CHAPTER 1: INTRODUCTION

1.1 Abstract:

- The FabricFit Android Tailoring App revolutionizes the traditional tailoring experience by seamlessly integrating technology into the process of creating custom-made garments. This innovative application combines cutting-edge features with user-friendly interfaces to provide a comprehensive and efficient solution for both tailors and customers. It introduces an innovative approach to custom garment creation by incorporating distinct profile modules for administrators, tailors, and customers. This comprehensive application seamlessly integrates technology into the traditional tailoring process, fostering collaboration and enhancing the overall user experience.

- Admins curate and manage the fabric catalog, ensuring a diverse selection for customers to choose from. Tailors and customers can easily access the catalog to explore and select fabrics for their custom garments.

- Customers input their measurements accurately using the measurement module, ensuring precision in tailoring. Tailors can access and review these measurements to guarantee a bespoke fit for each order.

- The order module facilitates the seamless creation, tracking, and management of orders. Customers can place orders, tailors can view and fulfill them, and admins can monitor the overall order workflow for efficiency.

- Secure payment processing is integrated into the app, ensuring a smooth transaction process for customers. Admins can track financial transactions, providing transparency and security.

- The delivery module enables users to track the progress of their orders in real-time. Tailors can update order status, and customers receive notifications as their custom garments move through the production and delivery phases.

1.2 Existing System and Need for System:

❖ Existing System:

- **Manual Ordering:**
  - Previously, customers needed to manually contact tailors through phone calls or in-person visits to place orders for custom garments.
  - This manual process was time-consuming and inconvenient, especially for new or temporary customers.
- **Payment Method:**
  - Payment methods were limited to cash transactions, posing challenges for customers trying to pay for tailoring services.
  - Regular customers might have had to visit tailors in person to make monthly payments.
- **Order Tracking:**
  - Customers had no systematic way to track the status of their orders or the progress of their tailored garments.
  - This lack of transparency could lead to uncertainty and delays in receiving the finished garments.
- **Data Management:**
  - Data management for customer information, payment records, and garment design preferences was typically paper-based and prone to errors in the traditional tailoring model.

❖ Need For System:

- **Efficient Garment Ordering System:**

  There is a critical need for an efficient and user-friendly mobile application to streamline the process of ordering custom garments through FabricFit. The app should simplify the customization and ordering of tailored clothing, reducing the complexity associated with traditional tailoring methods.
- **Payment Convenience:**

  The FabricFit Tailoring App should offer multiple payment options to enhance customer convenience. This includes secure online payment methods as well as traditional payment options like cash on delivery. Providing diverse payment choices caters to the preferences and needs of a wide range of customers.

1.3 Scope of System:

- FabricFit allows users to seamlessly place orders for custom garments through its mobile application, simplifying and modernizing the tailoring experience.

- The app supports two convenient payment methods: online payment and cash on delivery. This flexibility caters to the diverse preferences of the customer base, ensuring a smooth transaction process.

- Integration with mapping services, such as Google Maps, assists users in selecting the nearest and most reliable tailor. This feature enhances accessibility and convenience in choosing a tailor for garment customization.

- FabricFit ensures that tailors adhere to stringent quality standards, guaranteeing that customers receive garments of exceptional quality and craftsmanship.

- The app streamlines the tailoring process, reducing wait times and enhancing overall service efficiency. This contributes to a faster turnaround for custom garments.

- FabricFit replaces traditional paper-based data management with a digital platform for maintaining customer information, payment records, and garment design preferences. This transition improves accuracy, efficiency, and accessibility of data.

1.4 Operating Environment Hardware and Software

HARDWARE REQUIREMENTS:

Type	**Laptop**	**Android Mobile**
Processor	11th Gen Intel(R) Core i5 or AMD	Runs on every processor
RAM	8 GB	8 GB
HDD	100 GB Min	100 MB
Graphics	AMD Radeon Vega8	-

SOFTWARE REQUIREMENTS:

Type	**Laptop**	**Android Mobile**
Operating System	64-Bit, windows 10	Android
IDE	Android Studio	-
Front-End	ANDROID, XML	-
Backend/Database	JAVA, FIREBASE	-

1.5 Technology Requirement:

- **Front End:** XML: XML (eXtensible Markup Language):

  XML is commonly used for designing layouts and storing structured data in Android applications.

  - Layouts: Define the UI components and structure of different screens using XML in Android's layout files (res/layout folder).
  - Data Exchange: XML can be used for data interchange, such as storing configuration settings or exchanging data between the app and a server.

- **Back End:** JAVA

  Java is the official programming language for Android app development and is used for implementing the app's logic and functionality.

  - Business Logic: Implement the backend logic of the FabricFit app using Java, including order processing, payment handling, and communication with Firebase services.
  - User Interactions: Handle user interactions, input validation, and navigation within the app using Java.

- **Database:** FIREBASE

  Firebase is a comprehensive platform for developing mobile and web applications. It offers various services, including real-time database, authentication, and cloud functions.

  - Real-time Database: Store and retrieve dynamic data, such as customer profiles, order details, and payment information, using Firebase Realtime Database.
  - Authentication: Implement user authentication to secure user accounts and data.
  - Cloud Functions: Leverage Firebase Cloud Functions to implement serverless functions that handle specific backend processes, such as order tracking updates and payment verification.
  - Cloud Storage: Store and retrieve images or other files related to fabric catalog or design previews.
  - Development Environment: Android Studio: Use Android Studio as the integrated development environment (IDE) for Android app development, supporting both XML layout design and Java coding.
  - Additional Considerations: Ensure secure communication by using HTTPS for network requests and Firebase Authentication for user login.

CHAPTER 2: PROPOSED SYSTEM

2.1 Feasibility Study:

- **Economical Feasibility:**

- Detailed Cost Analysis: A comprehensive cost analysis, encompassing development, maintenance, marketing, and operational costs, has been conducted for the FabricFit Tailoring App. The project demonstrates financial viability, with potential revenue streams from user subscriptions and fees paid by tailors.
- Positive ROI Projection: The projected Return on Investment (ROI) for the FabricFit Tailoring App is positive, anticipating growth in user adoption and revenue over time.

- **Technical Feasibility:**

- Positive Technical Feasibility: The technical feasibility assessment for developing the FabricFit Tailoring App is positive. Skilled developers and necessary tools are readily available for the project.
- Integration of Location Services: The integration of Google Maps for location services is technically feasible, enhancing user experience by allowing customers to easily locate the nearest and most reliable tailors.
- Digital Data Management: Transitioning from paper-based data management to digital systems is technically feasible, ensuring accuracy and efficiency in storing customer information, payment records, and garment design preferences.

- **Operational Feasibility:**

- User-Friendly GUI: The proposed system employs a fully graphical user interface (GUI), ensuring a user-friendly experience. All inputs are self-explanatory, even for individuals with limited technical expertise.
- Training and Support: Necessary training and support for both users and tailors can be provided, facilitating a smooth onboarding process and ongoing operational efficiency.
- Data Security and Privacy: Addressing data security and privacy concerns is feasible, ensuring that customer information and transactional data are handled securely and in compliance with relevant regulations.

2.2 Objectives of The Proposed System:

- **Efficient Garment Ordering:**

  Enable users to seamlessly place orders for custom garments through a user-friendly mobile application, eliminating the need for manual and time-consuming ordering methods.

- **Payment Convenience:**

  Provide customers with flexible payment options within the FabricFit Tailoring App, including secure online payment and cash on delivery. This caters to different customer preferences and ensures a convenient transaction process.

- **Customer Segmentation:**

  Implement tailored payment options for both regular and temporary customers. Regular customers can benefit from a monthly payment plan, while temporary customers have pay-as-you-go choices. Offer both online and cash payment options for increased flexibility.

- **Location Services Integration:**

  Utilize Google Maps integration within the FabricFit Tailoring App to assist users in identifying and selecting the nearest and most reliable tailors. This enhances convenience and accessibility in choosing a tailor for garment customization.

- **Quality Assurance:**

  Implement a system that ensures tailors within the FabricFit network adhere to strict quality standards, guaranteeing that customers receive garments of exceptional quality and craftsmanship.

- **Efficiency Improvement:**

  Streamline the tailoring process within the FabricFit app to reduce wait times, enhance service efficiency, and provide prompt delivery of custom garments to customers.

2.3 Module Specifications:

1. Profile Module:

- **Admin Profile:**

  Register/login: Secure registration and login system for administrators.

  View Profile: Enable administrators to view their profiles, providing details such as name, role, and contact information.

  Edit Profile: Allow administrators to edit and update their profiles, ensuring accurate and current information is maintained.

  View/Manage Customer: Access and manage customer accounts, including user information and order history.

  View/Manage Tailor Profile: Edit and update the tailor's business information, contact details, and branding.

- **Customer Profile:**

  Register/login: Customer can create accounts Register and Login.

  View Profile: Customer can view profile, access, and review their personal information, order history, and any preferences they have set.

  Edit Profile: Customer can edit profile information this includes modifying contact details, updating addresses, and ensuring that their profile accurately reflects their preferences and requirements.

  Book Appointment: Customers can schedule appointments with tailors.

- **Tailor Profile:**

  Register/login: Tailors can create accounts and log in.

  View Profile: Tailor can view profile, access details such as business information, contact details, and any branding elements associated with their profile.

  Edit Profile: Tailor can edit profile information. modify business details, update contact information, and ensure that their profile aligns with their current branding and business identity.

  Schedule Appointment: Tailors can manage their appointment schedule.

2. Catalog Module:

- **Admin:**

  Manage Catalog Records: Admin can manage and update catalog records, ensuring that they have accurate and up-to-date information to fulfill customer orders effectively.

- **Customer:**

  View Catalog: Customer can explore and view the FabricFit catalog, gaining insights into fabric designs, product listings, and categorized items.

  Select Catalog: Customers have the capability to select products from the catalog that align with their preferences and design choices.

- **Tailor:**

  Upload Catalog: Tailor has the capability to upload and manage Catalog Designs efficiently.

  Update Catalog Records: Tailor can update Catalog Designs, facilitating the maintenance of accurate and current Catalog Designs information.

  Manage Catalog Details: Tailor can manage catalog details, including quality, price, and quantity of the fabrics they offer.

  Change Prices, Quality, Quantity: Tailor can Change Catalog details as adjusting prices, updating quality attributes, and modifying quantities based on requirements.

3. Measurement Module:

- **Admin:**

  Manage Customer Records: Admin can manage and update customer records, ensuring that they have accurate and up-to-date information to fulfill customer measurement effectively.

- **Customer:**

  Upload Measurement Records: Customer have the capability to upload and manage Measurement records efficiently within the FabricFit customer portal.

  Update Measurement Records: FabricFit allows customer to update Measurement records, facilitating the maintenance of accurate and current measurement information.

  Delete Measurement Records: Customer have the authority to delete Measurement records as needed, adhering to data privacy and compliance requirements.

- **Tailor:**

  Maintain Measurement Records: The Maintain measurement Records functionality ensures the ongoing accuracy and completeness of customer information within the FabricFit system.

  View Measurement Records: Tailor has the authority to view customer records as needed, adhering to data privacy and compliance requirements.

4. Order Module:

- **Admin:**

  Maintain/ Manage Order Records: Admins have the capability to maintain and manage customer order records, ensuring accuracy and completeness of information related to customer orders.

- **Customer:**

  Make Order: Customers can initiate the order process, selecting tailors and specifying their garment requirements within the FabricFit app.

  Cancel Order: Customers have the flexibility to cancel orders if needed, providing a user-friendly and responsive order management system.

- **Tailor:**

  Update Order: Tailors can make real-time updates to the status of orders, ensuring accurate and timely information is communicated to customers

  View Order: Tailors have access to a list of incoming orders, providing them with details on the garments to be tailored and customer preferences.

