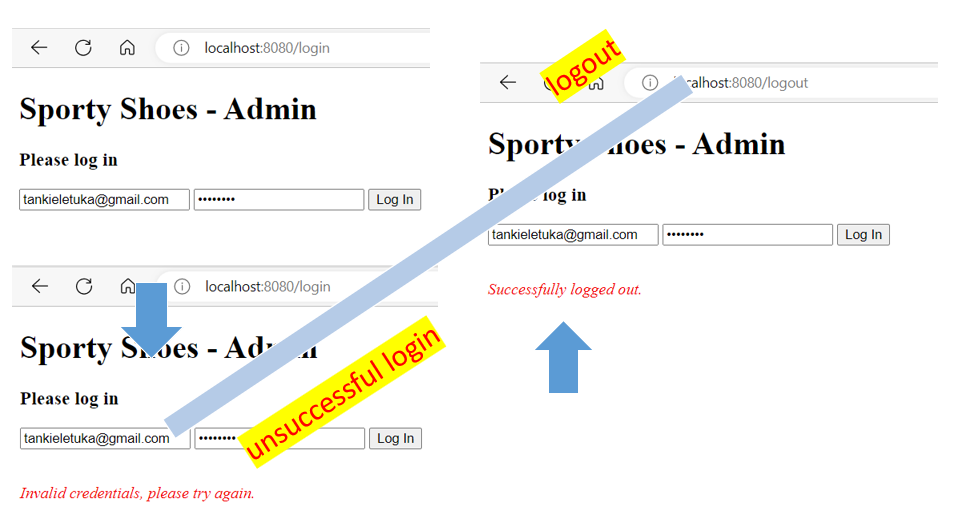


**Main Menu**

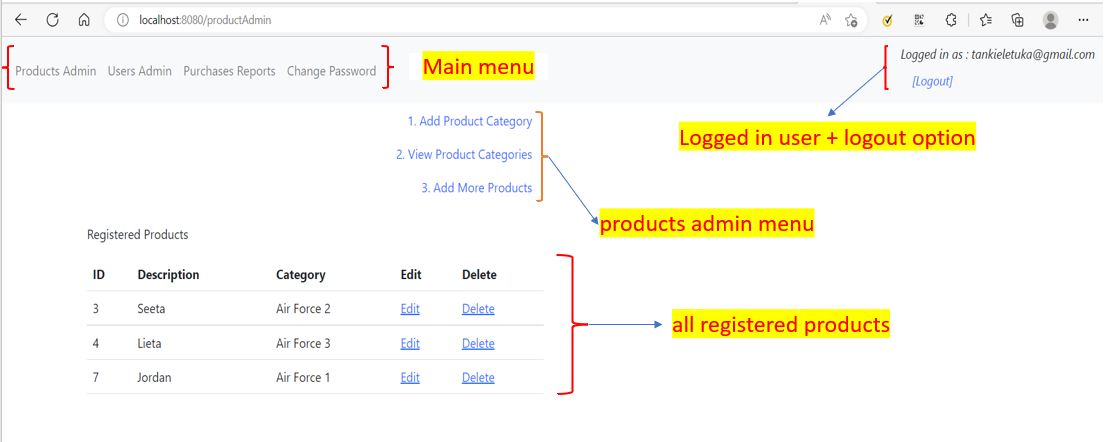
* **Products Admin** – Manages Products and Categories - opens with the list of all registered products
  + Sub-menus
    - Add Product Category
    - View Product Categories
    - Add More Products
  + Product list has an option to **EDIT** or to **DELETE** product.
* **Users Admin** – Manages users. Also opens with the list of all registered users.
  + Sub-menus
    - Register More Users
    - Search Specific User
* **Purchases Reports** – view all purchases made.
  + Sub-menus
    - Purchases By Date
    - Purchases by Date and Category
    - Purchases by Date and User
* **Change Password** – for the logged in admin to change their password

