

**Author**: Jagjeet Reehal

**Project**: Media Oasis Commercial Mobile App

**GitHub Repository**: <https://github.com/jagree03/media-oasis-mobile>

**Description:**

Media Oasis is a personal project that builds a stand-alone Android-based mobile app acting as an expansion to the fictional “Oasis” brand, built on top of my previous Café Oasis project from 2023/2024 that aimed to be a coffee shop mobile application for Android that sells different types of coffee, biscuits & drinks. Media Oasis is essentially a reformed & enhanced version of Café Oasis that is now selling entertainment products such as Movies, Video Games, DVD’s and other visual media in physical form. Media Oasis aims to be an improvement over Café Oasis and will fix problems and missing content from the predecessor.

# Analysis and design

1. **Functional requirements**

*Outlining functional requirements that the application must meet from the varying components forming the entire application.*

*Media Oasis uses the system design of designating 2 roles, the Customer – also known as the client and the Administrator who is responsible for controlling the back-end of the system.*

|  |  |
| --- | --- |
| **What the system should do** | **Requirement** |
| CUSTOMER – Allow clients to register and login to accounts | The system should allow the client to register an account onto the system by allowing them to input their details, as the account information is stored in the database therefore, they should be able to login to that account when they open the application. |
| CUSTOMER – Allow clients to browse items in the catalogue | After a successful login, the system should allow the client to look through a selection of items sold in the Media Oasis catalogue, these will include a list of movies, TV Show boxsets, DVD’s, CD’s, Video Games and any other form of physical media. |
| CUSTOMER – Allow clients to add desired items to their order via shopping cart | After browsing through items, the system should allow the client to purchase an item, so they should be able to add items into a shopping cart that will manage the items of the order. |
| CUSTOMER – Allow clients to view and modify their order in the shopping cart | The system should allow the client to check their shopping cart and provide the ability to modify it (i.e. changing the quantity of an item or removing an item from the shopping cart – both of which would cause the total order cost to be updated.) |
| CUSTOMER – Allow clients to place orders and receive order confirmations. | When the client is ready to order, the system should allow the client to enter the checkout screen from the shopping cart, where they can input their payment details and finally place a successful order. The application should then display an order confirmation message and send an order confirmation push-notification specifying the details of the order to the client. |
| CUSTOMER – Allow clients to write feedback/reviews and provide ratings for various items. | The system should allow the client to write and provide positive or negative feedback for items being sold on the catalogue as well as a rating bar element that will allow the client to scale their opinion toward sold items via a 5 star rating. |
| CUSTOMER – Allow clients to receive push-notifications regarding order status updates. | The system should allow the client, who has placed one or more orders at this point – to receive push-notifications about their order and its status (i.e. if the order is ready, the status of that order changes from “preparing” to “ready to be collected”, When this update is applied, a push-notification is sent to the client, informing them of the new status of the order. |
| ADMIN – Allow administrators to register and login to specialized accounts in order to access a core admin menu. | The system should allow administrators to login to their accounts in order to access a core admin menu that manages key information such as payment info, items for sale i.e. the overall database as well as customers accounts, feedback and orders.  **In terms of registration, allowing administrators to register on the registration screen can lead to security issues, so by default – a single admin account is initially added to the database and that particular administrator can register accounts through the core admin menu to allow other administrators to login. This prevents any non-admin client from making an administrator account.** |
| ADMIN – Allow administrators to manage items being sold on the catalogue. | The system should allow administrators to add, edit, update or delete items from the list of items being sold on the Media Oasis catalogue. |
| ADMIN – Allow administrators to view incoming orders placed by clients, including order information and status. | As clients are placing orders; the system should allow administrators to view these placed orders with detailed information such as items included in the order, quantity, total cost, date and time of the order and the order status. |
| ADMIN – Allow administrators to update the status of orders | The system should allow administrators to update the order status attribute of an order (essentially updating the Order record within the database).  The order status can be updated to 3 values: “Cancelled”, “Preparing” or “Ready to Collect”. |
| ADMIN – Allow administrators to receive and check client feedback and ratings on various items. | If clients give feedback on an item and provide a rating (out of 5 stars via the rating bar element) - The system should allow administrators to view client feedback, ratings and approve them to allow them to become visible to other clients. |
| ADMIN – Allow administrators to send push-notifications to clients/customers regarding order status updates or promotions | The system should allow administrators to send push-notifications to registered clients about any order status updates and promotional material. |