  Maintain Order Records: Tailors can efficiently maintain order records, ensuring they are well-prepared for production and delivery based on detailed customer preference.

5. Payment and Delivery Module:

- **Payment Module:**

  Regular Customers: Monthly payment option, streamlining the process for ongoing services.

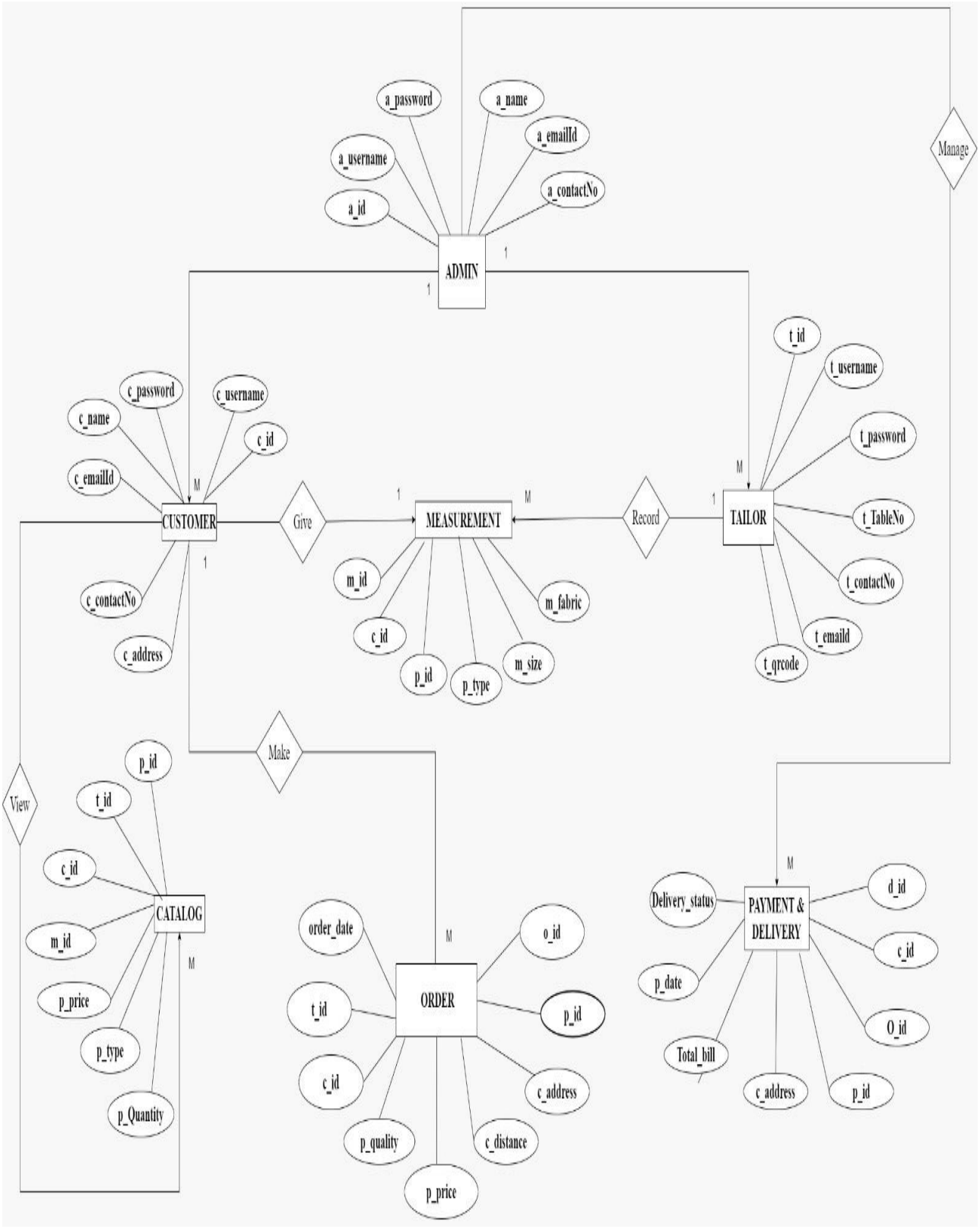
  Temporary Customers: Online payment: Allows temporary customers to pay immediately.

  Cash on Delivery: Payment option upon the delivery of their custom garments.

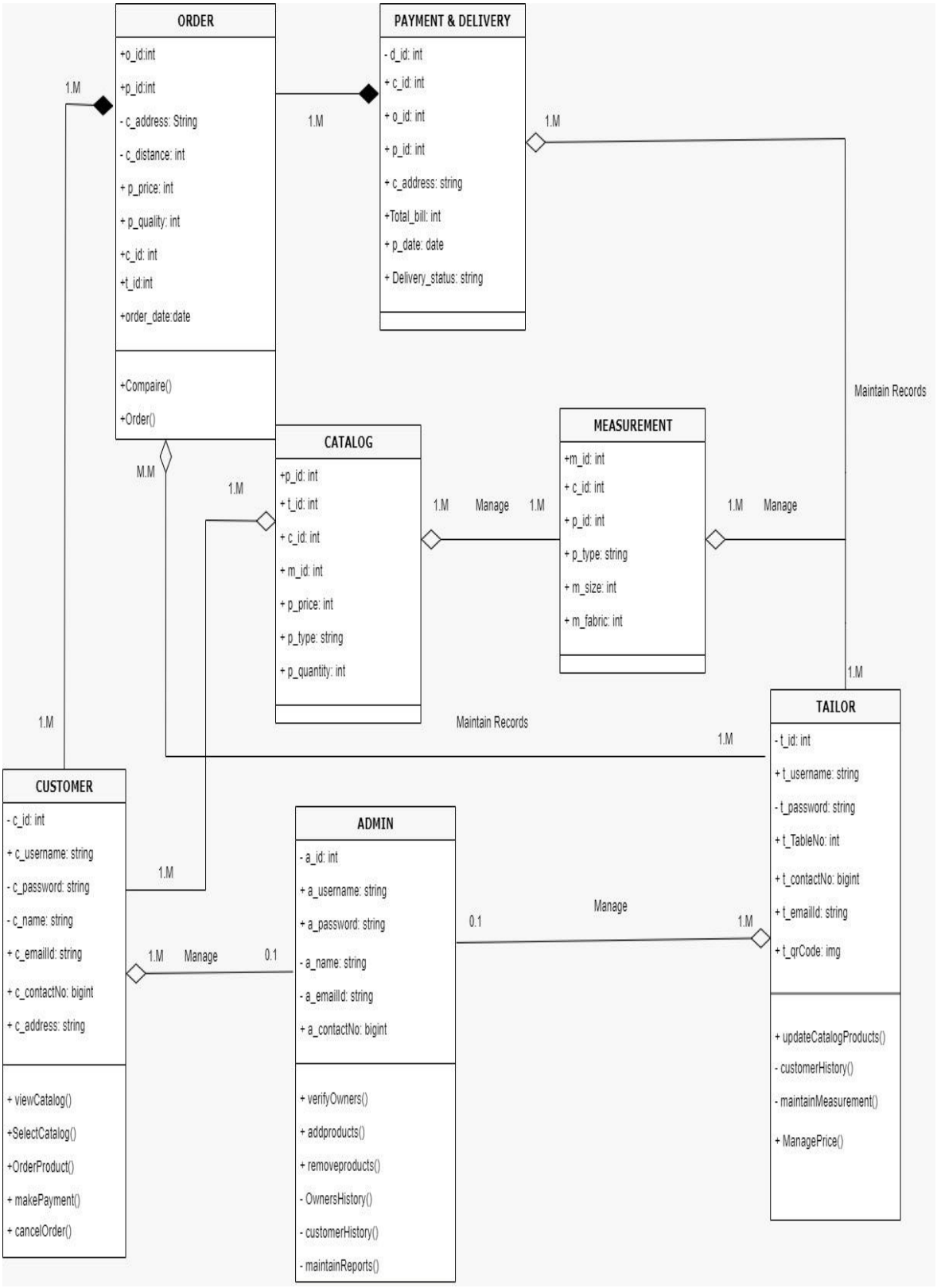
  Delivery Management: Tracks the status of their custom garment delivery in real-time.