**Detailed View with Screenshots**

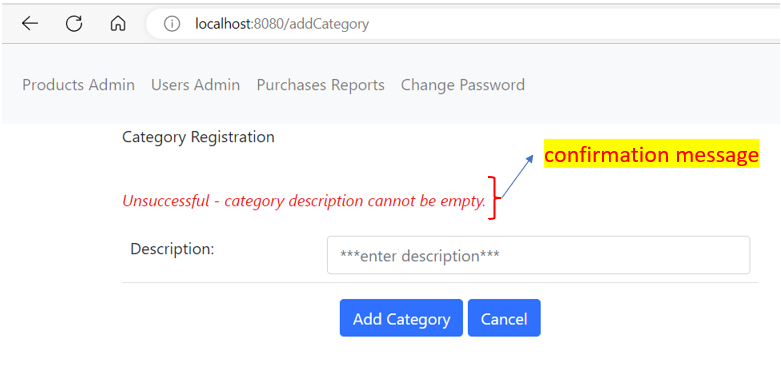
1. **Login / Logout**



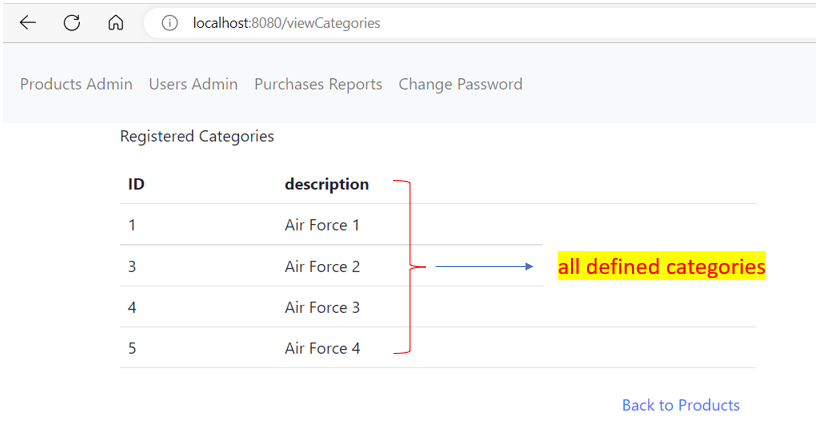
1. **Products Admin** – After successful login – the system goes to the **Products Admin** page.



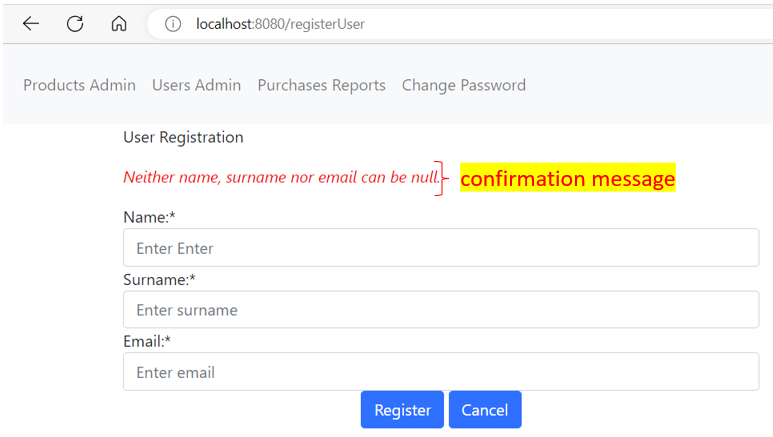
* 1. **Add Product Category** – provide the category description and “Add Category” OR “Cancel” to go back to the Products Admin. The system will give the confirmation of the action whether successful or not.



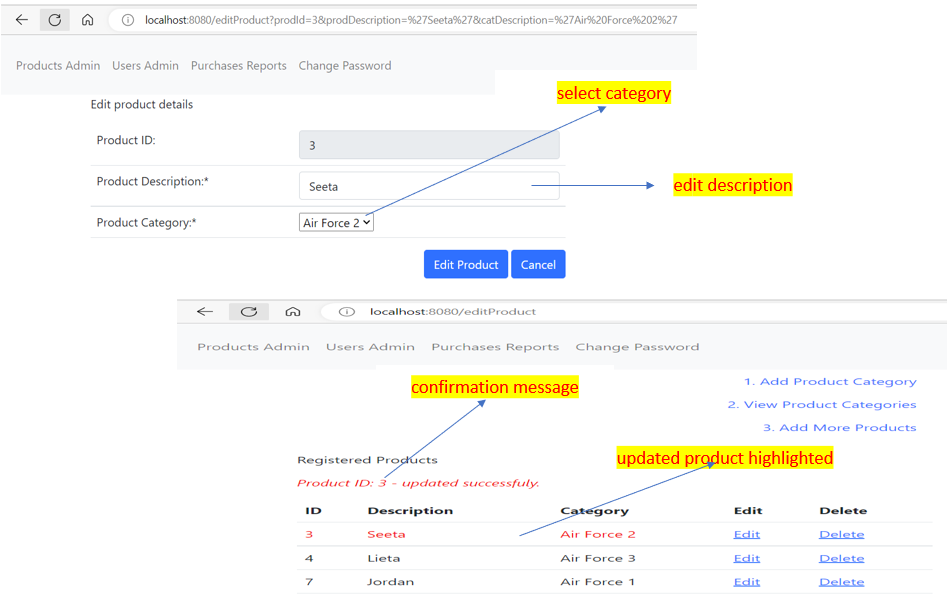
* 1. **View Products Categories** – Click on “Back to Products” OR the “Product Admin” link to go back to Products Admin