1. **Non-functional requirements**

*Outlining non-functional requirements that the application must meet from the varying components forming the entire application.*

*Non-functional requirements form a set of criteria that describe how the system and its features should operate and behave under any existing or non-existing constraints. The criteria focuses on quality attributes such as performance, security, scalability, usability, maintainability.*

|  |  |
| --- | --- |
| **What the system should do** | **Requirement** |
| SYSTEM – Provide a simple effective user interface to enhance **usability**. | The system should have a visually pleasant but effective user interface design that is navigable for the client – this includes consistent layout and clear presentation of UI components such as buttons, text input fields etc. |
| SYSTEM – Enhance **maintainability** by forming the system with clear, documented and tested modular components | High **maintainability** allows the application to be maintained or enhanced for the future, having modular components written in legible, clear, readable code with documentation and high testability via unit tests will allow the system to perform with increased **reliability** as well as **scalability** due to components being modular allowing them to be expanded, made more efficient or allowing the creation of new components entirely. |
| CUSTOMER – Provide **validation** of registration details to increase **integrity** and **security.** | When a client registers their account on to the system and has entered all their details, the process of **validation** should initiate to ensure inputted data meets specific requirements i.e. no null fields, alphabetic characters in the phone number field, password must be 8 characters long, contain a unique character, uppercase letter etc.  This enhances **integrity** of the system because it shows consistency of data being stored e.g. the phone number attribute only stores a set of numbers and not any letters, it enhances **security** by forcing clients to create stronger passwords. |
| CUSTOMER – Securely store username and password data to enhance **security**. | Upon account creation, the system should encrypt username and password details to increase **security** and store them in the database to prevent directly storing information in plain text that is highly vulnerable to unauthorised access in the scenario that the database is hacked and accessed by an unauthorised individual. |
| CUSTOMER – Provide authentication and secure login to enhance **security**. | When a customer wants to login to their account on the system, the system should check if the inputted password matches the encrypted password in the database. An authentication system will also be implemented giving the client 3 attempts to input the correct account information else they will need to wait for a period of time before logging in again. This reduces the effectiveness of brute force attacks and helps increase **security**. |
| ADMIN – Prevent customers from creating an admin account to enhance **security**. | Initially there would be at least one master admin account in the system/database, this higher-level admin account has the privilege of modifying the admin account database and thus can be used to create administrator accounts via the core admin menu. Account info can then be shared by this particular admin to new admin clients who can login and access the core admin back-end functionalities albeit they will not be able to create admin accounts as only a higher-level admin account can have access to this feature. This design increases **security** by negating the need for allowing admin account registration from the registration screen which is not secure as any non-admin client would be able to register as an admin through this method and it prevents non-high-level admins from creating their own one or more admin accounts. |

1. **Use cases with descriptions and diagrams**
2. **UML designs**
3. **User interface designs**

# Database

1. **How databases will be associated with this application**
2. **Database tables**
3. **ERD – Entity Relationship Diagram**

# Software architectural patterns used for developing the application including UI Logic, Input Logic and Business Logic

# Testing and security considerations

1. **Testing**
2. **Security considerations and practices**

# Final reflection

1. **An evaluation of the system and reflection**

The development for this application started slowly in August 2024 as it was still in its planning stages with brief drafts, but the development started in October 2024

1. **Overview of the software development life cycle (SDLC) used for the application**

Tools

* Android Studio
* Figma to design user-interface
* Google Images to locate images for items that will be sold
* Doxygen for documentation

Planned UI Design on Figma are located in the Media Oasis UI Screen Designs From Figma.pdf file, which displays all the planned user-interface activity designs in terms of colour, font, positioning, spacing and UI controls.