CHAPTER 3: ANALYSIS AND DESIGN

3.1 Entity Relationship Diagram (ERD):

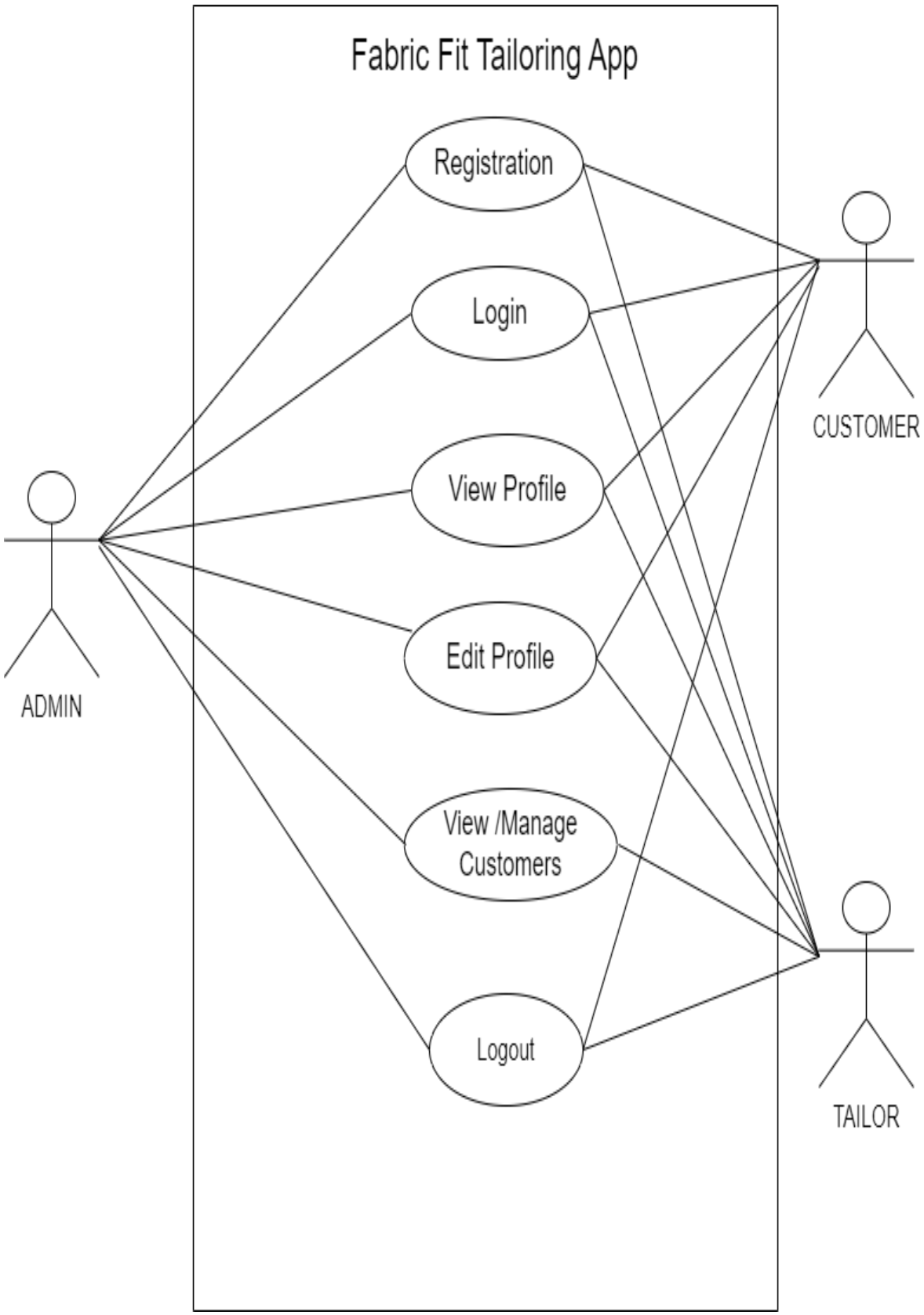


3.2 Class Diagram:

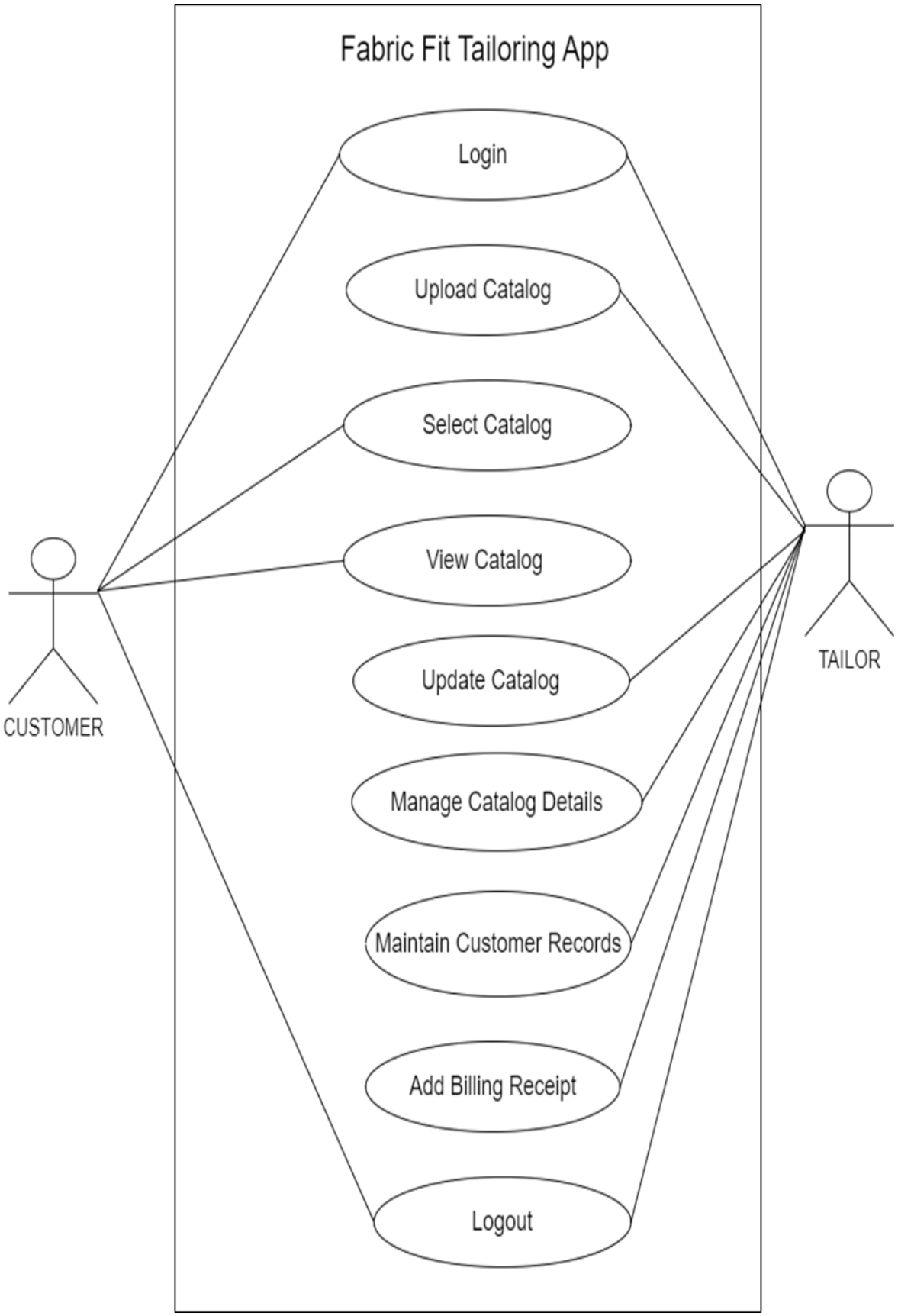


3.3 Use Case Diagrams:

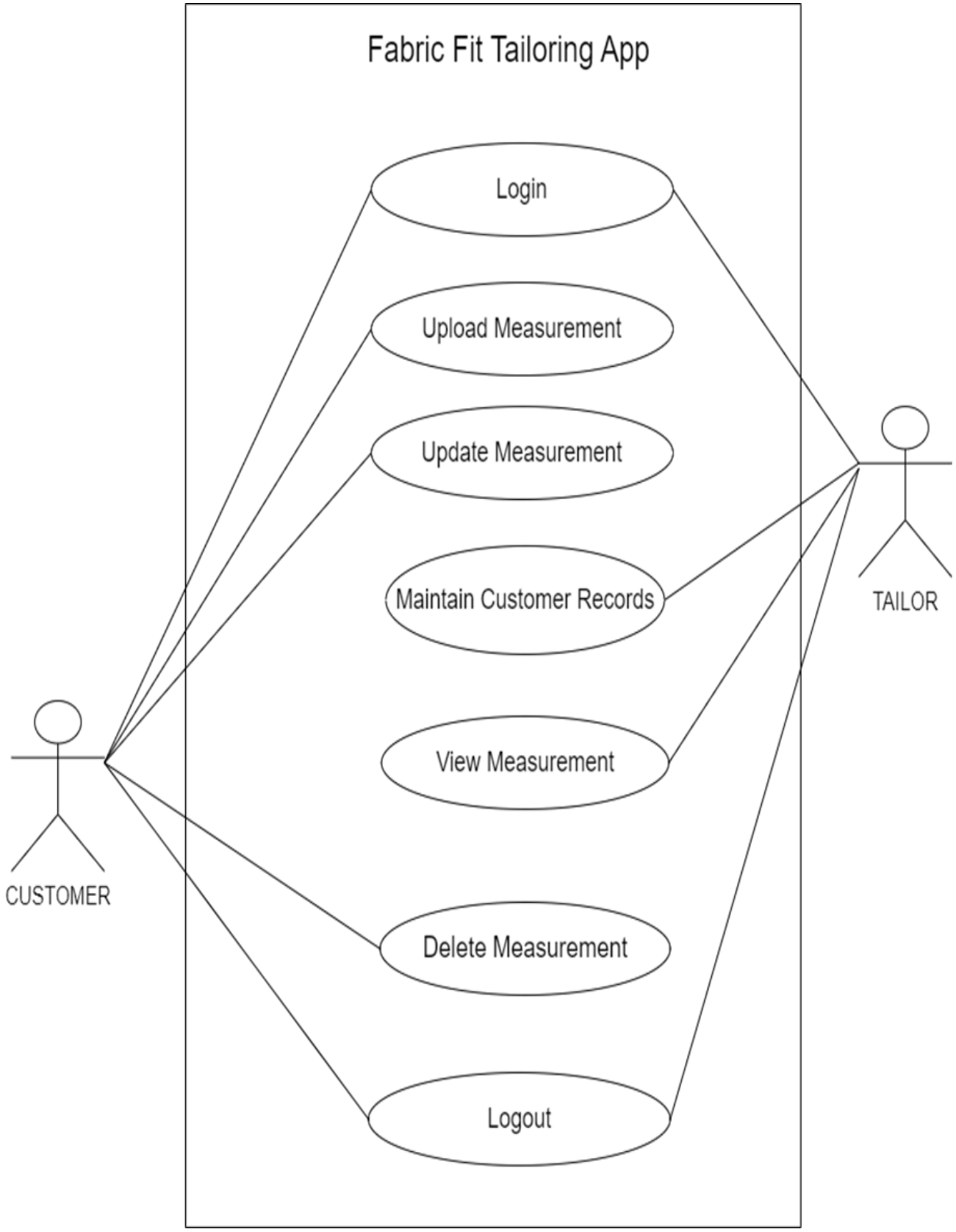
USE CASE DIAGRAM FOR PROFILE



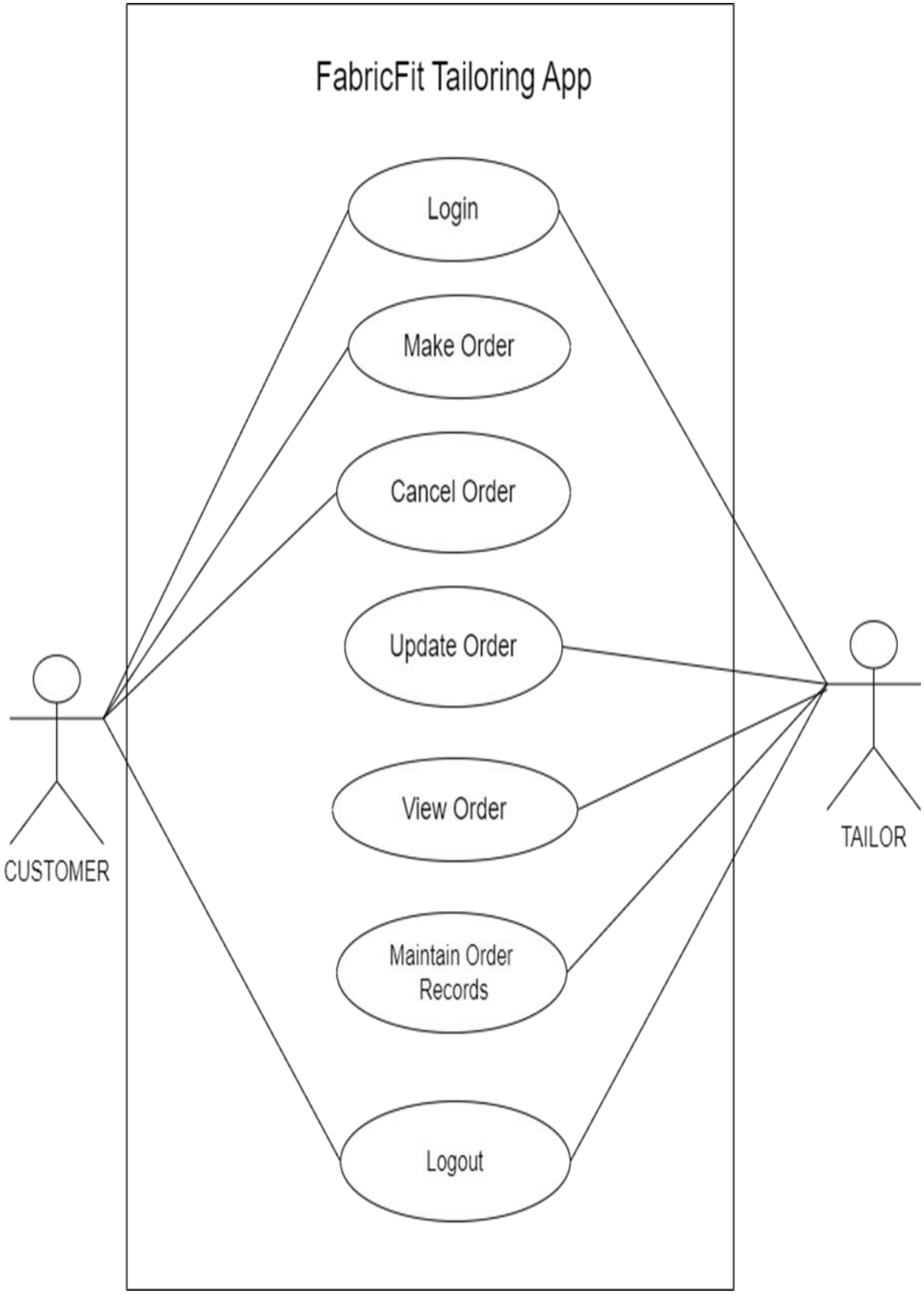
USE CASE DIAGRAM FOR CATALOG