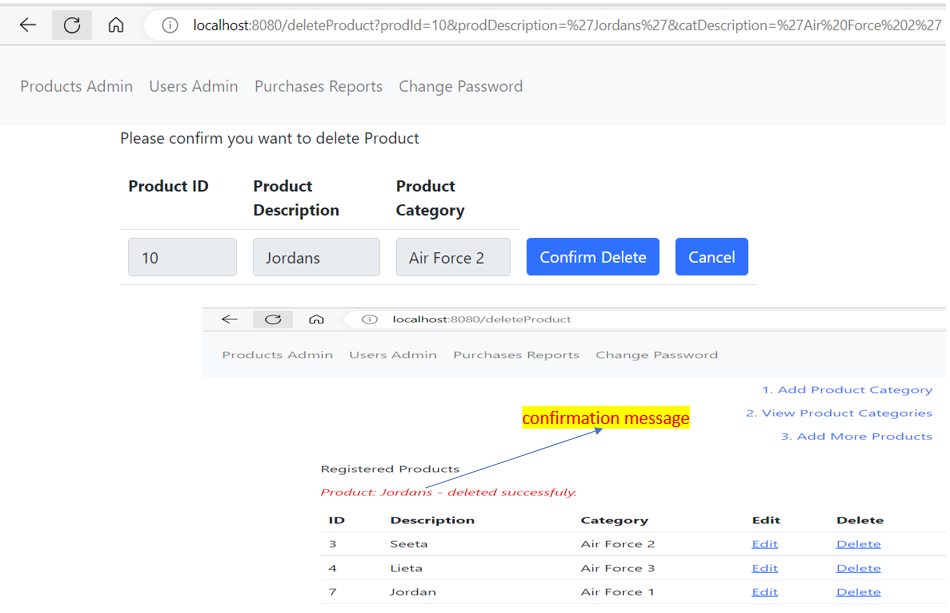
* 1. **Add More Products** – Add the product description and select the product category then “Add Product” or “Cancel” to return to the Products Admin page.

****

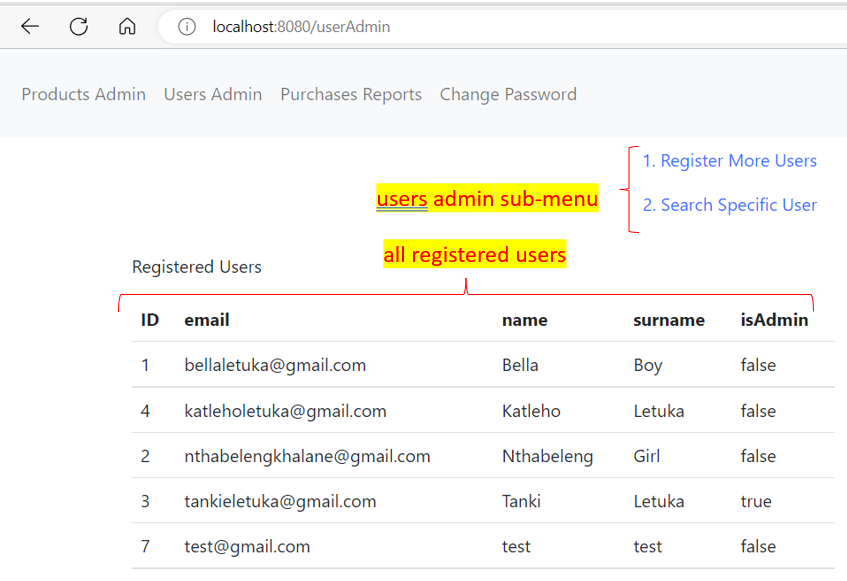
* 1. **Edit** – edit product description and/or category and “Edit Product” or “Cancel” to go back to Products Admin.

