

**PANDA MALL**

**Functional Specifications and Methodology**

**Project Advisor:**

Mr. Razi-uddin

**Group Members:**

Asjad Iftikhar 18L-0951

Muhammad Zain 18L-1109

Tayyab Waseem 18L-1017

National University of Computer and Emerging Sciences

Department of Computer Science

Lahore, Pakistan

Table of Contents

[1. Introduction 1](#_Toc86747722)

[1.1 Purpose of this Document 1](#_Toc86747723)

[1.2 Intended Audience 1](#_Toc86747724)

[1.3 Definitions, Acronyms, and Abbreviations 1](#_Toc86747725)

[2. General Description 2](#_Toc86747726)

[2.1 User Characteristics 2](#_Toc86747727)

[2.2 Domain Overview 2](#_Toc86747728)

[3. Functionality 3](#_Toc86747729)

[3.1 Functional Requirements 3](#_Toc86747730)

[3.1.1 Functional requirements for Users 3](#_Toc86747731)

[3.1.2 Functional requirements for Administrators 3](#_Toc86747732)

[3.1.3 Functional requirements for Stores 3](#_Toc86747733)

[3.2 Non-Functional Requirements 3](#_Toc86747734)

[3.2.1 Availability 3](#_Toc86747735)

[3.2.2 Usability 3](#_Toc86747736)

[3.2.3 Reliability 4](#_Toc86747737)

[3.2.4 Scalability 4](#_Toc86747738)

[3.2.5 Data Integrity 4](#_Toc86747739)

[3.2.6 Performance 4](#_Toc86747740)

[3.3 Assumptions 4](#_Toc86747741)

[4. System Architecture 5](#_Toc86747742)

[5. Use Cases 7](#_Toc86747743)

[5.1 Login Account 7](#_Toc86747744)

[5.2 Create Account 7](#_Toc86747745)

[5.3 Reset Password 8](#_Toc86747746)

[5.4 Logout 8](#_Toc86747747)

[5.5 Edit Profile 9](#_Toc86747748)

[5.6 View Product Details 9](#_Toc86747749)

[5.7 Buy Product 10](#_Toc86747750)

[5.8 Add to Favorites 10](#_Toc86747751)

[5.9 Remove from favorites 11](#_Toc86747752)

[5.10 View favorites list 11](#_Toc86747753)

[5.11 Filter Product by Type 12](#_Toc86747754)

[5.12 Filter Product by Size 12](#_Toc86747755)

[5.13 Filter Product by Color 13](#_Toc86747756)

[5.14 Filter Product by Waist 14](#_Toc86747757)

[5.15 Filter Product by Price Range 14](#_Toc86747758)

[5.16 Filter Product by Discount Factor 15](#_Toc86747759)

[5.17 Approve Products 15](#_Toc86747760)

[5.18 Suspend Product 16](#_Toc86747761)

[5.19 View Product 16](#_Toc86747762)

[5.20 View home page 17](#_Toc86747763)

[5.21 Add product 17](#_Toc86747764)

[5.22 Delete a product 18](#_Toc86747765)

[5.23 Update product 18](#_Toc86747766)

[5.24 Give Feedback 19](#_Toc86747767)

[6. Graphical User Interfaces 20](#_Toc86747768)

[6.1 Admin panel 20](#_Toc86747769)

[6.2 Store GUI 21](#_Toc86747770)

[6.3 Login GUI 23](#_Toc86747771)

[6.4 Sign Up GUI 23](#_Toc86747772)

[6.5 User/Client Pages GUI 24](#_Toc86747773)

[7. Database Design 26](#_Toc86747774)

[7.1 ER Diagram 26](#_Toc86747775)

[7.2 Data Dictionary 27](#_Toc86747776)

[7.2.1 User 27](#_Toc86747777)

[7.2.2 Favourite 27](#_Toc86747778)

[7.2.3 Product 28](#_Toc86747779)

[7.2.4 Purchase History 28](#_Toc86747780)

[7.2.5 Store 29](#_Toc86747781)

[7.2.6 Admin 29](#_Toc86747782)

[7.2.7 Discount Factor 29](#_Toc86747783)

[8. Risk Analysis 30](#_Toc86747784)

[9. System Requirements 31](#_Toc86747785)

[9.1 Hardware Requirements 31](#_Toc86747786)

[9.2 Software Requirements 31](#_Toc86747787)

[10. References 32](#_Toc86747788)

List of Figures

[*Figure 1: Level 1 System Architecture 5*](#_Toc86749716)

[*Figure 2: Level 2 System Architecture 6*](#_Toc86749717)

[*Figure 3: Login Screen 20*](#_Toc86749718)

[*Figure 4: Manage Products 20*](#_Toc86749719)

[*Figure 5: Update product page 21*](#_Toc86749720)

[*Figure 6: Store home page 21*](#_Toc86749721)

[*Figure 7: Pop up window 22*](#_Toc86749722)

[*Figure 8: Edit product 22*](#_Toc86749723)

[*Figure 9: Login page 23*](#_Toc86749724)

[*Figure 10: Sign Up page 24*](#_Toc86749725)

[*Figure 11: Favorite’s page 24*](#_Toc86749726)

[*Figure 12: Product Details page 25*](#_Toc86749727)

[*Figure 13: Product Details page 25*](#_Toc86749728)

[*Figure 14: Product Details page 26*](#_Toc86749729)

[*Figure 15: ER Diagram 27*](#_Toc86749730)

# Introduction

The manual inspection of your asset can be tricky and inaccurate at times and when it contains multiple sources then handling each of them at the same time becomes quite impossible. This leads to less maintenance which increases the risk of your asset failure and ultimately leads to costly repairs, which are approximately 30% of the life-cycle costs of a vehicle [1]. With the advancement in technology it is questionable to not have a system that can provide automated timely inspection reports with predictive failure analysis and potentially enable mechanisms to avoid or counteract failure, or recommend maintenance of asset prior to failure.

The purpose of this project is to create an automated system that can handle huge amount of data coming from various Teltonika IOT devices i.e., FMB 920 and FMB 630, and then using Artificial Intelligence and Data Science, to optimize and analyze it in such a way that it can predict precise components failure and avoid the unnecessary inspection and provide with a maintenance schedule that is based on the usual condition of the vehicle in its operating conditions [2].

Our main focus will be on vehicle components live inspection, maintenance and failure. The system will be gathering huge data of different components of a vehicle from installed IOT devices which will contain geographical location, trip data and crash detection etc. to train our predictive analysis model that will eventually help us in detection of failure points. We will also be maintaining a record of all previous maintenance and failures of components of each asset to further enhance our predictions with time.

## Purpose of this Document

The purpose of this document is to give a detailed overview of the project. The document would be working as a guideline for the development team on what is to be actually implemented. Moreover, this would explain all the functional/ non-functional requirements of the project. The document would be providing a detailed description of the use cases and the system architecture. Furthermore, some insight on what the actual system would look like in its final form. The database describes what data would be stored in order to work on the system. Also, the hardware/ software requirements are also described in this document.

## Intended Audience

The intended audience of Erohal are the managers and owners of large organizations. These large organizations have numeric complex assets in abundance which they have to manage, maintain and analyze, and hence have a very tedious and complicated process for it.

## Definitions, Acronyms, and Abbreviations

Erohal: Name of the product

Asset: A valuable item having a complexity that makes it tedious to maintain

Inspection: A detailed observation of the components of an asset.

Issue: Problematic components identified during inspections

Service Request: Asset or its components maintenance and servicing.

Work order: Record of work done for resolving issues.

Preventive Maintenance: Ability to predict maintenance schedules for assets and its components.

Anomaly detection: Ability to automatically identify problematic components and then earlier in time.

# General Description

## User Characteristics

The End Users are classified into two groups namely the Administrator and Vendor/Store owners. The General Users will be publicly everyone. The Administrator will approve/reject products and can edit the products as well. The Store owner will register themselves to our website first. Then they will be able to add, update, view and remove products. The General Users can register themselves by creating a new account. They can buy products on the profile-based recommendations and also by using specific filters provided. They can also maintain their profile and manage brands in their favorites section.

## Domain Overview

Panda Mall aims to ease the tedious task of web surfing and scrolling social media pages just to find the right product for purchase. Studies [1] show that this problem of “choice-overload” can be “detrimental” and lead to indecision, unsatisfactory experience.