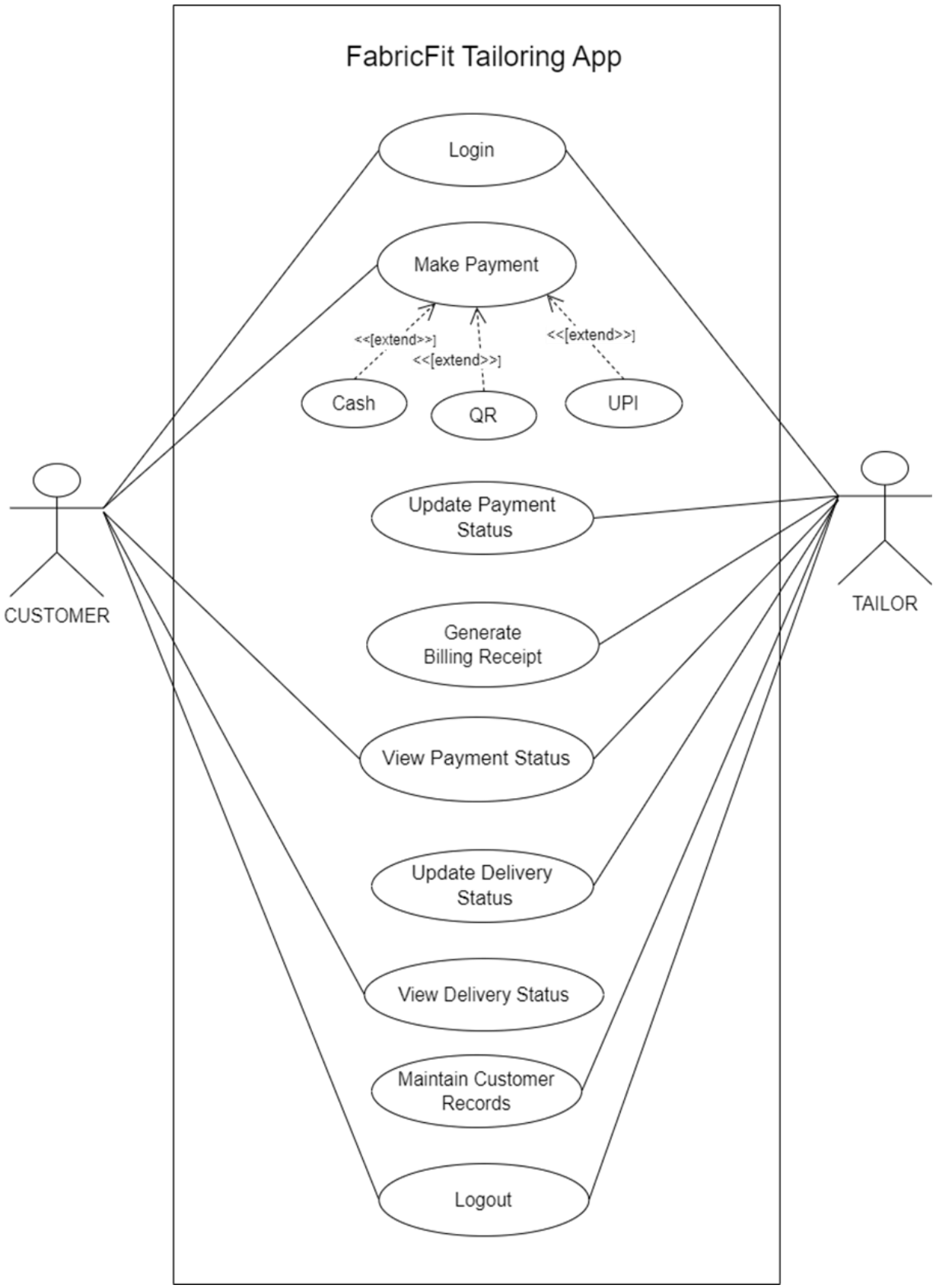
USE CASE DIAGRAM FOR MEASUREMENT



USE CASE DIAGRAM FOR ORDER

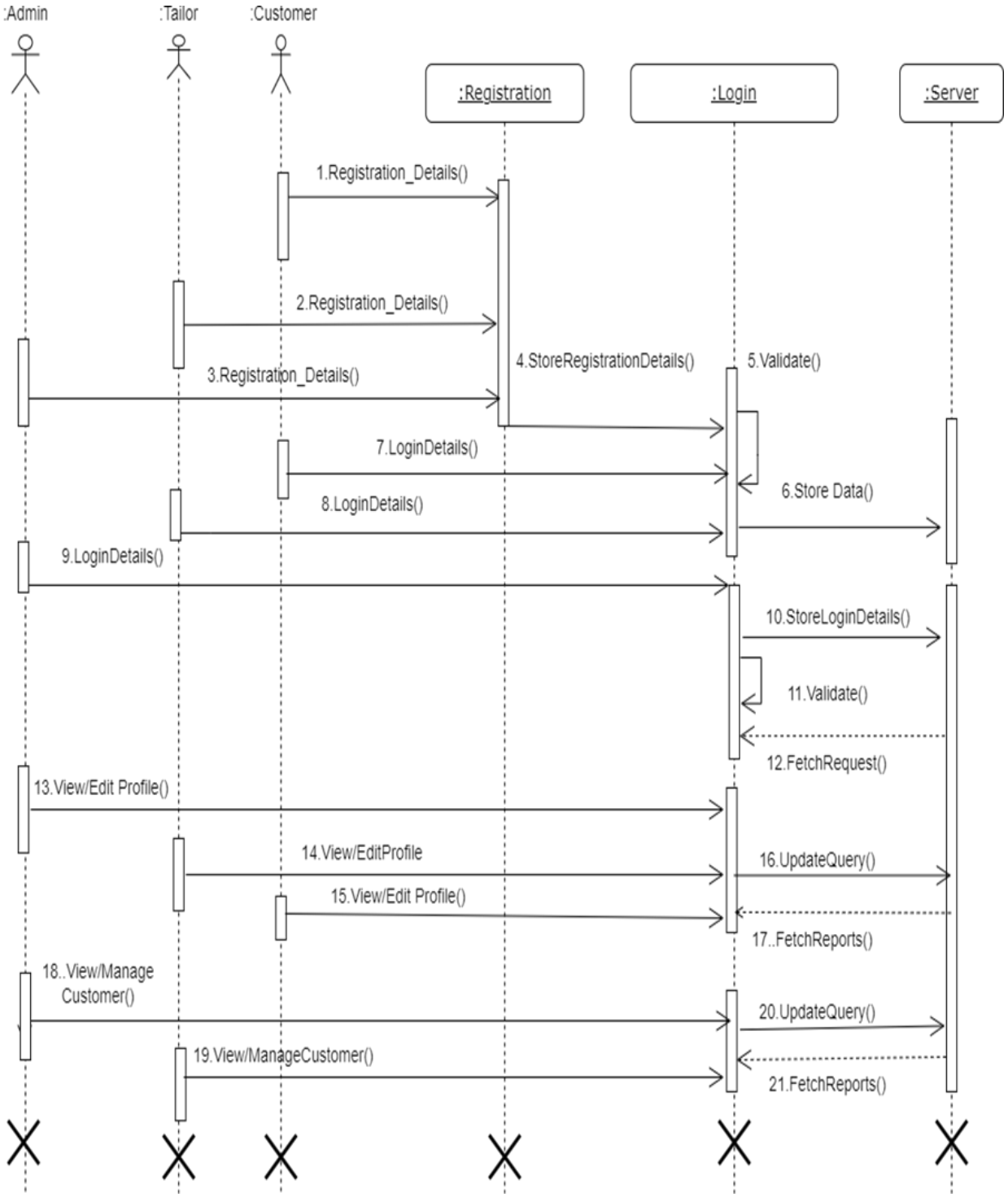


USE CASE DIAGRAM FOR PAYMENT AND DELIVERY

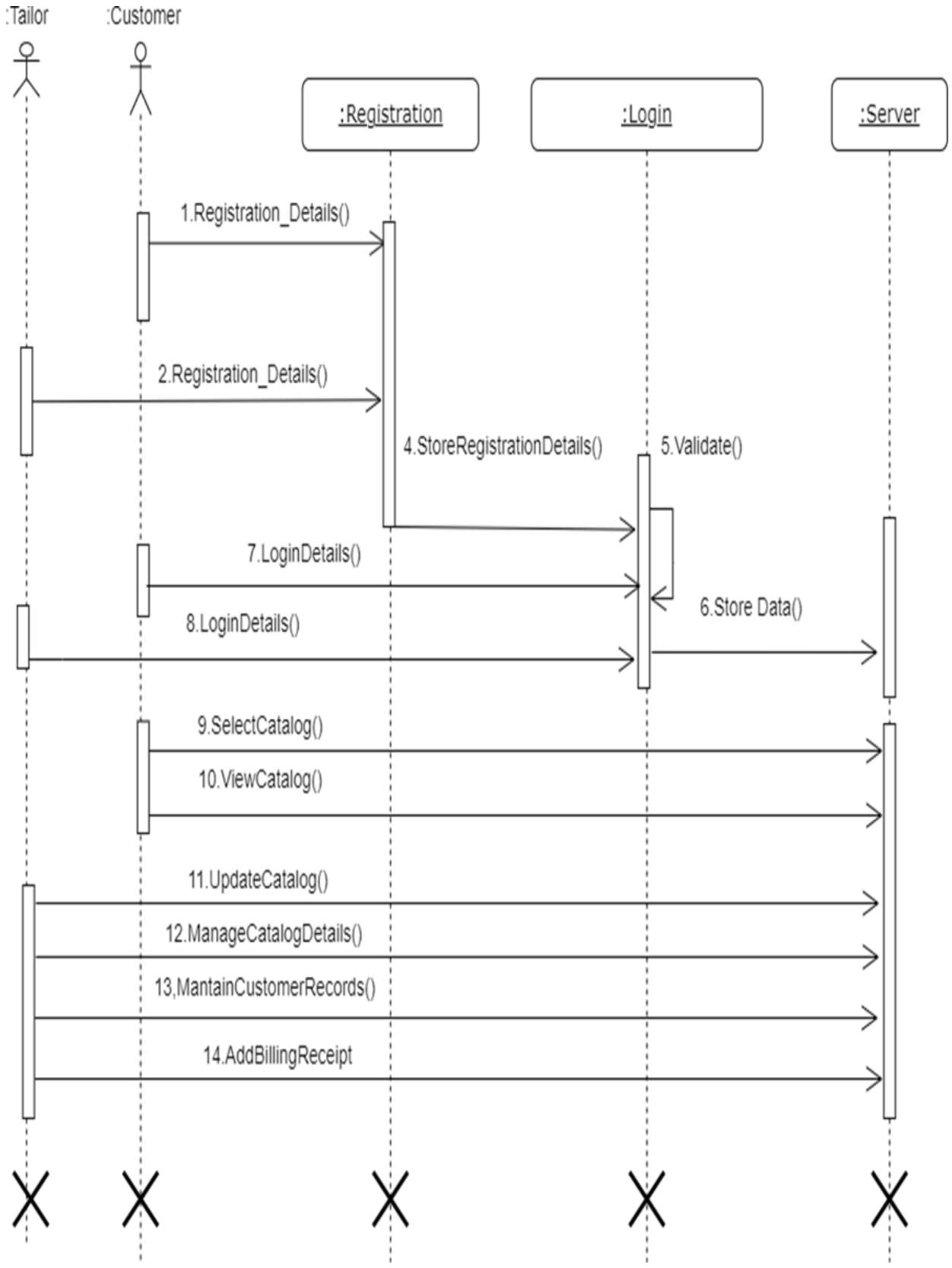


3.4 Sequence Diagram:

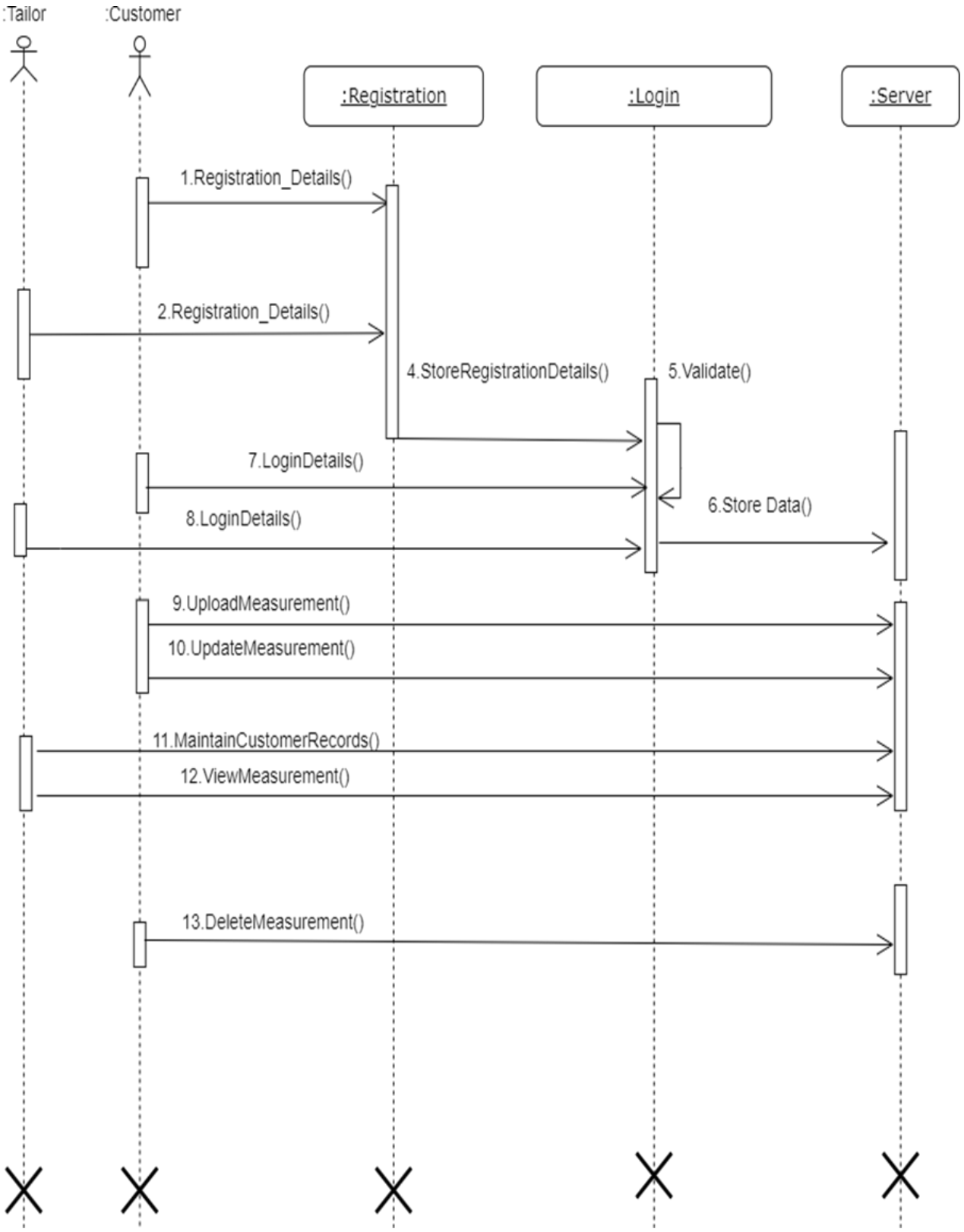
SEQUENCE DIAGRAM FOR PROFILE