****

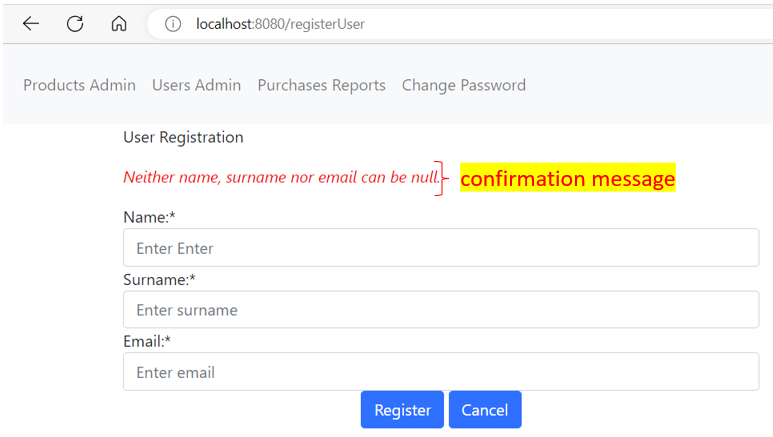
* 1. **Delete** – delete a registered product. Click on “Delete” and then “Confirm Delete” OR cancel to go back to the Products Admin. On Deletion, the system will displace the confirmation message.

****

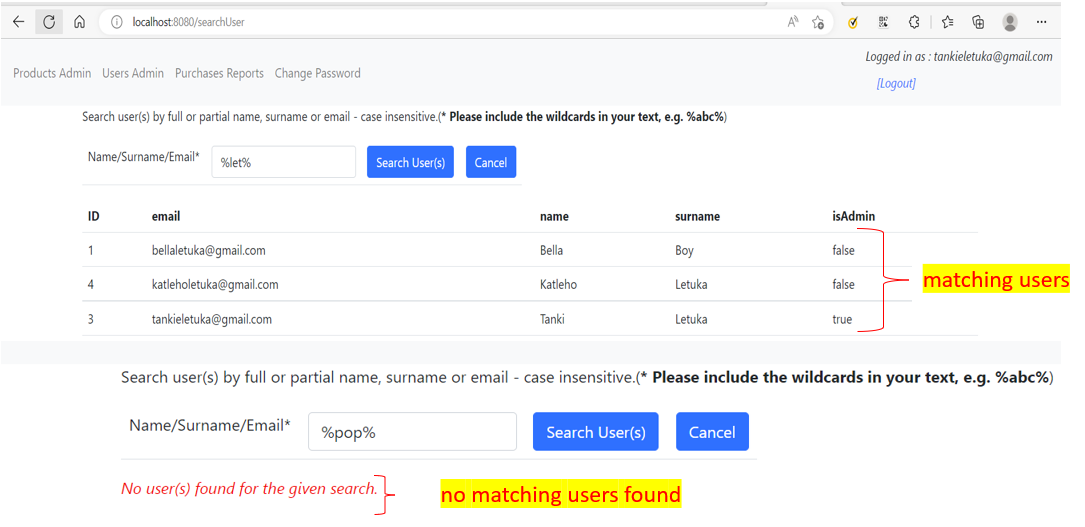
1. **Users Admin** – Management of users by the admin



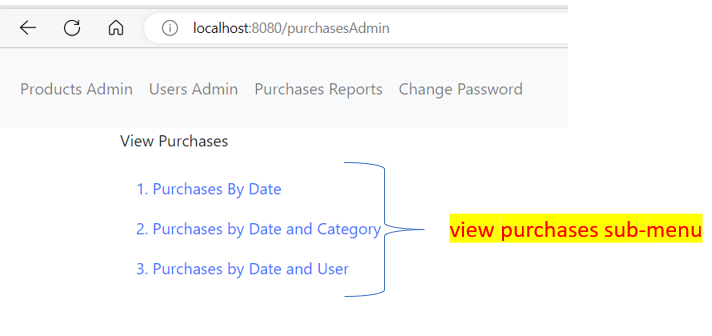
* 1. **Register More Users** – enter name, surname and email (unique) and “Register” or “Cancel” to return to the “User Admin” main menu. The system will give the confirmation of the action whether successful or not.