Panda Mall website will be Artificially Intelligent (AI) based which uses a content-based Recommendation System (RS) model to filter out a manageable few best option to buy from, for a customer, based on their preferences, purchase history and feedback. Data about brands and products will be obtained from:

1. Websites of brands directly.
2. Registration at Panda Mall.

The results obtained from the model will be presented in a simplified web interface along with all the necessary information for customers to make a buying decision with ease

# 

# Functionality

## Functional Requirements

### Functional requirements for Users

* System will allow user to login by authenticating user login credentials.
* System will allow user to create a new account by providing required information.
* System will allow user to reset password via email.
* System will allow user to logout.
* System will allow user to edit profile.
* System will display clothing products based on user’s profile.
* System will allow user to filter out products based on Type.
* System will allow user to filter out products based on Sizes.
* System will allow user to filter out products based on Colors.
* System will allow user to filter out products based on Waists.
* System will allow user to filter out products based on Price Range.
* System will allow user to filter out products based on Discount factor.
* System will allow user to filter out products based on Brands.
* System will allow user to view selected product details.
* System will allow user to buy product by redirecting to respected page.
* System will allow user to add products to favorites.
* System will allow user to give feedback on the product bought.

### Functional requirements for Administrators

* Administrators can approve/reject products.
* Administrators can update products.
* Administrators can view products.

### Functional requirements for Stores

* Stores can add products.
* Stores can update products.
* Stores can view products.
* Stores can remove products.

## Non-Functional Requirements

### Availability

System will be available for the users for at-least 160 hours a week.

### Usability

System will have an intuitive design.

* Our system interface will be easy to learn and user interactive. A rookie will require at-most 5 seconds to learn the interface.

### Reliability

System will show the result of the desired filter within 3 seconds.

### Scalability

System will ensure there will be no significant performance degrade for at-most 100 users.

### Data Integrity

System will ensure user’s personal data is not tempered by unauthorized source.

### Performance

* All data views are paginated to allow limited and required information to prevent long load times.

## Assumptions

* Users should have basic knowledge of using computer and web browsers.
* Reliable internet connection should be available for the user.

# System Architecture

Following is the diagram representing the system architecture of our project:

Level 1:

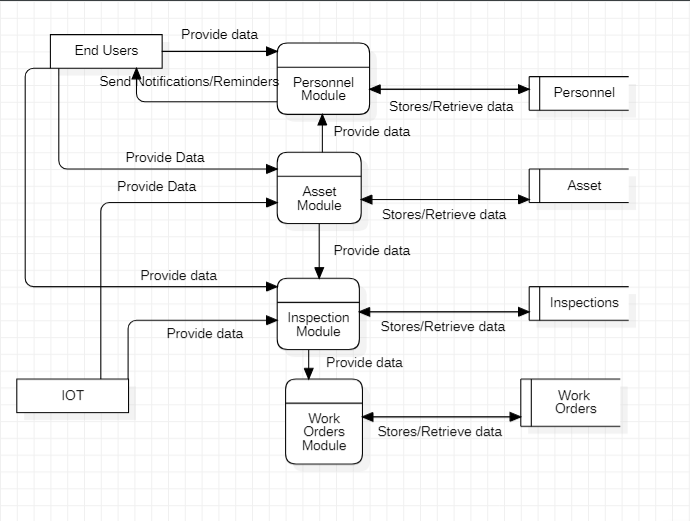


Figure 1: Level 1 System Architecture

*This is the level 1 diagram of system architecture*

Level 2:

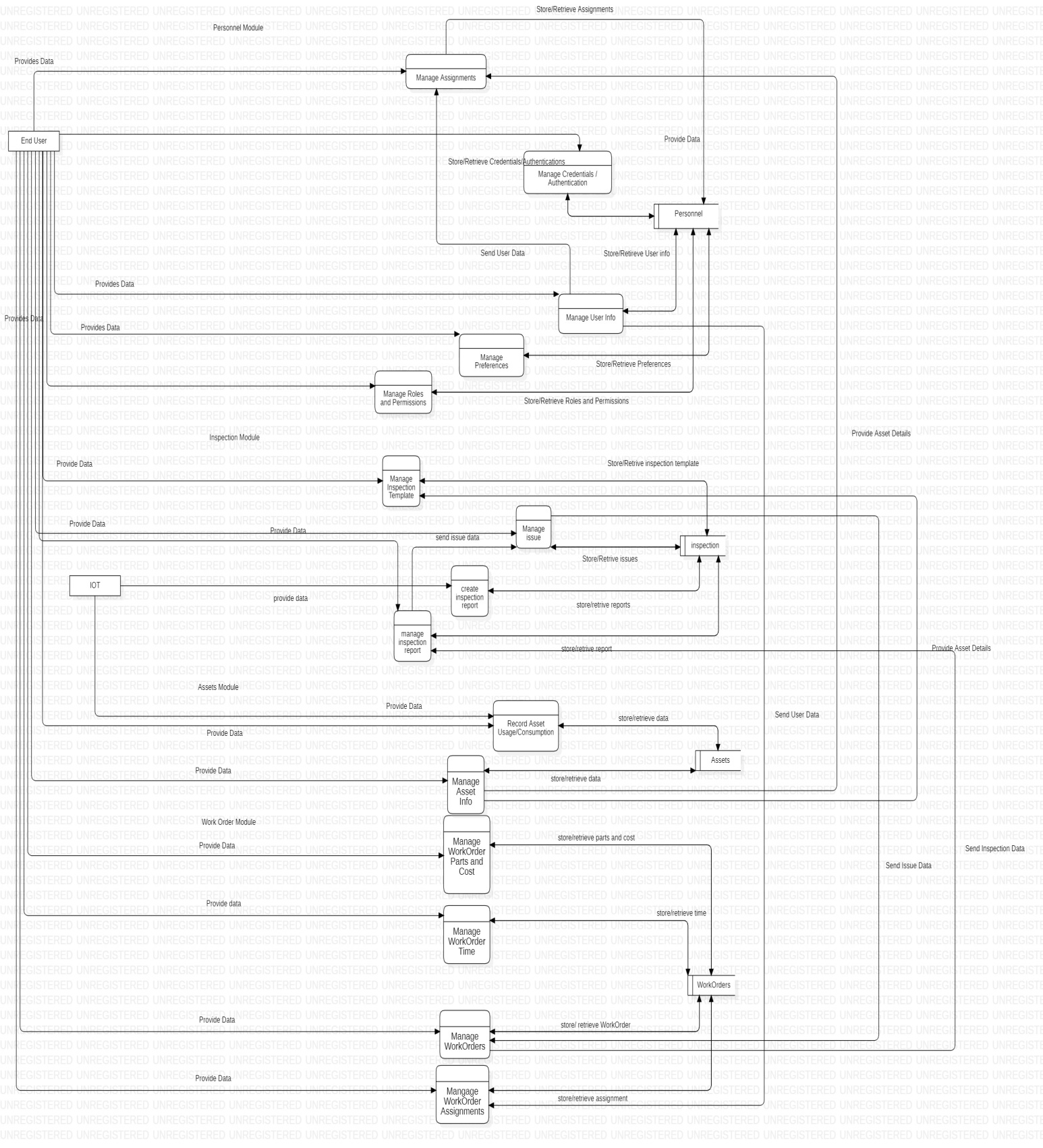


Figure 2: Level 2 System Architecture

*This is the level 2 diagram of system architecture*

# Use Cases

## Login Account

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Login Account | | |
| **Actors** | | Client, Admin, Store Owner | | |
| **Summary** | | User will be able to login into account. | | |
| **Pre-Conditions** | | User should be registered in the system.  User should not already be logged in. | | |
| **Post-Conditions** | | User shall be logged in successfully. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User opens the login page. | | 2 | System displays Login page asking for username and password. |
| 3 | User enters valid login credentials and press login button. | | 4 | System verifies the user credentials, establishes session and redirects the user to the home page. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 3 | User enters invalid login credentials and presses the login button. | | 4-A | System prompts the error message: *Incorrect username or password entered.* |

## Create Account

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Create Account | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to create a new account by providing the required information. | | |
| **Pre-Conditions** | | User clicked on Sign-up button from the login page. | | |
| **Post-Conditions** | | User’s account shall be created successfully. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the Sign-up button from the Login page. | | 2 | System displays Sign-up page asking for required information. |
| 3 | User enters valid information and presses submit button. | | 4 | System verifies the information, creates a new account and redirects the user to the login page. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 3 | User enters invalid information and presses submit button. | | 4-A | System prompts the error message: *Incorrect/Missing required information* |

## Reset Password

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Reset Password | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to reset his/her account password. | | |
| **Pre-Conditions** | | User should have an existing account.  User clicked on “Forget Password?” button from Login page. | | |
| **Post-Conditions** | | User shall receive reset password request on his/her email. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Forget Password?” button from the Login page. | | 2 | System displays a prompt asking for user’s username. |
| 3 | User enters username and presses submit button. | | 4 | System verifies the username, sends an email to the respective user account and redirects the user to the login page. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 3 | User enters invalid username and presses submit button. | | 4-A | System prompts the error message: *Username not found* |

## Logout

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Logout | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to logout his/her account. | | |
| **Pre-Conditions** | | User should have logged in.  User clicked on “Logout” button from Home page. | | |
| **Post-Conditions** | | User shall be logged out. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Logout?” button from the home page. | | 2 | System will logout the user and redirect the user to Login page. |
| **No Alternative Flow** | | | | |