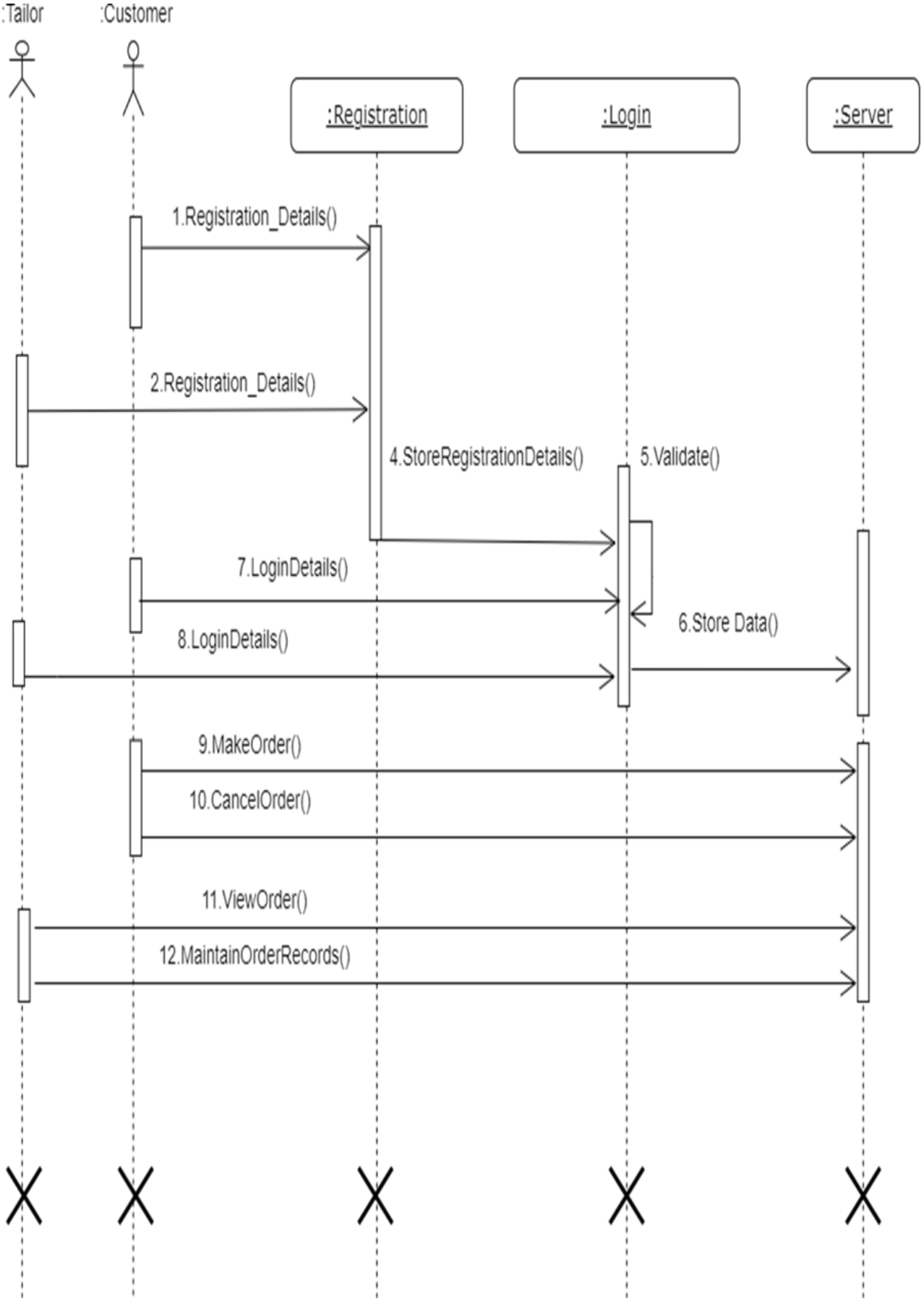
SEQUENCE DIAGRAM FOR CATALOG



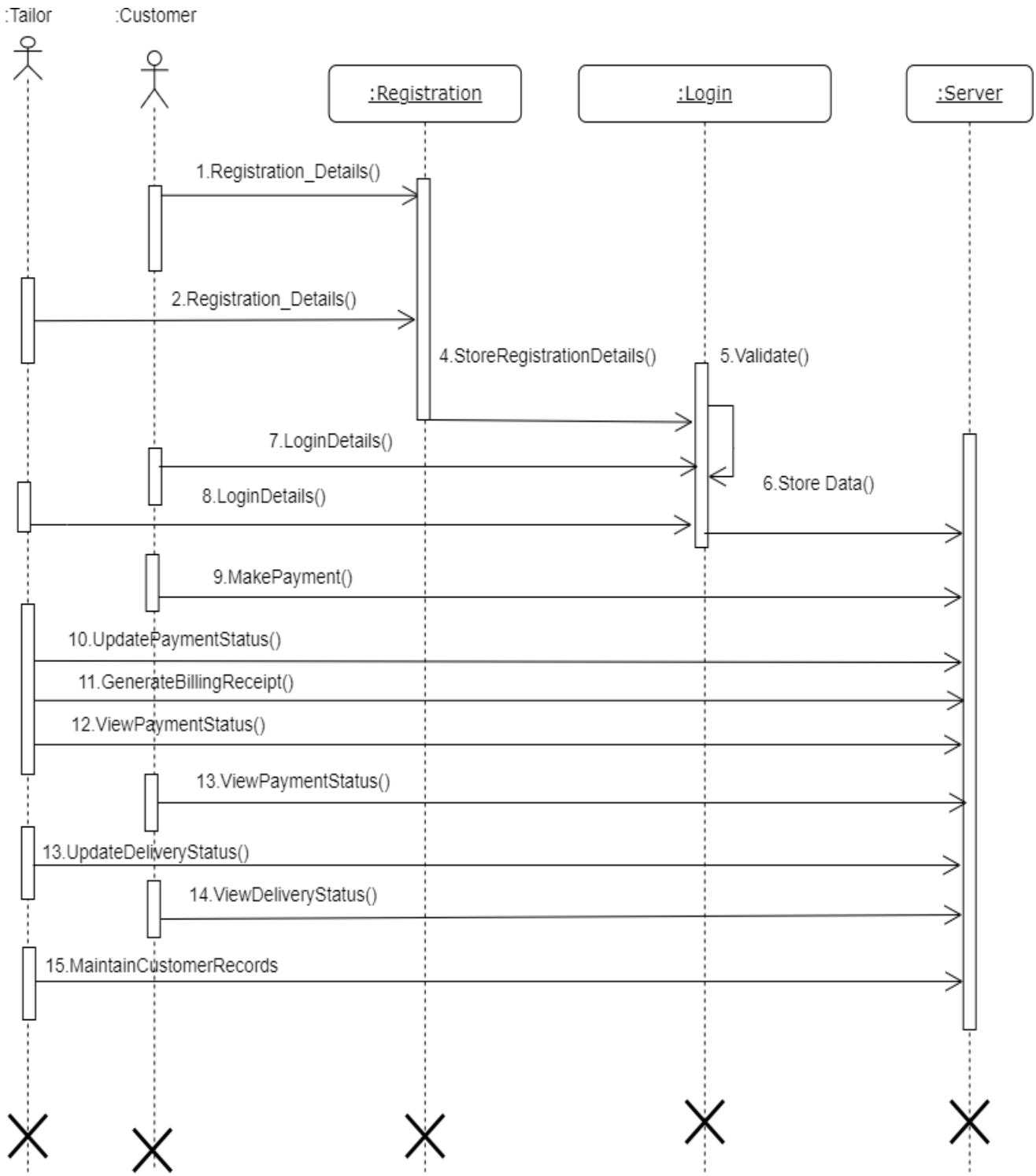
SEQUENCE DIAGRAM FOR MEASUREMENT



SEQUENCE DIAGRAM FOR ORDER

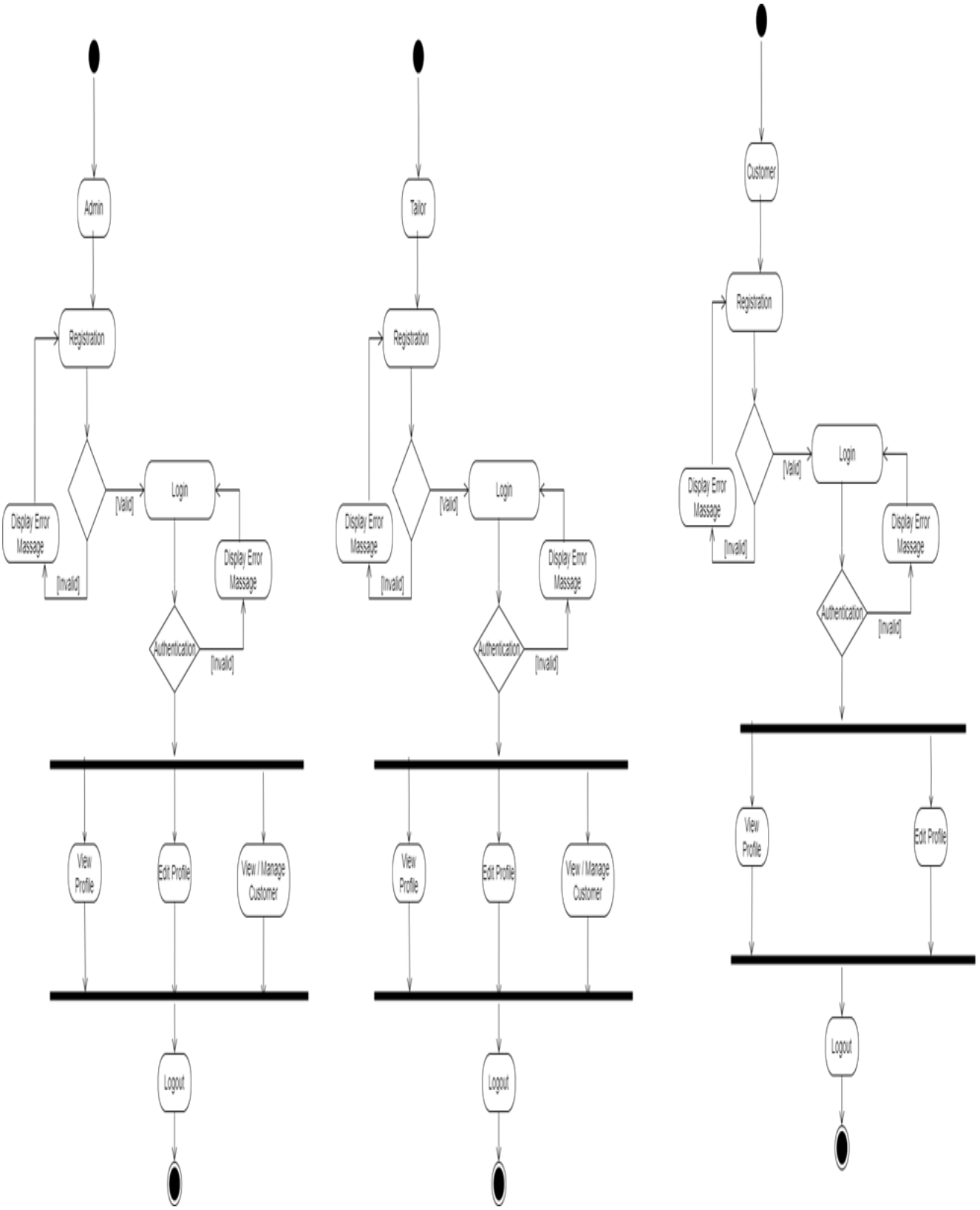


SEQUENCE DIAGRAM FOR PAYMENT AND DELIVERY

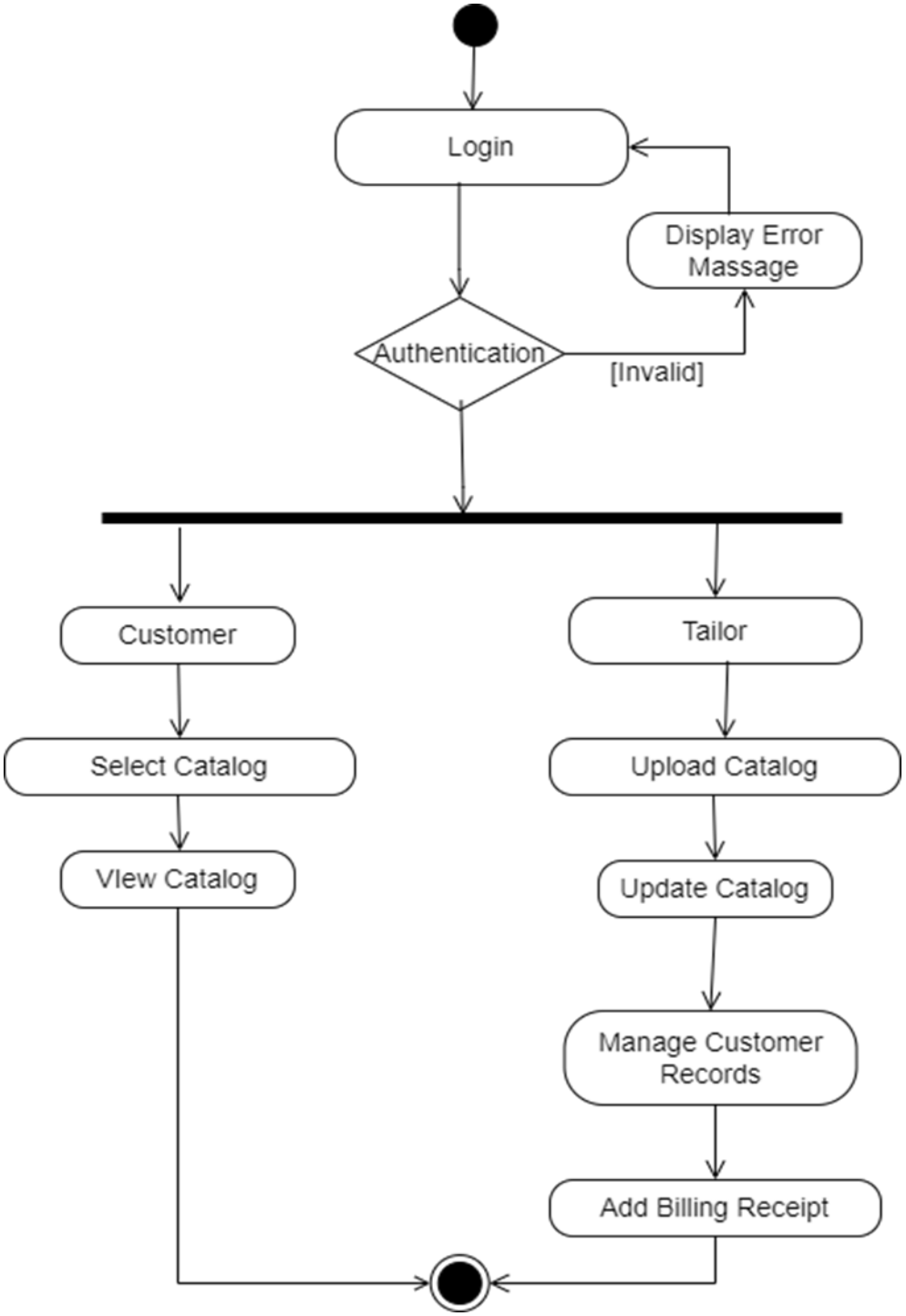


3.5 Activity Diagram

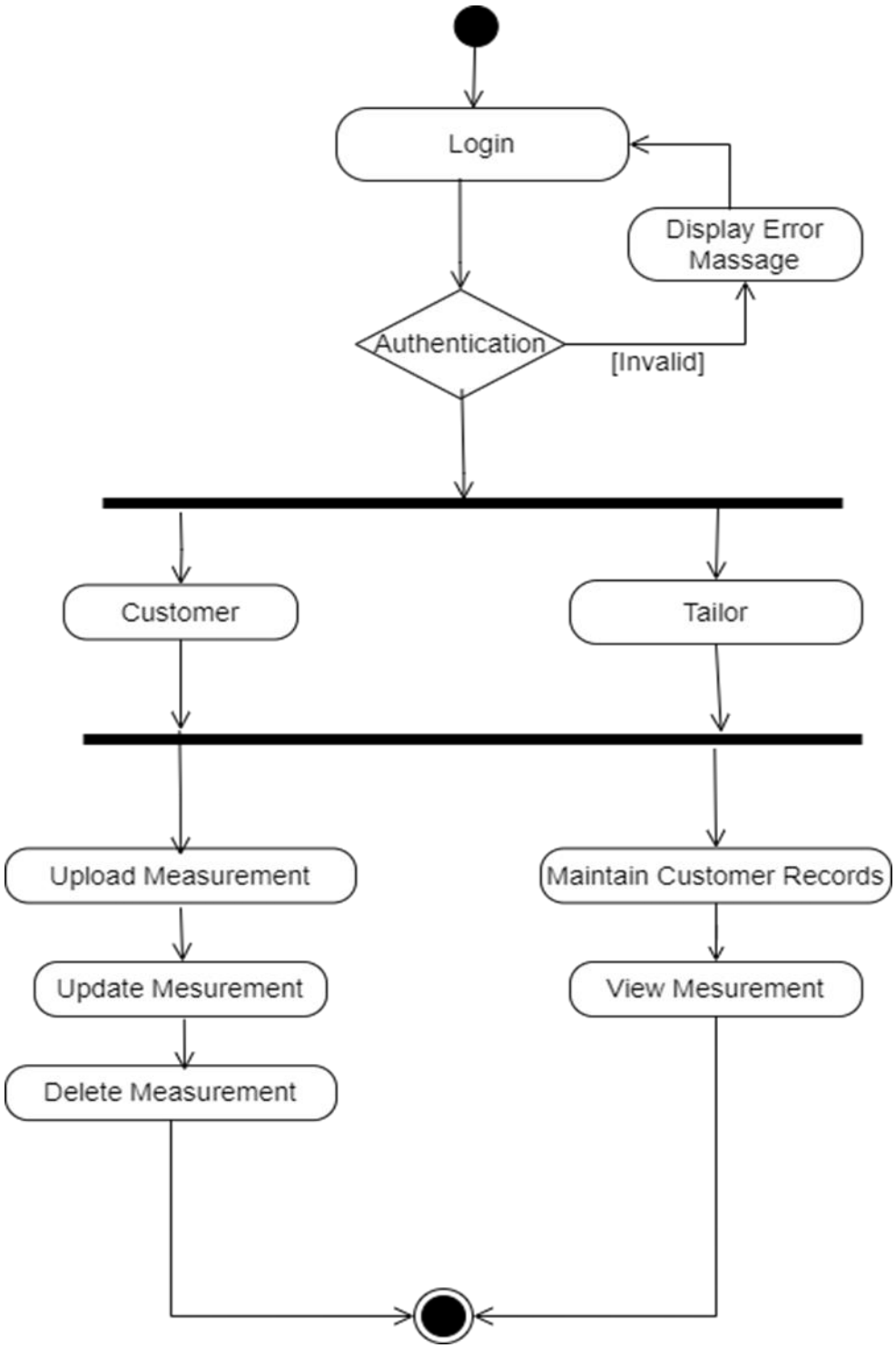