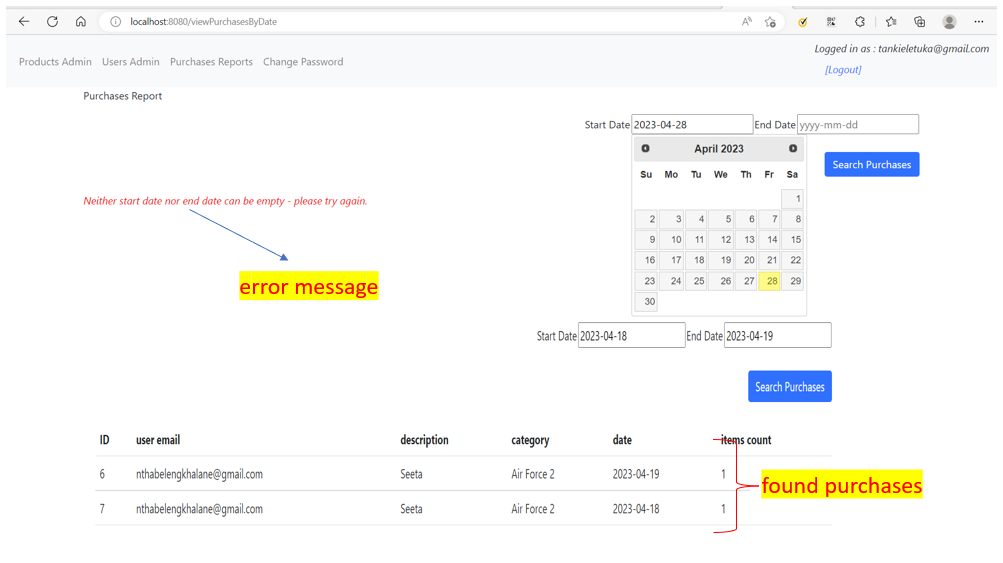
* 1. **Search Specific User** – search user by specified characters in wildcards, eg… %let%. These could be full or partial values and the system will search ***on name, surname and email ids***. The System will then display users found or an error message below depending on the search.



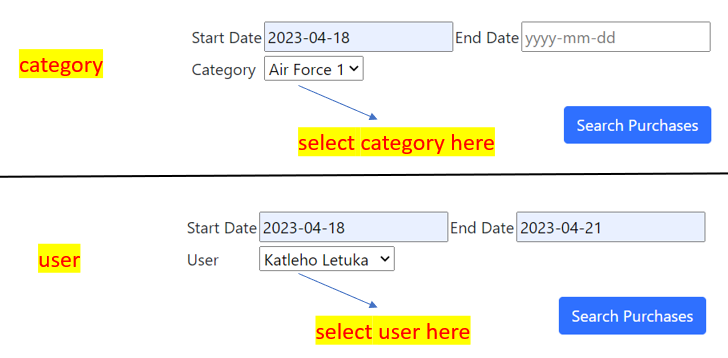
1. **Purchases Reports** – view purchases reports.



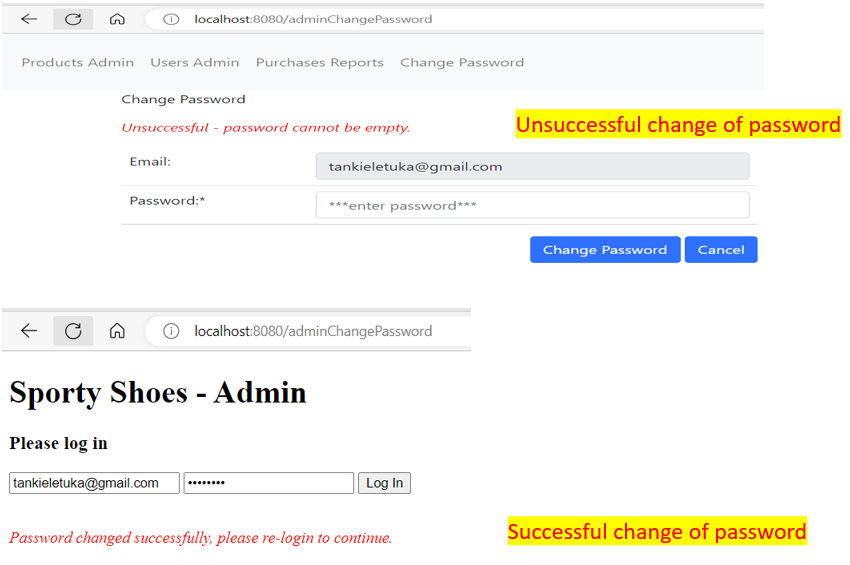
* 1. **Purchases by Date** – specify the start and end date by using the date picker and “Search Purchases”. The system will either display an error message or the purchases if any are found.



* 1. **Purchases by “Date and Category” or “Date and User”** – specify the start date and end date using the date picker and select the category of products OR user, then “Search Purchases”.