## Edit Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Edit Profile | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to edit his/her profile. | | |
| **Pre-Conditions** | | User should have logged in.  User clicked on “Edit Profile” button from Home page. | | |
| **Post-Conditions** | | User’s profile shall be updated. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Edit Profile” button from the Home page. | | 2 | System displays a new page containing user’s current information. |
| 3 | User edits his/her profile by providing valid information and presses update button. | | 4 | System validates the information, updates his/her profile and redirects the user to the Home page. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 3 | User enters invalid profile information and presses update button. | | 4-A | System prompts the error message: *Information provided is incorrect.* |

## View Product Details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View Product Details | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to view the details of the selected product. | | |
| **Pre-Conditions** | | User should have logged in.  User should have clicked on a “Details” button. | | |
| **Post-Conditions** | | Users shall be to see the product details. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Details” button from the Home page. | | 2 | System displays a pop-up containing product details. |
| **No Alternative Flow** | | | | |

## Buy Product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Buy a Product | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to buy a product from the respected brand’s website. | | |
| **Pre-Conditions** | | User must be logged in.  User must have clicked on “Detail” button for the selected product. | | |
| **Post-Conditions** | | System will redirect user to the respective brand website. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Buy” button located in the product details. | | 2 | System redirects the user to the respected link of the product. |
| **No Alternative Flow** | | | | |

## Add to Favorites

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Add to favorites | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to add a product to his/her favorites list. | | |
| **Pre-Conditions** | | User must be logged in.  Selected product should not be in the favorites list | | |
| **Post-Conditions** | | System will update the favorites list by adding a product. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Favorites” icon from the selected product. | | 2 | System adds the product to his/her favorites list. |
| **No Alternative Flow** | | | | |

## Remove from favorites

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Remove from favorites | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to remove a product from his/her favorites list. | | |
| **Pre-Conditions** | | User must be logged in.  Selected product should be in the favorites list | | |
| **Post-Conditions** | | System will update the favorites list by removing a product. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Favorites” icon from the selected product. | | 2 | System will remove the product from his/her favorites list. |
| **No Alternative Flow** | | | | |

## View favorites list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View favorites list | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to view favorites list. | | |
| **Pre-Conditions** | | User must be logged in. | | |
| **Post-Conditions** | | User shall be able to view his/her favorites list. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “View favorites” button from the home page. | | 2 | System redirects the user to the new page containing a list of user’s favorite items. If there are no products in the favorites list then an empty list will be displayed. |
| **No Alternative Flow** | | | | |

## Filter Product by Type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Filter product by type | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to filter out products based on product type. | | |
| **Pre-Conditions** | | User must be logged in. | | |
| **Post-Conditions** | | User will be able to view filtered product list. | | |
| **Special Requirements** | | Reliability (filtering takes less than 3 seconds). | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Filter” button from the home page. | | 2 | System displays a dropdown menu containing different types of filters. |
| 3 | User selects the “Type” filter from the dropdown menu. | | 4 | System displays a pop-up of types for clothing products e.g. (Men’s T-Shirts, Men’s Polo Shirts, Men’s Trousers). |
| 5 | User selects a type from the pop-up window. | | 6 | System reloads the home page and displays the products based on the user’s filtering criteria. |
| **No Alternative Flow** | | | | |

## Filter Product by Size

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Filter product by size | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to filter out products based on product sizes. | | |
| **Pre-Conditions** | | User must be logged in. | | |
| **Post-Conditions** | | User will be able to view filtered product list. | | |
| **Special Requirements** | | Reliability (filtering takes less than 3 seconds). | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Filter” button from the home page. | | 2 | System displays a dropdown menu containing different types of filters. |
| 3 | User selects the “Size” filter from the dropdown menu. | | 4 | System displays a pop-up of sizes for clothing products e.g. (Small, Large). |
| 5 | User selects a size from the pop-up window. | | 6 | System reloads the home page and displays the products based on the user’s filtering criteria. |
| **No Alternative Flow** | | | | |

## Filter Product by Color

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Filter product by color | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to filter out products based on product color. | | |
| **Pre-Conditions** | | User must be logged in. | | |
| **Post-Conditions** | | User will be able to view filtered product list. | | |
| **Special Requirements** | | Reliability (filtering takes less than 3 seconds). | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Filter” button from the home page. | | 2 | System displays a dropdown menu containing different types of filters. |
| 3 | User selects the “Color” filter from the dropdown menu. | | 4 | System displays a pop-up of colors for clothing products e.g. (Red, Green). |
| 5 | User selects a color from the pop-up window. | | 6 | System reloads the home page and displays the products based on the user’s filtering criteria. |
| **No Alternative Flow** | | | | |

## Filter Product by Waist

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Filter product by waist | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to filter out products based on product waist. | | |
| **Pre-Conditions** | | User must be logged in. | | |
| **Post-Conditions** | | User will be able to view filtered product list. | | |
| **Special Requirements** | | Reliability (filtering takes less than 3 seconds). | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Filter” button from the home page. | | 2 | System displays a dropdown menu containing different types of filters. |
| 3 | User selects the “Waist” filter from the dropdown menu. | | 4 | System displays a pop-up of waists for clothing products e.g. (30-inch, 32-inch). |
| 5 | User selects a waist from the pop-up window. | | 6 | System reloads the home page and displays the products based on the user’s filtering criteria. |
| **No Alternative Flow** | | | | |

## Filter Product by Price Range

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Filter product by price range. | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to filter out products based on product price range. | | |
| **Pre-Conditions** | | User must be logged in. | | |
| **Post-Conditions** | | User will be able to view filtered product list. | | |
| **Special Requirements** | | Reliability (filtering takes less than 3 seconds). | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Filter” button from the home page. | | 2 | System displays a dropdown menu containing different types of filters. |
| 3 | User selects the “Price” filter from the dropdown menu. | | 4 | System displays a pop-up of price range for clothing products e.g. (Rs. 1200-1800, Rs. 750-1000). |
| 5 | User selects a price range from the pop-up window. | | 6 | System reloads the home page and displays the products based on the user’s filtering criteria. |
| **No Alternative Flow** | | | | |

## Filter Product by Discount Factor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Filter product by discount factor | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to filter out products based on product discount factor. | | |
| **Pre-Conditions** | | User must be logged in. | | |
| **Post-Conditions** | | User will be able to view filtered product list. | | |
| **Special Requirements** | | Reliability (filtering takes less than 3 seconds). | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Filter” button from the home page. | | 2 | System displays a dropdown menu containing different types of filters. |
| 3 | User selects the “Discounted” filter from the dropdown menu. | | 4 | System displays a pop-up of Discount range for clothing products e.g. (20% - 30% off). |
| 5 | User selects a Discount range from the pop-up window. | | 6 | System reloads the home page and displays the products based on the user’s filtering criteria. |
| **No Alternative Flow** | | | | |

## Approve Products

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Approve product | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will be able to see the details of a product and approve it to be published on the website | | |
| **Pre-Conditions** | | Admin should be logged in the system and also on products list view page in the manage products section | | |
| **Post-Conditions** | | Product is approved or rejected. | | |
| **Special Requirements** | | There should be an approval request pending. | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Admin click on a store in manage products section. | | 2 | Redirect the admin to a page showing all products of the store. Pending approval ordered on top. |
| 3 | Click on a particular product | | 4 | Display all the attributes of the product. |
| 5 | Admin click on approve button at the bottom | | 6 | System will approve the product and publish on the website and notify the store. |
| 7 | Admin click exit button. | | 8 | Display list of products. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 5 | Admin click on reject button alongside the approve button. | | 6-A | System will not publish the product on the website and notify the store. |

## Suspend Product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Suspend product | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will be able to view and terminate a published product. | | |
| **Pre-Conditions** | | Admin should be logged in the system and also on products list view page in the manage products section. | | |
| **Post-Conditions** | | System will immediately suspend the product from the website. | | |
| **Special Requirements** | | Product must already be published | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Admin click on a store in manage products section. | | 2 | Redirect the admin to a page showing all products of the store. Pending approval ordered on top. |
| 3 | Click on a particular product | | 4 | Display all the attributes of the product. |
| 5 | Click on “terminate” button at the bottom. | | 6 | System suspends the product and exit product view page. |
| **No Alternative Flow** | | | | |

## View Product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View products | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will be able to see the attribute of a product | | |
| **Pre-Conditions** | | Admin should be logged in the system and also on manage products section. | | |
| **Post-Conditions** | | System will display the attributes and values of the product. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Admin click on store entry in the table ordered based on approval requests. | | 2 | System will display the list of products sold by the store. |
| 3 | Admin click on a particular product from the table. | | 4 | System will display the attributes and values of the product. |
| **No Alternative Flow** | | | | |