ACTIVITY DIAGRAM OF PROFILE



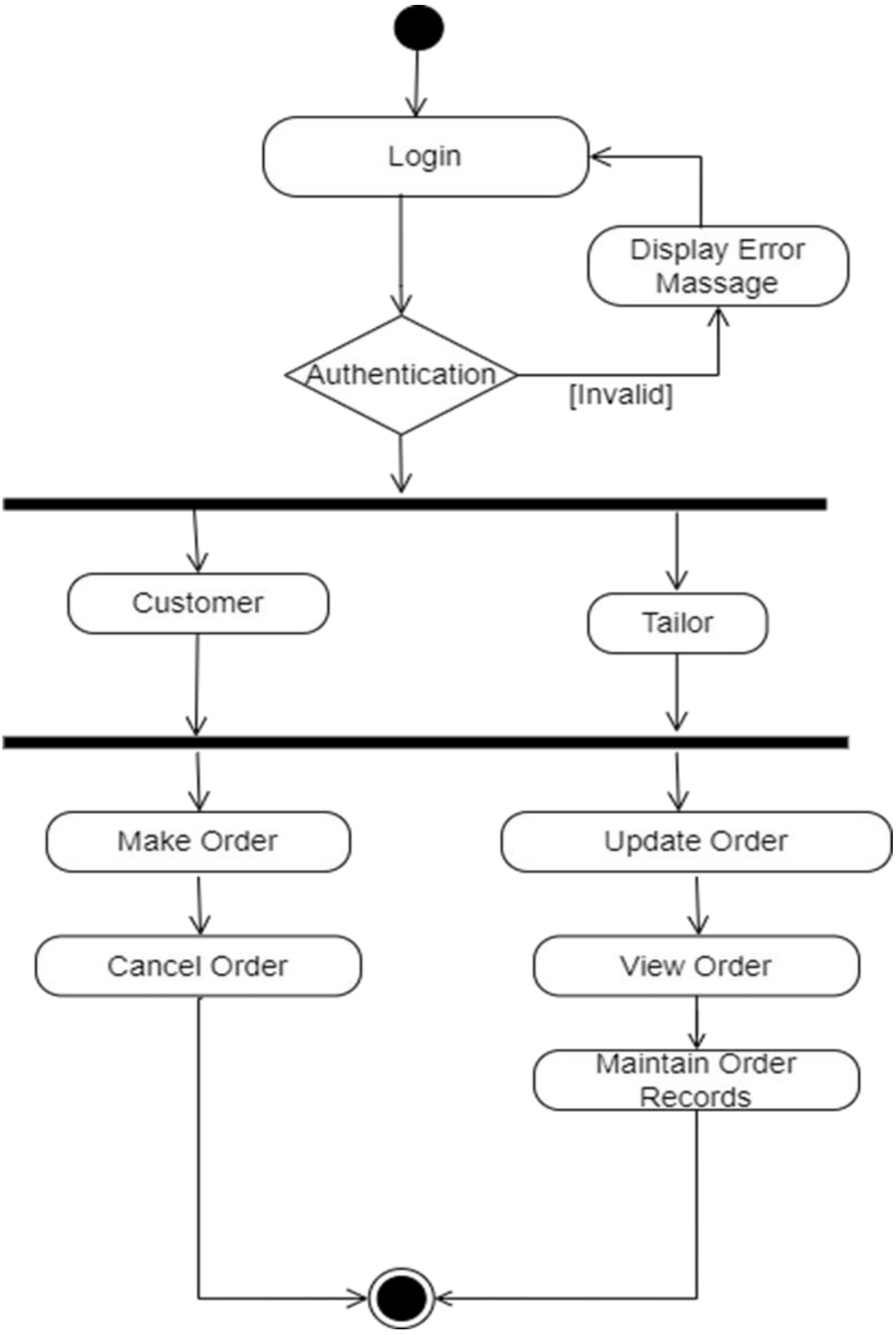
ACTIVITY DIAGRAM OF CATALOG



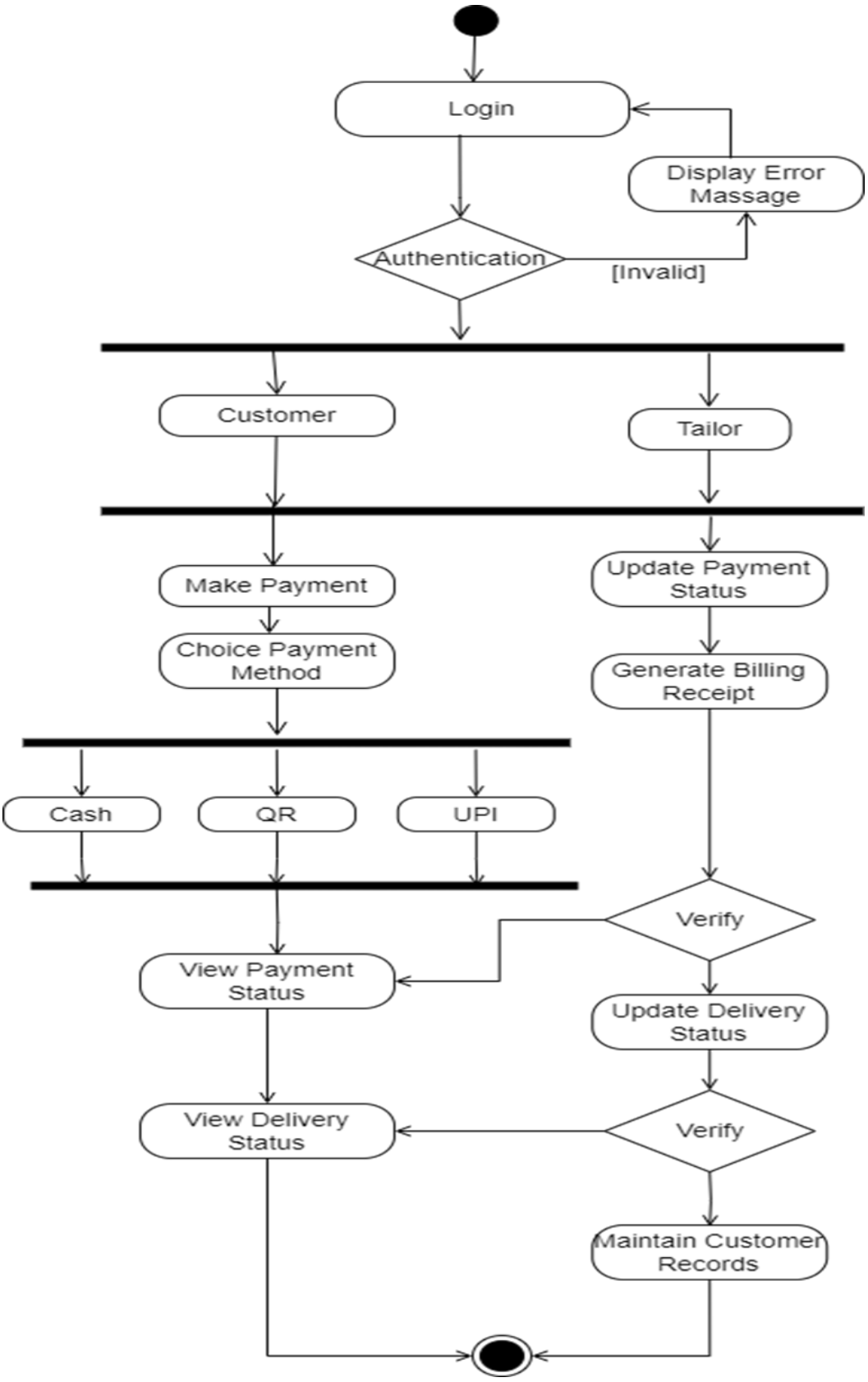
ACTIVITY DIAGRAM OF MEASUREMENT



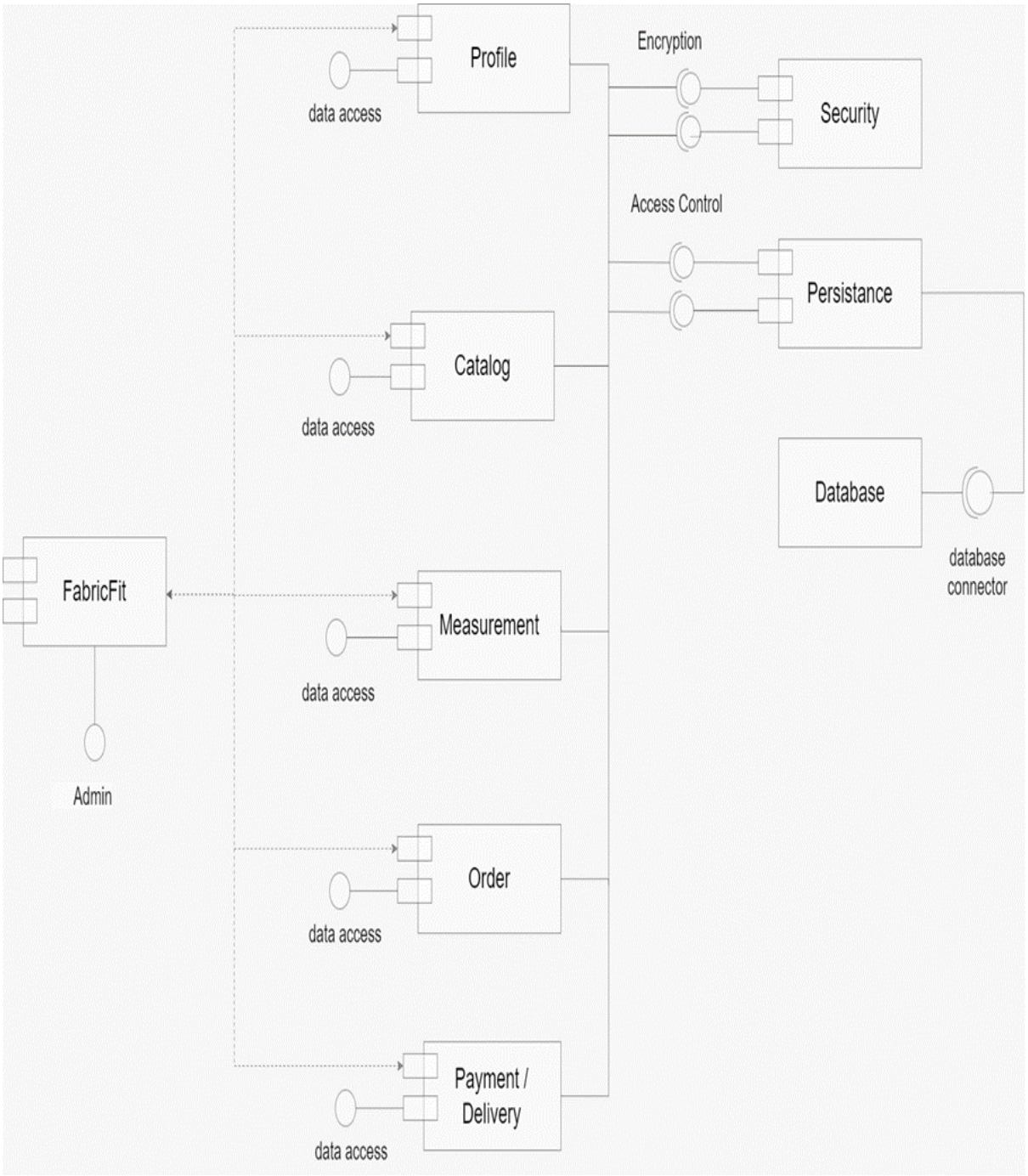
ACTIVITY DIAGRAM FOR ORDER



ACTIVITY DIAGRAM FOR PAYMENT AND DELIVERY



3.6 Component Diagram



3.7 Table Design

TABLE NAME: ADMIN

Sr.no	Field Name	Field Size	Data Type	Description	Constraints
1	a_id	8	Int	Admin id	Primary Key
2	a_username	10	Varchar	Admin Username	Not Null
3	a_password	12	Varchar	Admin Password	Not Null
4	a_name	12	Varchar	Admin Name	Not Null
5	a_emailId	27	Varchar	Admin Email Id	Not Null
6	a_contactNo	11	Int	Admin Contact	Not Null