****

1. **Change Password** – Logged in Admin change password. The system will display the error message on an unsuccessful attempt OR will take the user to the login page to log in with the new password.



1. **Scrum Documentation**

**Sprint One – (Entities, System design and Frameworks)**

**Sprint planning**

**Goal**: Identify the framework to use for the project and entities required.

**Target**: 4 days

Framework: Spring Boot

Identified objects/entities

* ***user*** (id, name, surname, email, isadmin, password)
* ***category*** (id, description)
* ***product*** (id, description, category id),
* ***purchases*** (id, product id, user email, date, count)

Packages

* dao – data access objects, for access to specific information from the entities (users, purchases and products)
* entity – for all entities identified
* mvc – for all controllers (add, log, modify and view)
* repo – for all interfaces to for database CRUD operations
* service – for the system login for all entities

Database

* tables (category, product, purchases and user)

**Daily Scrum**

|  |  |
| --- | --- |
| **Day 1** | Object/s identified and system logic mapped |
| **Day 2** | Objects relationships mapped – controllers identified |
| **Day 3** | Entity classes created |
| **Day 4** | Testing of entity classes and connection to the Database |

Sprint Review / Introspection

* User hibernation instead of SQL for data access.

**Sprint Two (Database Connection and Data Access through hibernation)**

**Sprint Planning**

**Goal**: Use hibernation for all database crud operations

**Target**: 6 days

**Daily Scrum**

|  |  |
| --- | --- |
| **Day 1** | Database connection troubleshooting |
| **Day 2** | Database connection troubleshooting continues – read more on hibernation |
| **Day 3** | Database connection solved and basic functions like select all achieved. |
| **Day 4** | Implementation of the repository interface functions for specific DAO |
| **Day 5** | Revisit of the entity classes for hibernation |
| **Day 6** | Testing data access for other operations – delete and update |
| **Day 7** | Continue integration and testing for all errors |
| **Day 8** | Redesign of the data access objects to match the data structure from hibernation |

Sprint Review / Introspection

* Hibernation functionality to be explored more with the use of custom SQL in some instances though this was not necessary in the end.
* 2 days beyond target because of the need to study hibernation, specifically; include the provision in the planning going forward.

**Sprint Three (Frontend)**

**Sprint Planning**

**Goal**: Implementation of the front end and implementation of the controllers.

**AddController** – for all data additions in the database.

**ModifyController** – for all update operations

**ViewController** – for all data access

**LogController** – for logging, logout and system password change

**Target**: 10 days

**Daily Scrum**

|  |  |
| --- | --- |
| **Day 1** | Implementation of the AddController for categories |
| **Day 2** | Implementation of the AddController for products |
| **Day 3** | Implementation of the AddController for users |
| **Day 4** | * Implementation of the LogController for login and logout purposes * Revisit the AddController to include functionality for a logged in user check |
| **Day 5** | Implementation of the ViewController, for categories and products |
| **Day 6** | Implementation of the ViewController for users and purchases |
| **Day 7** | Revisit ALL controllers for URL display – redirection |
| **Day 8** | Work on improving the display on the JSP pages |
| **Day 9** | Testing for and improvement on the JSP pages |
| **Day 10** | Testing, correcting typos and finalizing documentation. |

Sprint Review / Introspection

* Consistency in the design of the controllers would make the implementation faster.

1. **Concepts Used**

* JSP
* JSTL
* Java Persistence
* Hibernation
* Lombook – for constructors on classes
* Spring boot
* JPA (CrudRepository)
* MySQL Database

1. **Improvements Recommendation**

* Deletion of products – restrict restriction so that products that have been purchased before cannot be deleted for reporting purposes.
* Database design – use of foreign keys, specifically to restrict relationships on deletion or modification of data
* Add an option of other users to login, other than Admin to add purchases (all purchases were added through the backend for this project)
* Products and Purchases do not include prices in this project (just to avoid unnecessary typing in creating products or purchases), so the improvement would be on that.
  + This improvement would have to go back to deletion of products – to design the database and all functions so that the product is never deleted but just activated or deactivated; the purchase date would have to be between the active and deactivate date so that the correct price is picked.

1. **Author Details**

Names: Tanki Letuka

Email: [tanki.letuka@vm.co.mz](mailto:tanki.letuka@vm.co.mz) (Vodacom Mozambique)

GitHub Repository: <https://github.com/tankieletuka/phasethreeproject.git>