A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generatedA person looking at a shelf

Description automatically generated

Figure 3 Example of UI Design for Payment

Figure 2 Example of UI Design for Menu

Figure 1 Example of UI Design for Login

**References**

* Rounded Button Tutorial [*https://medium.com/swlh/rounded-button-in-android-studios-fbd1ee0b3161#:~:text=Color%20and%20corner%20radius%20for,%E2%80%9D%40drawable%2Frounded\_corner%20%E2%80%9D%20.*](https://medium.com/swlh/rounded-button-in-android-studios-fbd1ee0b3161%23:~:text=Color%20and%20corner%20radius%20for,%E2%80%9D%40drawable%2Frounded_corner%20%E2%80%9D%20.)
* White Border Layout Code <https://stackoverflow.com/questions/38787794/android-adding-border-around-textview>

*<?xml version="1.0" encoding="utf-8"?>*

*<shape xmlns:android="http://schemas.android.com/apk/res/android"*

*android:shape="rectangle" >*

*<stroke*

*android:width="2dp"*

*android:color="#cdcdcd" />*

*</shape>’*

* Adding custom fonts <https://developer.android.com/develop/ui/views/text-and-emoji/fonts-in-xml>
* Changing the launcher icon of your Android app in Android Studio <https://www.youtube.com/watch?v=bJjHgWjiAKw>
* Showing / Hiding password in editText of type textPassword <https://www.scribd.com/document/716398701/Android-Show-Hide-Password-in-Edittext>
* Changing spinner text size and colour <https://stackoverflow.com/questions/9476665/how-to-change-spinner-text-size-and-text-color>
* Changing the colour of the ratingBar <https://stackoverflow.com/questions/32810341/android-change-color-of-ratingbar-to-golden>
* Slidable Navigation Drawer Tutorial and Explanation <https://www.youtube.com/watch?v=do4vb0MdLFY>
* Fragments Tutorial and Explanation <https://www.youtube.com/watch?v=-vAI7RSPxOA>
* Alert Dialog and Buttons + Status Bar Colour modification <https://www.geeksforgeeks.org/how-to-set-buttons-inside-an-alert-dialog-in-android/>
* Reducing size of the ratingBar <https://smartandroidians.blogspot.com/2020/02/how-to-reduce-size-of-ratingbar.html>
* CountDown Timer <https://stackoverflow.com/questions/54095875/how-to-create-a-simple-countdown-timer-in-kotlin>
* Fixing canvas drawing too large issue: <https://www.reddit.com/r/androiddev/comments/px384u/javalangruntimeexception_canvas_trying_to_draw/>

*“u/CaptainRoger10 - Because the image size is too large, reduce the image size.”*

* Using findViewById() method within a fragment<https://www.repeato.app/how-to-use-findviewbyid-in-a-fragment/>
* Using the Kotlin addTextChangeListener <https://stackoverflow.com/questions/40569436/kotlin-addtextchangelistener-lambda>

**Image Sources**

* Media Oasis logo [*https://logo.com/logos/artificial-intelligence*](https://logo.com/logos/artificial-intelligence)
* Man looking at DVD Section (Splash) [*https://www.istockphoto.com/photo/this-better-have-loads-of-action-in-it-gm158220812-22390209*](https://www.istockphoto.com/photo/this-better-have-loads-of-action-in-it-gm158220812-22390209)
* Logout Button *<https://www.flaticon.com/free-icon/log-out_7756285?term=logout&page=1&position=29&origin=tag&related_id=7756285>*
* [placeholder image](https://www.flaticon.com/free-icon/log-out_7756285?term=logout&page=1&position=29&origin=tag&related_id=7756285) *[https://archive.org/details/placeholder-image](https://www.flaticon.com/free-icon/log-out_7756285?term=logout&page=1&position=29&origin=tag&related_id=7756285)*
* Checkmark [*https://commons.wikimedia.org/wiki/File:Eo\_circle\_green\_checkmark.svg*](https://commons.wikimedia.org/wiki/File:Eo_circle_green_checkmark.svg)
* PayPal [*https://upload.wikimedia.org/wikipedia/commons/a/a4/Paypal\_2014\_logo.png*](https://upload.wikimedia.org/wikipedia/commons/a/a4/Paypal_2014_logo.png)
* Credit / debit card [*https://www.kindpng.com/picc/m/13-130835\_credit-card-credit-card-vector-png-transparent-png.png*](https://www.kindpng.com/picc/m/13-130835_credit-card-credit-card-vector-png-transparent-png.png)
* Shopping Cart Icon[*https://www.flaticon.com/free-icon/shopping-cart\_1413908*](https://www.flaticon.com/free-icon/shopping-cart_1413908)