## View home page

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View home page | | |
| **Actors** | | Stores | | |
| **Summary** | | User will be able to see all the products of their store. | | |
| **Pre-Conditions** | | User should be logged in. | | |
| **Post-Conditions** | | User successfully view all of their products | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
|  | **Actor Action** | | **System Response** | |
| 1 | Store user logins successfully | | 2 | System loads the home page with a list view of all products |
| **Alternative Flow** | | | | |
| 1 | User clicks on the home from the navigation bar. | | 2-A | System loads the home page with a list view of all products |

## Add product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Add product | | |
| **Actors** | | Stores | | |
| **Summary** | | User will be able to add products to their stores. | | |
| **Pre-Conditions** | | User should be on logged in. | | |
| **Post-Conditions** | | User added product to store successfully. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on Add product. | | 2 | Create new product page is displayed, showing product detail, product description, product category, gallery and attachments needed to be filled. |
| 3 | User enters the details and press add button. | | 4 | System checks the necessary fields needed to be filled and create that product and set “pending approval” for the product.  System prompts the success message: *Product sent for approval.* Redirects the user back to home page. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 3 | User miss important fields and press save. | | 4-A | System highlighted the required fields in red and prompts the error message:  *Please fill required fields!* |

## Delete a product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Delete a product | | |
| **Actors** | | Store | | |
| **Summary** | | User will be able to delete a product from store. | | |
| **Pre-Conditions** | | User should be on logged in. | | |
| **Post-Conditions** | | User deleted a product successfully. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User select a product(s) from the home page. | | 2 | System records the selections. |
| 3 | User clicks on the delete button. | | 4 | System pops a small window with a warning. |
| 5 | User selects, “I’m sure” button. | | 6 | System deletes the selected products and prompts the successful message: *Products removed Successfully*. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 5 | User clicks “No” button. | | 6-A | System returns to the home page. |

## Update product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Update product | | |
| **Actors** | | Store | | |
| **Summary** | | User will be able to update the attribute values of a product from store like price or availability. | | |
| **Pre-Conditions** | | User should be on logged in. | | |
| **Post-Conditions** | | User updated a product attribute value successfully. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User select a product from the home page. | | 2 | System records the selection. |
| 3 | User clicks on the update button. | | 4 | System displays an editable product form with values pre-filled. |
| 5 | User enters the details and press update button. | | 6 | System checks the necessary fields needed to be filled and create that product and set “pending approval” for the product.  System prompts the success message: *Product sent for approval.* Redirects the user back to home page. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 5 | User miss important fields and press save. | | 6-A | System highlighted the required fields in red and prompts the error message:  *Please fill required fields!* |
| 5 | User clicks ‘cancel’ | | 6-A | Redirects the user back to home page. |

## Give Feedback

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Give feedback | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to give feedback on products bought. | | |
| **Pre-Conditions** | | User should be logged in.  User should have bought the product.  User should be on the product details page. | | |
| **Post-Conditions** | | User’s feedback will be stored. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User will give feedback for the bought product by using star ratings. | | 2 | System will store user’s feedback in the database. |
| **No Alternative Flow** | | | | |

# Graphical User Interfaces

## Admin panel

This GUI shows login page for an admin.

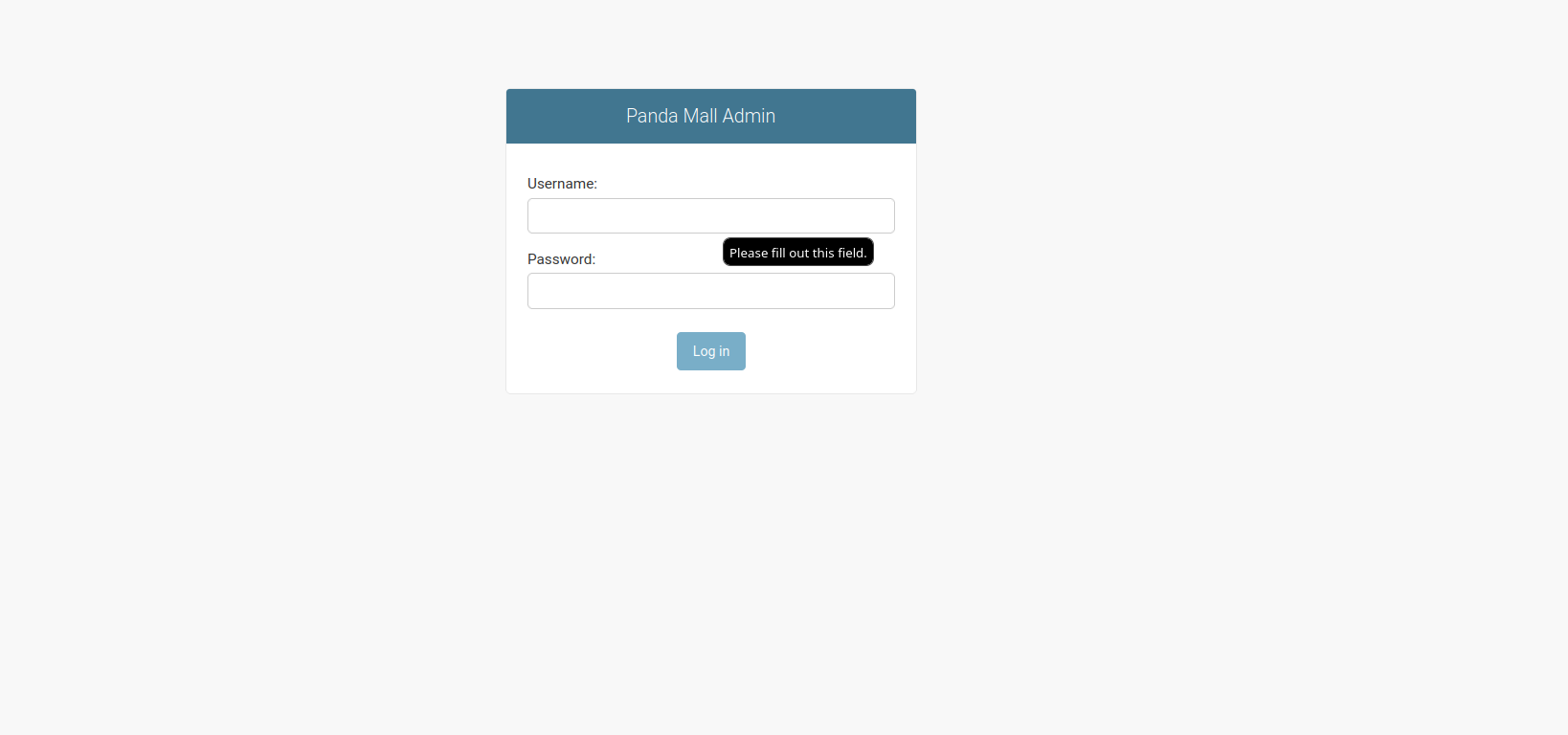


Figure 3: Login Screen

Admin panel login screen.

This GUI shows the “Manage Products” section of the admin panel.

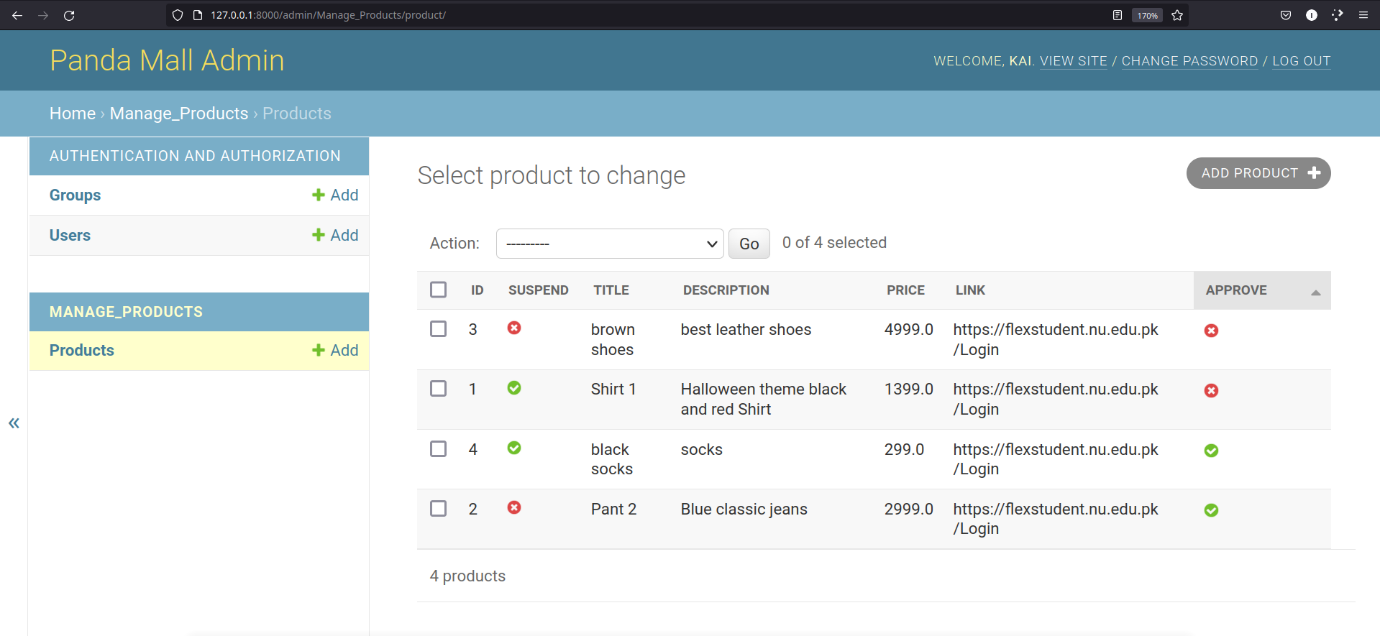


Figure 4: Manage Products

Manage Products Section

This GUI shows the edit page where an admin can approve, reject or suspend a product.

