

Q-Shopping App

Introduction

Since the pandemic, many stores in Qatar have closed due to a lack of customers. As a result, many stores try different ways to reach out to their customers.

This android app project aims to develop an online shopping store that can be used by store's owners to list their business online and will also allow customers to buy the products directly from the store.

The app is based on the android platform. Customers can buy the products directly from the store. This will help local businesses to register their products on the app. It will eventually help local businesses to increase their profits and people to buy the goods from the comfort of their homes.

Users will be able to see the products from the mobile app. In addition, the application will contain a home screen where all the basic information about the store and details of the products will be available. This app will be easy to operate and user-friendly.

OBJECTIVE OF THE PROJECT

The main aim of this project is to design, develop, and implement an online shopping application based on Android. This app will allow local stores, small business owners and start-ups to start their businesses and reach out to their customers.

The clients will also not need to go to physical shops to purchase items as they will be able to do it through the applications. This will save the client's transportation money and time.

Benefits of the proposed work

- It saves time for customers in quickly view various items on the application
- The ability to view and purchase items anytime, anywhere, with Internet access.
- Provides information about products.
- User-friendly interface.
- Easy to use and simple to understand.
- Quick and saves lots of time for the user.

Use Cases and description

| User Cases | Description |
|-------------------|--|
| U1 - Register | Allow users to register to use the application |
| U2 - Login | Allow users to login. The login is a prerequisite for all other use cases except the registration. Therefore, only logged-in users can use the App. |
| U3 - View Product | <p>The view product allows users to</p> <ul style="list-style-type: none">• View the list of available products• Search products• Add product to cart• View product details• Filter products <p>Alternative [Admin User] If the user is logged in as admin, they should get extra features in addition to the above features, which is the ability to</p> <ul style="list-style-type: none">• Add product• Update Product• Delete Product |
| U4 – Cart | <p>The user should be able to</p> <ul style="list-style-type: none">• Add a product to the cart• Remove a product from the cart• Place order of the products inside the cart |
| U5 - Place Order | <p>Customers should be able to place an order for the products in the cart. The system should</p> <ul style="list-style-type: none">• Show the products in the cart• Products unit price and quantity• The total cost for all the products in the cart |

Fall 2022 - CMPS 312 – Mobile App Development Course Project

| | |
|-----------------------------|--|
| | <ul style="list-style-type: none">• Choice of delivery address• A way to place the order |
| U6 - Post Review | The customer should be able to <ul style="list-style-type: none">• Rate product• Post reviews of the product• Edit and delete their review |
| U7 - Track Order | The customer should be able to track the product delivery status. The status of the product is either <ul style="list-style-type: none">• Processing, if the product is in the store• Shipped, if it left the store• Delivered, if the user received the product |
| U8 - Admin Dashboard | The admin should be able to <ul style="list-style-type: none">• View all the placed orders• Sort and filter orders by status• Update the order status |
| U9 - Admin Reports | <p>Get the products for a period and by status. For example, the admin user can enter the from-date and to-date and select product status: <i>All</i> (this is the default value), <i>Processing</i>, <i>Shipped</i> or <i>Delivered</i>.</p> <p>Upon submission, the app should display the product details. In addition, the total amount and count of products should be displayed.</p> <p>In case the status is set to <i>All</i>, then the totals and counts should be provided for <i>Processing</i>, <i>Shipped</i>, or <i>Delivered</i> and <i>Grand Total</i></p> |

Deliverables

Seek further clarification about the requirements/deliverables during the initial progress meeting with the instructors.

Note that further important **clarifications** maybe **modified/added** to the project requirements. So, always follow up with the GitHub project folder.

1. Application design document that includes the UI design and the Class Diagrams for Entities and Repositories and View Models. Remember that 'there is elegance in simplicity'! Start by designing the UI wireframe (sketch). Decide the UI components and the layout either on paper or use a design tool such as [Balsamiq](#), [Adobe XD](#), [Canva](#), [Figma](#) or you can use [paper-prototype](#).
Note: During the weekly project meetings with the instructor, you are required to present and discuss your design with the instructor and get feedback. You should only start the implementation after addressing the feedback received about your design.

Fall 2022 - CMPS 312 – Mobile App Development Course Project

2. Implement UI for each use case following best design practices. The UI should be fully working using data loaded from json files. You can also download the original data from <https://fakestoreapi.com/docs>
3. Design and implement the app navigation to achieve the app use cases. It should be fully working, and the user can navigate from one activity to another in intuitive and user-friendly way. Implement the **entities** and **repositories** using Kotlin. They should be fully working. Create some test data to ease testing. First, test them using a main function that displays the results to the console before using them in the UI.
4. Document the testing of UI and repositories using screen shots illustrating the testing results.
5. Every team member should submit a description of their project contribution. Every team member should demo their work and answer questions during the demo.

Push your implementation and documentation to your group GitHub repository as you make progress.

Due dates

Phase 1 (UC1-UC6)

Submission: Saturday, October 8, 2022.

Demo presentation and Q&A: Class time Sunday and Tuesday, October 9 and 11, 2022.

Alter project code: Lab time Tuesday, Wednesday, and Thursday, October 11, 12 and 13, 2022.

Phase 2 (All UCs)

Submission: Saturday, November 12, 2022.

Demo presentation and Q&A: Class time Sunday and Tuesday, November 13 and 15, 2022.

Alter project code: Lab time Tuesday, Wednesday, and Thursday, November 15, 16 and 17, 2022.