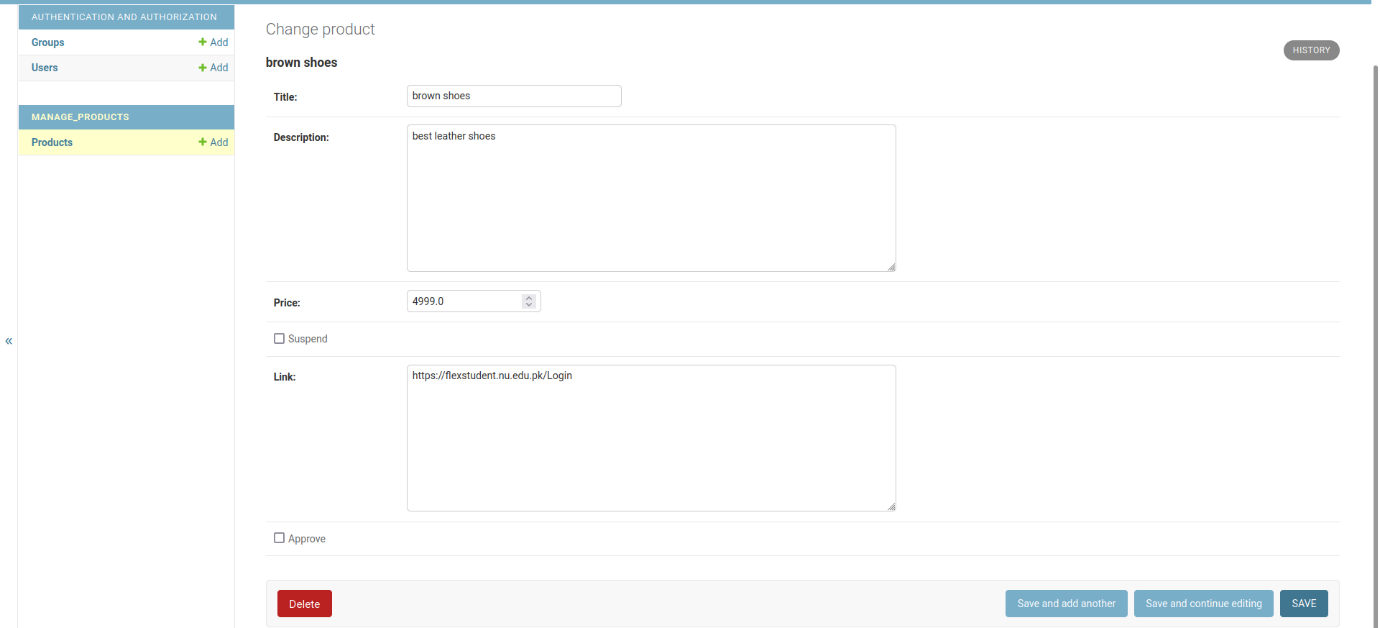


Figure 5: Update product page

Update product page at admin panel

## Store GUI

This GUI shows the home page of a store.

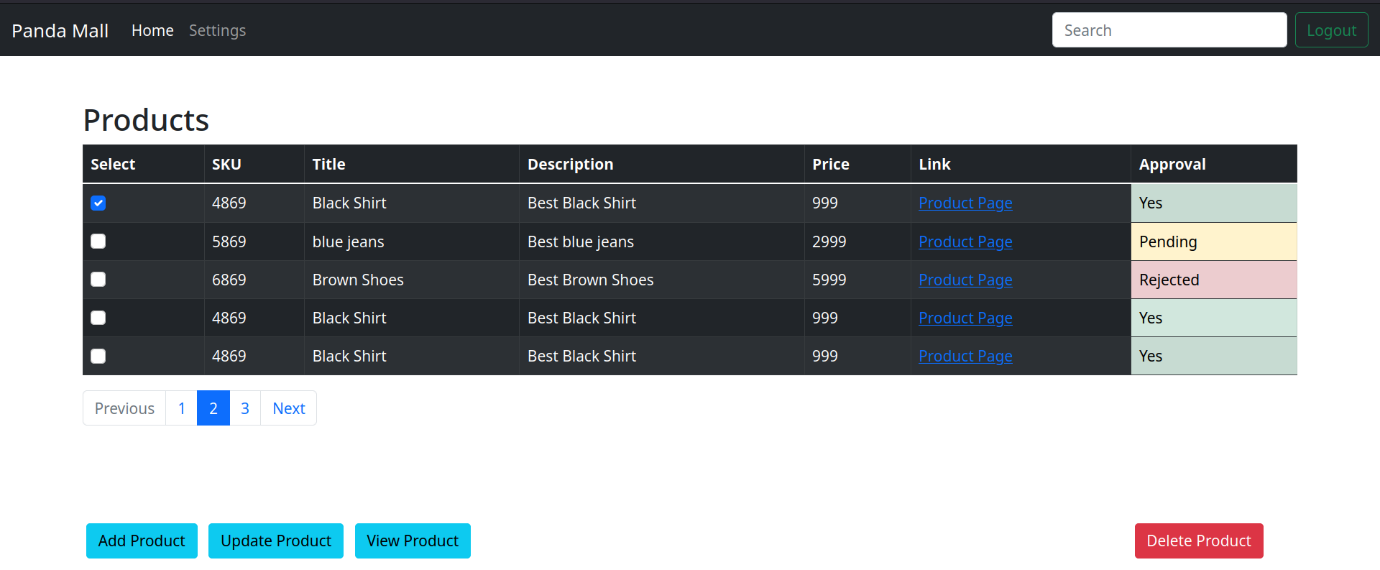


Figure 6: Store home page

Store home page

This GUI shows the delete pop up window before a product is deleted.

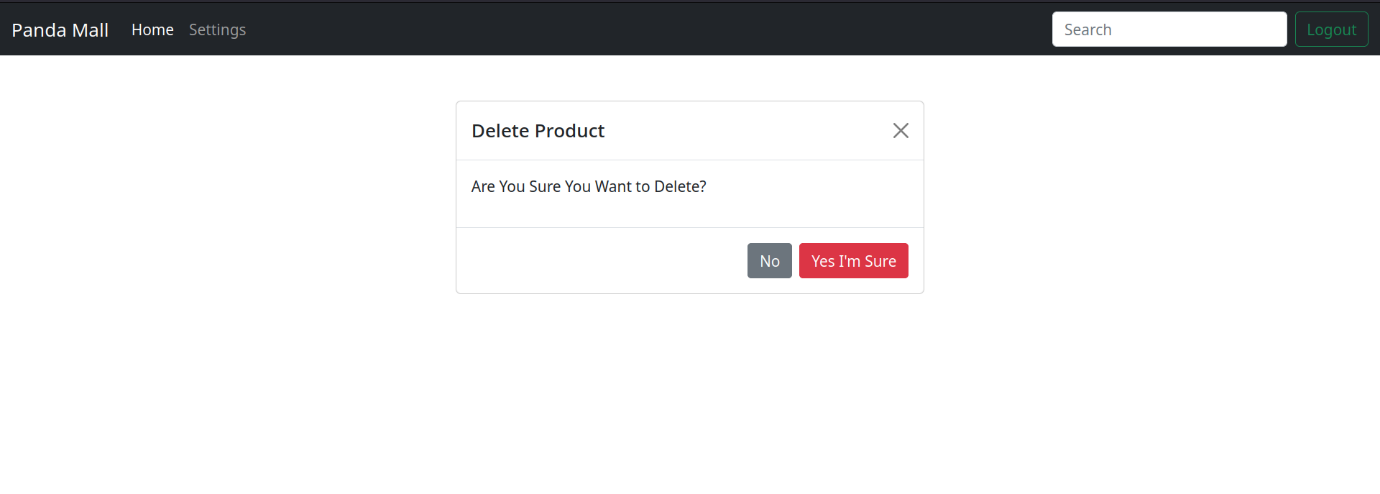


Figure 7: Pop up window

Pop up window when delete button clicked

This GUI shows the details of a product and allows to edit the product. This GUI is same for add products and update as the latter has prefilled values.

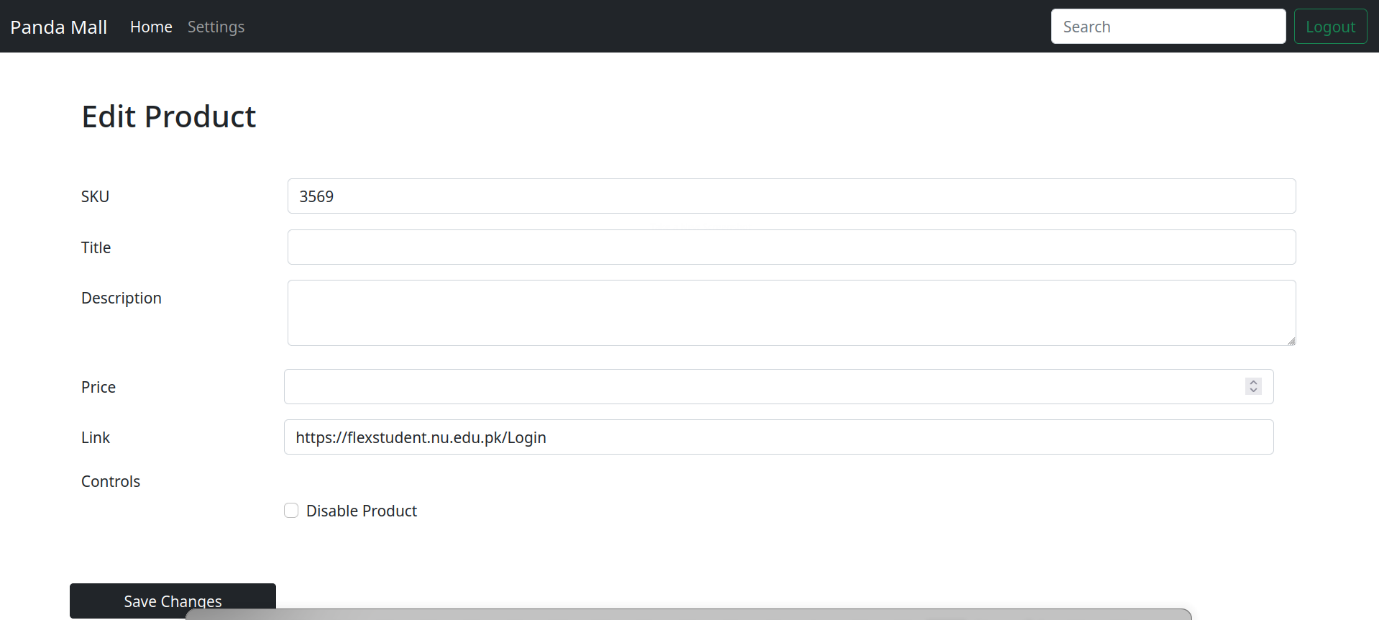


Figure 8: Edit product

Edit product page for store type user

## Login GUI

This GUI shows the login page of our website.

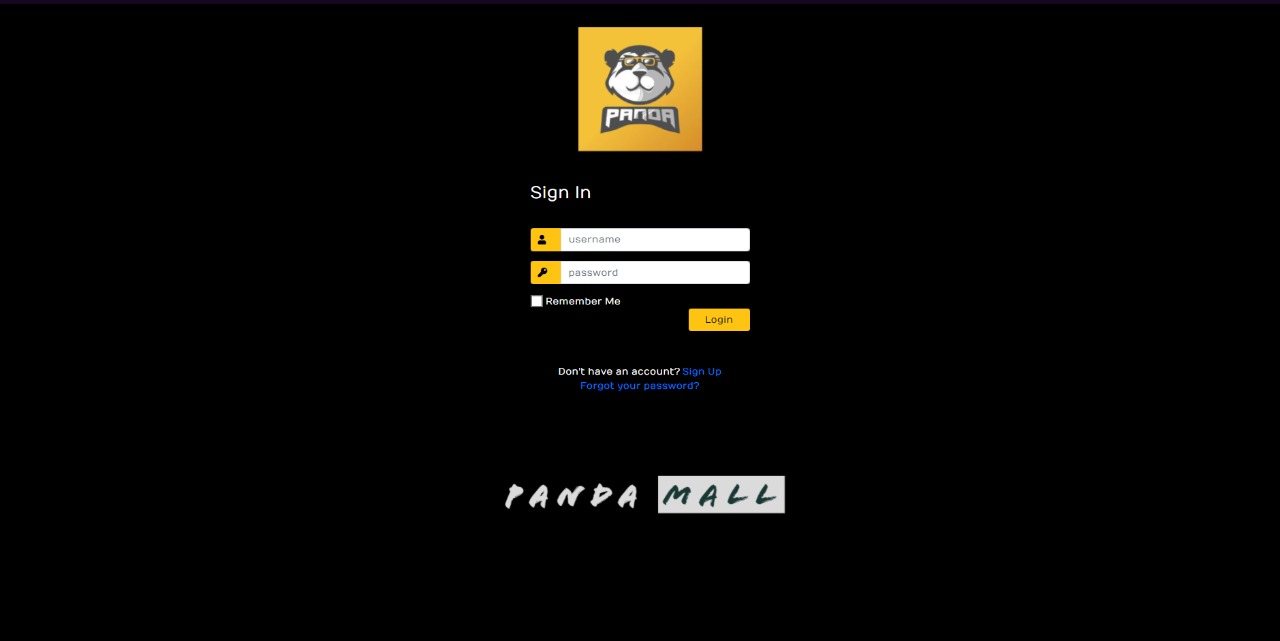


Figure 9: Login page

Website’s login page

## Sign Up GUI

This GUI shows the Sign-Up page of our website.

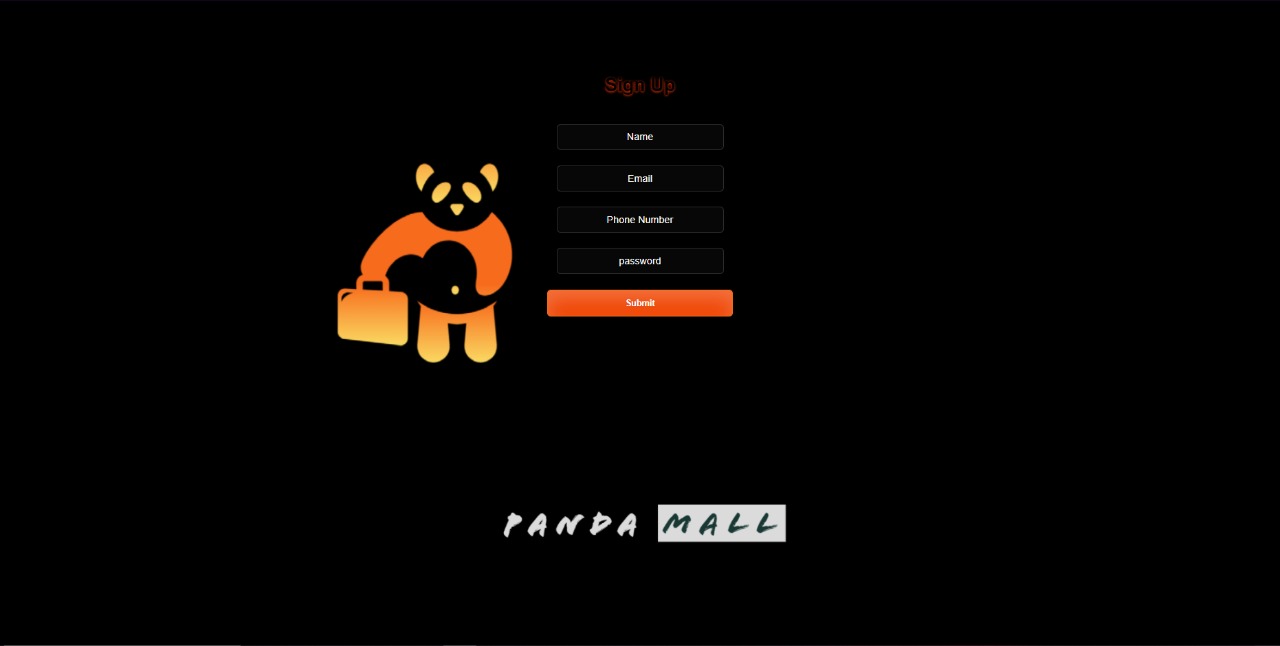


Figure 10: Sign Up page

Website’s sign-up page

## User/Client Pages GUI

This GUI shows the favourites page of our website. All of user’s favourites products are placed on this page.

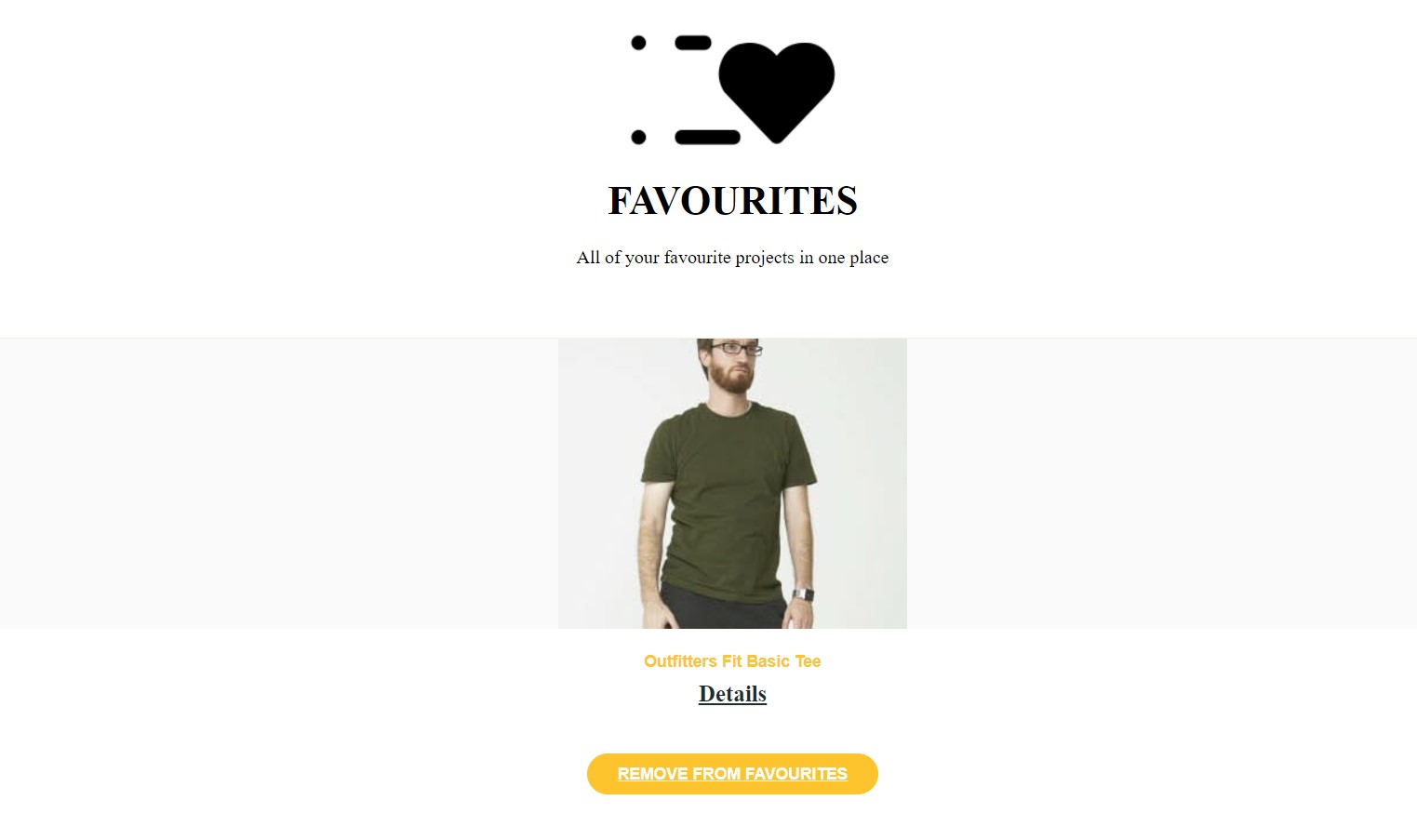


Figure 11: Favorite’s page

Website’s favourites page

This GUI shows the product details pop up window of our website. From this page we can buy a product along with viewing its details. Moreover, we can give feedback of a product that we have bought.



Figure 12: Product Details page

Products Detail page

This GUI shows the Home page for the user where user can see products to buy from. User can add product to favorite’s list, can see the details of the product. Also, user can use filters to filter out specific product. User can also search a specific product and also can see his profile.

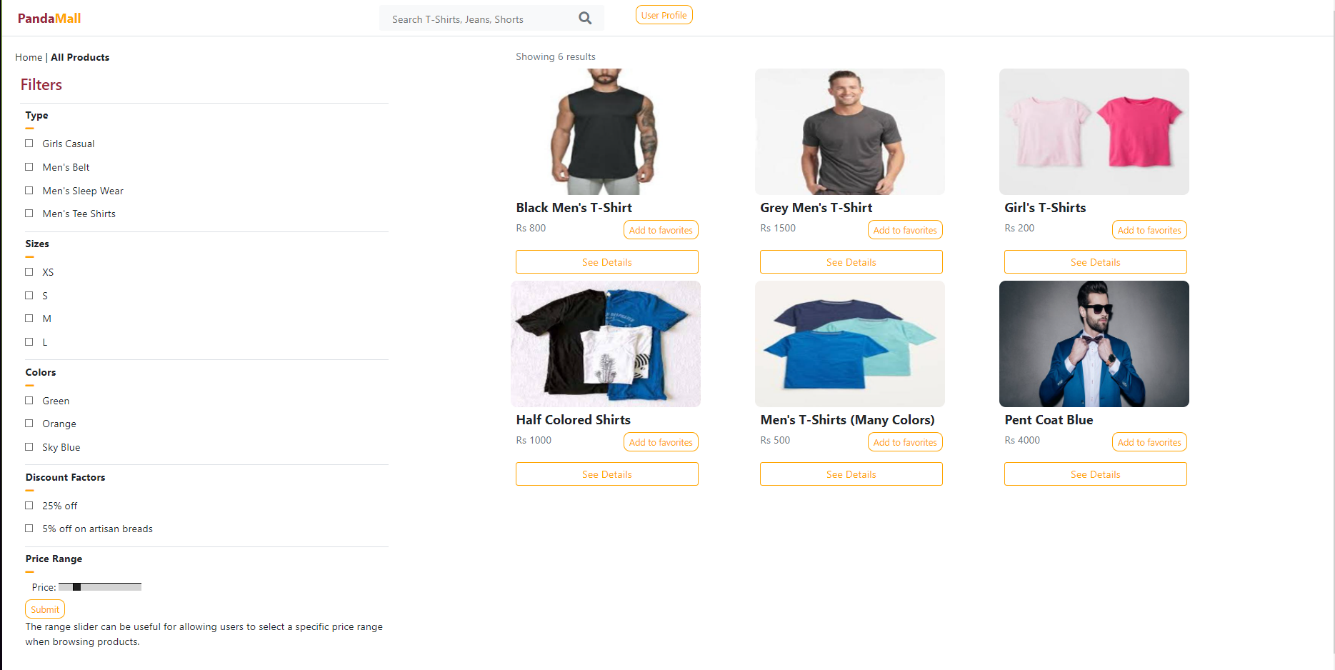


Figure 13: Home Page

Home page containing products

This GUI shows the User’s profile where he can see his information. User can access feedback page and favorite’s page from here as well. Also user can sign out from here as well.

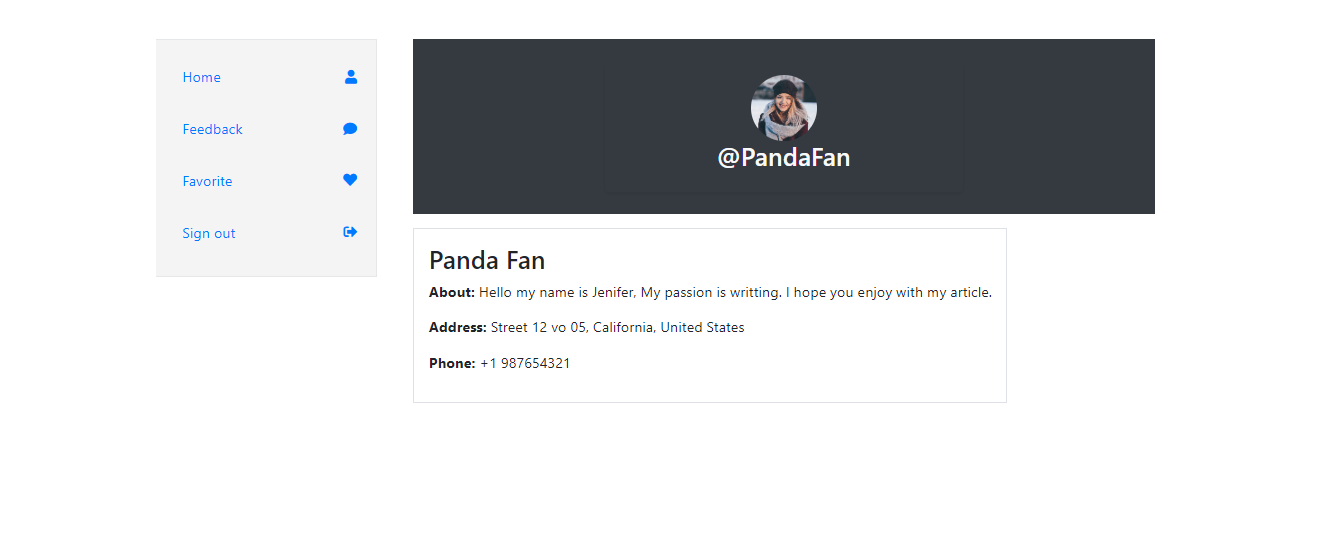


Figure 14: Profile page

Profile containing user information

# Database Design

## ER Diagram

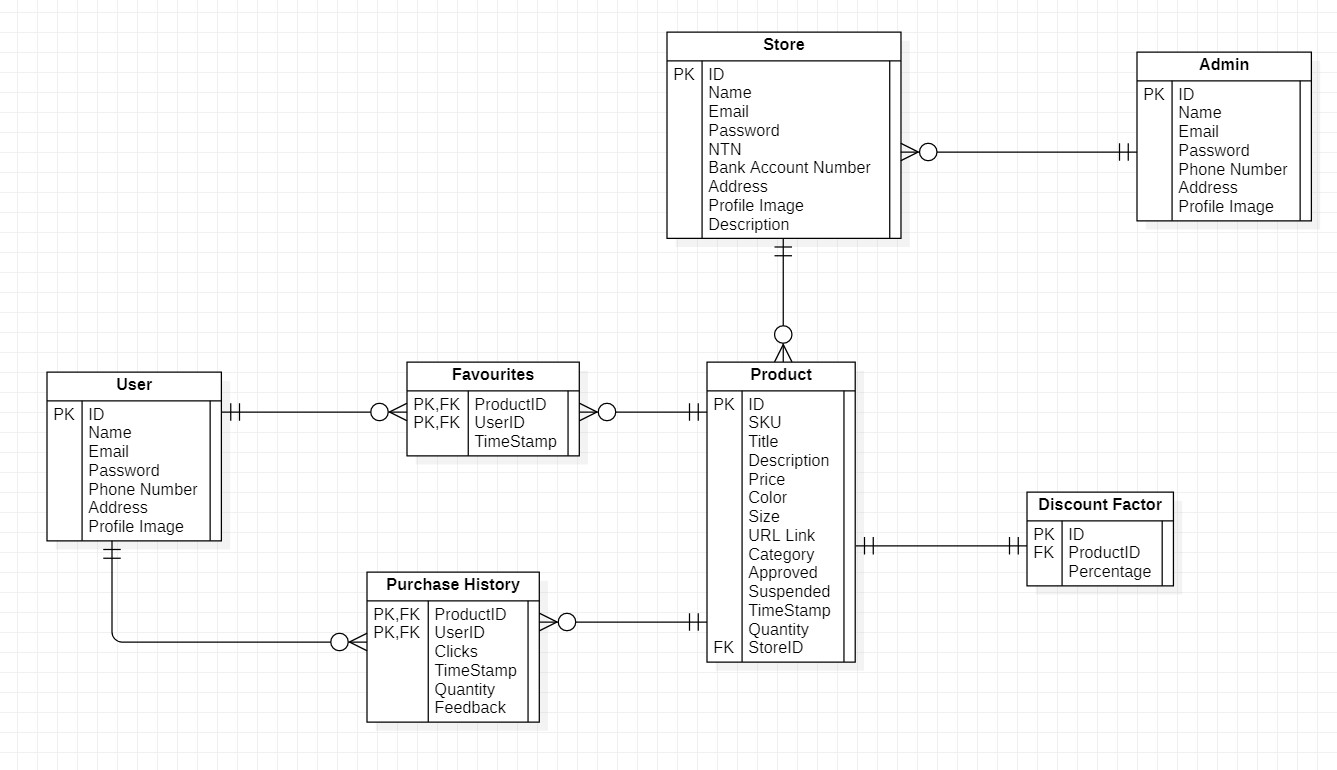


Figure 15: ER Diagram

This shows the ER Diagram of our system database

## Data Dictionary

### User

Table 55: User Data Dictionary

This is the data dictionary of user.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Types** | **Example** |
| ID | Long | 1992 |
| Name | String | Ben White |
| Email | String | [benwhite@gmail.com](mailto:benwhite@gmail.com) |
| Password | String | ArsenalFC |
| Phone Number | String | 0321-xxxxxxx |
| Address | String | 852-B Milaad St, Block B Faisal Town, Lahore, Punjab 54770 |
| Profile Image | String | myWebsite.com/proiflepic.jpg |

### Favourite

Table 56: Favourites Data Dictionary

This is the data dictionary of favourites.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Types** | **Examples** |
| ProductID  (FK from ID of Product table) | Long | 221 |
| UserID  (FK from ID of Product table) | Long | 1919 |
| TimeStamp | datetime | 2016-12-21 00:00:00.000 |

### Product

Table 57: Product Data Dictionary

This is the data dictionary of Product.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Types** | **Examples** |
| ID | Long | 23 |
| SKU | String | K91919 |
| Title | String | Shorts |
| Description | String | Any description |
| Price | Float | 99.99 |
| Color | String | Red |
| Size | String | [Medium](https://somename.com/) |
| URL | String | My-website.com/product=10 |
| Category | String | Men |
| Approved | Boolean | True |
| Suspended | Boolean | False |
| TimeStamp | DateTime | 2016-12-21 00:00:00.000 |
| Quantity | Int | 10 |
| StoreID (FK) | Long | 101 |

### Purchase History

Table 58: Purchase History Data Dictionary

This is the data dictionary of Purchase History.

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Types** | **Examples** |
| ProductID (PK, FK) | Long | 23 |
| UserID (PK, FK) | Long | 12 |
| Clicks | Int | 20 |
| TimeStamp | DateTime | 2016-12-21 00:00:00.000 |
| Quantity | Int | 15 |
| Feedback | String | Great Product |

### Store

Table 59: Store Data Dictionary

This is the data dictionary of Store.

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Types** | **Examples** |
| ID | Long | 7 |
| Name | String | Tayyab |
| Email | String | [tayyab@gmail.com](mailto:tayyab@gmail.com) |
| Password | String | Qwerty1234 |
| NTN | String | 231242134-9 |
| Bank Account Number | String | HQM21HJ819 |
| Address | String | 134 A Muslim Town |
| Profile Image | String | myWebsite.com/proiflepic.jpg |
| Description | String | Detailed Info |

### Admin

Table 60: Admin Data Dictionary

This is the data dictionary of Admin.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Types** | **Examples** |
| ID | Long | 23 |
| Name | String | Tayyab |
| Email | String | [tayyab@gmail.com](mailto:tayyab@gmail.com) |
| Password | String | Tayyabqwerty |
| Phone Number | String | 03401913211 |
| Profile Image | String | myWebsite.com/proiflepic.jpg |

### Discount Factor

Table 61 Discount Factor Data Dictionary

This is the data dictionary of Discount.

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Types** | **Examples** |
| ID | Long | 3 |
| ProductID (FK) | Long | 29 |
| Percentage | Float | 70.1 |

# 

# Risk Analysis

A recommendation system using filtering algorithm can be exposed to several problems with some risks attached to them.

* Overflow of client requests on the server side can result in server being unable to handle the client load. As result client will face delays in his/her requests.
* Server could crash due to a network or software issue which will result in clients not getting a response from server.
* Server can be at the end of a natural disaster that can result in server’s hardware fault.
* There could be an internet disconnectivity issue where the server is located that can result in server not responding to client requests.
* There could be DDoS (Distributed Denial of Service) attack at server side that can again result in server not responding to client requests.
* Our website can be a victim of “Name Jacking” that can result in our client traffic being diverted from our server.
* “SQL Injection” can also be a possible threat to our database as it can result in our valuable information being stolen from our database.

# System Requirements

## Hardware Requirements

* A IOT device that supports a browser.
* Internet connection should have at least 1Mbps.
* A web server to host our website.

## Software Requirements

* Web browser in the system.

# References

1. Reutskaja, E., Lindner, A., Nagel, R. et al. “Choice overload reduces neural signatures of choice set value in dorsal striatum and anterior cingulate cortex.” *Nature Human Behavior,* vol 2, no. 925–935, Oct, 2018. [Online serial]. Available: <https://doi.org/10.1038/s41562-018-0440-2>[Accessed Sept. 30, 2021]