TABLE NAME: CUSTOMER

Sr.no	Field Name	Field Size	Data Type	Description	Constraints
1	c_id	8	Int	Customer id	Primary Key
2	c_username	10	Varchar	Customer Username	Not Null
3	c_password	12	Varchar	Customer Password	Not Null
4	c_name	12	Varchar	Customer Name	Not Null
5	c_emailId	27	Varchar	Customer Email Id	Not Null
6	c_contactNo	11	Int	Customer Contact	Not Null
7	c_address	30	Varchar	Customer Address	Not Null

TABLE NAME: TAILOR

Sr.no	Field Name	Field Size	Data Type	Description	Constraints
1	t_id	8	Int	Tailor id	Primary Key
2	t_username	10	Varchar	Tailor Username	Not Null
3	t_password	12	Varchar	Tailor Password	Not Null
4	t_tableNo	12	Int	Tailor Table No	Not Null
5	t_emailId	27	Varchar	Tailor Email Id	Not Null
6	t_contactNo	11	Int	Tailor Contact	Not Null
7	t_qrcode	1	Int	Tailor QR code	Not Null

TABLE NAME: CATALOG

Sr.no	Field Name	Field Size	Data Type	Description	Constraints
1	p_id	8	Int	Product id	Primary Key
2	t_id	8	Int	Tailor id	Foreign Key
3	c_id	8	Int	Customer id	Foreign Key
4	m_id	8	Int	Measurement id	Foreign Key
5	p_price	5	Varchar	Product Price	Not Null
6	p_type	5	Varchar	Product type	Not Null
7	p_quantity	10	Varchar	Product Quantity	Not Null

TABLE NAME: MEASUREMENT

Sr.no	Field Name	Field Size	Data Type	Description	Constraints
1	m_id	8	Int	Measurement id	Primary Key
2	c_id	8	Int	Customer id	Foreign Key
3	p_id	8	Int	Product id	Foreign Key
4	p_type	9	Varchar	Product type	Not Null
5	m_size	4	Int	Material size	Not Null
6	m_fabric	10	Varchar	Material fabric	Not Null

TABLE NAME: ORDER

Sr.no	Field Name	Field Size	Data Type	Description	Constraints
1	order_date	8	Int	Order date	Primary Key
2	t_id	8	Int	Tailor id	Foreign Key
3	c_id	8	Int	Customer id	Foreign Key
4	p_quality	10	Varchar	Product Quantity	Foreign Key
5	p_price	5	Int	Product Price	Not Null
6	c_distance	10	Varchar	Customer Distance	Not Null
7	c_address	30	Varchar	Customer Address	Not Null
8	p_id	8	Int	Product Id	Foreign Key
9	o_id	8	Int	Order Id	Foreign Key

TABLE NAME: PAYMENT AND DELIVERY

Sr.no	Field Name	Field Size	Data Type	Description	Constraints
1	delivery_status	10	Int	Delivery Status	Not Null
2	p_date	8	Int	Product Date	Not Null
3	Total_bill	5	Int	Total Bill	Not Null
4	c_address	30	Varchar	Customer Address	Not Null
5	p_id	8	Int	Product Id	Primary Key
6	o_id	8	Varchar	Order id	Foreign Key
7	c_id	8	Varchar	Customer id	Foreign Key
8	d_id	8	Int	Delivery Id	Foreign Key

3.8 Data Dictionary:

Sr.no	Field Name	Field Size	Data Type	Description	Constraints	Table Name
1	a_contactNo	11	Int	Admin Contact No	Not Null	Admin
2	a_emailId	27	Varchar	Admin Email Id	Not Null	Admin
3	a_id	8	Int	Admin Id	Primary Key	Admin
4	a_name	12	Varchar	Admin Name	Not Null	Admin
5	a_password	12	Varchar	Admin Password	Not Null	Admin
6	a_username	10	Varchar	Admin Username	Not Null	Admin
7	c_address	30	Varchar	Customer Address	Not Null	Customer
8	c_contactNo	11	Int	Customer Contact No	Not Null	Customer
9	c_emailId	27	Varchar	Customer Email Id	Not Null	Customer
10	c_id	8	Int	Customer Id	Primary Key, Foreign Key, Foreign Key	Customer, Measure ment, Order
11	c_name	12	Varchar	Customer Name	Not Null	Customer
12	c-password	12	Varchar	Customer Password	Not Null	Customer
13	c_username	10	Varchar	Customer Username	Not Null	Customer
14	delivery_status	10	Varchar	Delivery Status	Not Null	Payment & Delivery
15	m_id	8	Int	Measurement Id	Primary Key	Measurement
16	m_size	10	Varchar	Measurement Size	Not Null	Measurement
17	o_id	8	Int	Order Id	Primary Key, Foreign Key	Order, Payment & Delivery
18	o_quantity	10	Int	Order Quantity	Not Null	Order
19	o_totalamt	10	Int	Order Total Amount	Not Null	Order
20	order_date	8	Int	Order Date	Not Null	Order
21	pat_descri	10	Varchar	Pattern Description	Not Null	Catalog
22	pat_id	8	Int	Catalog Id	Primary Key, Foreign Key, Foreign Key	Catalog, Measure ment, Order
23	pat_img	30	Varchar	Pattern Image	Not Null	Catalog

24	pat_name	10	Varchar	Pattern Name	Not Null	Catalog
25	pat_prices	10	Int	Pattern Prices	Not Null	Catalog
26	p_id	8	Int	Payment Id	Primary Key	Payment & Delivery
27	payment_status	10	Varchar	Payment Status	Not Null	Payment & Delivery
28	t_contactNo	10	Int	Tailor Contact No	Not Null	Tailor
29	t_emailId	27	Varchar	Tailor Email Id	Not Null	Tailor
30	t_id	8	Int	Tailor Id	Primary Key, Foreign Key	Tailor, Order
31	t_password	12	Varchar	Tailor password	Not Null	Tailor
32	t_qrcode	1	Int	Tailor QR Code	Not Null	Tailor
33	t_tableNo	12	Int	Tailor Table No	Not Null	Tailor
34	t_username	10	Varchar	Tailor Username	Not Null	Tailor

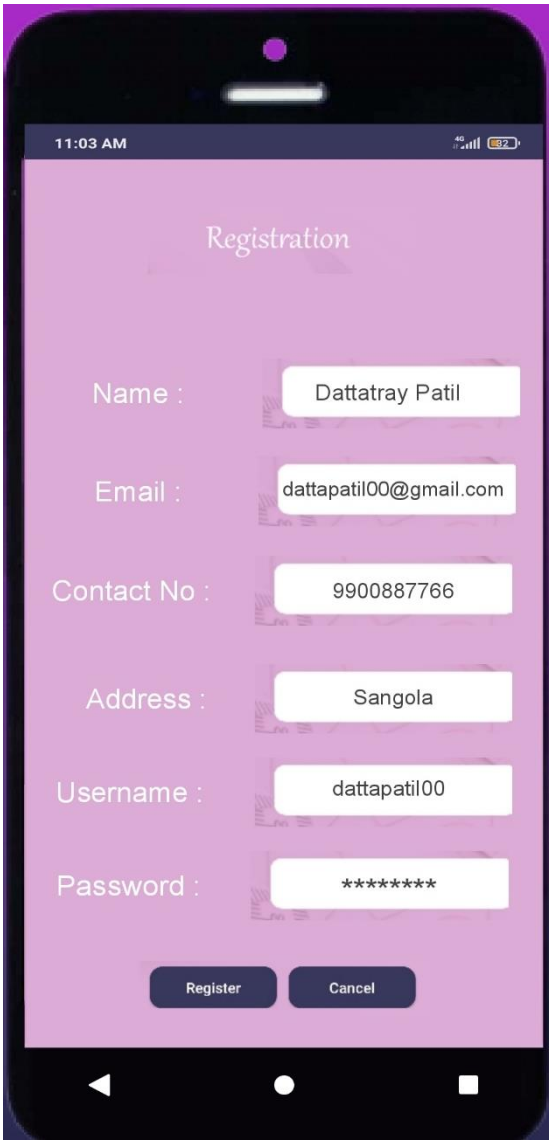
3.9 Sample Input and Output Screens

Splash Screen:

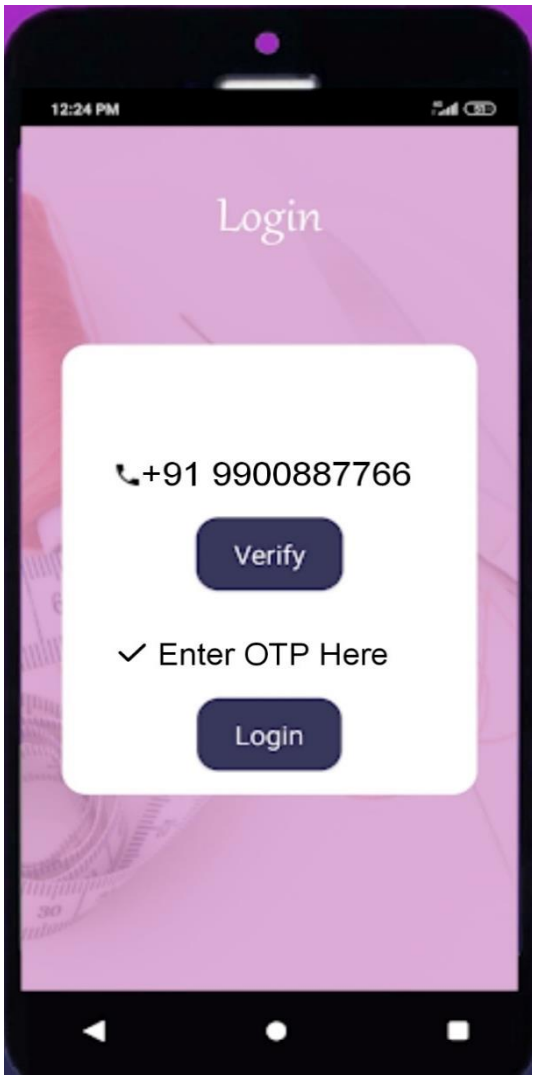
About Us:

Login:

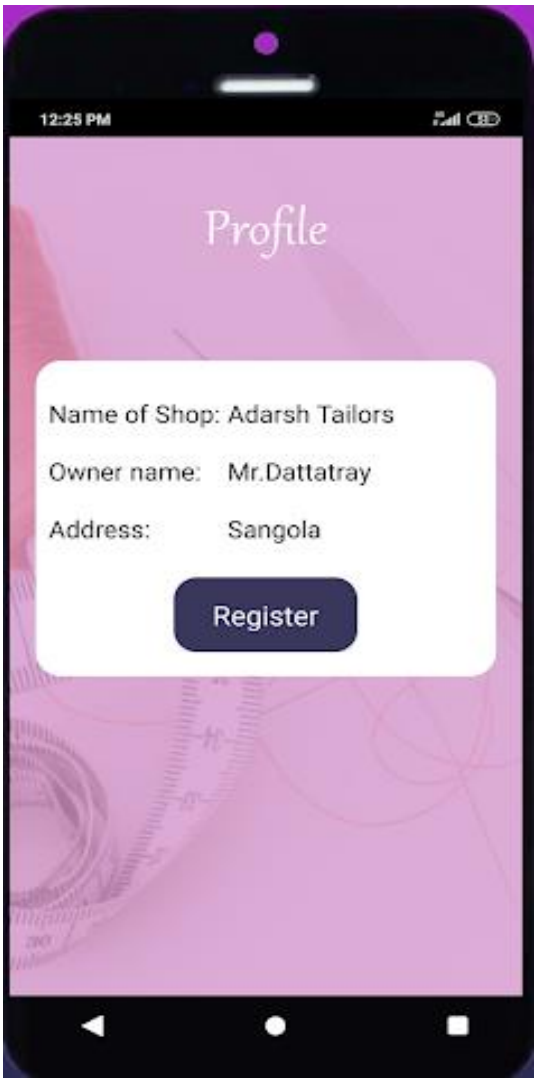
Registration:



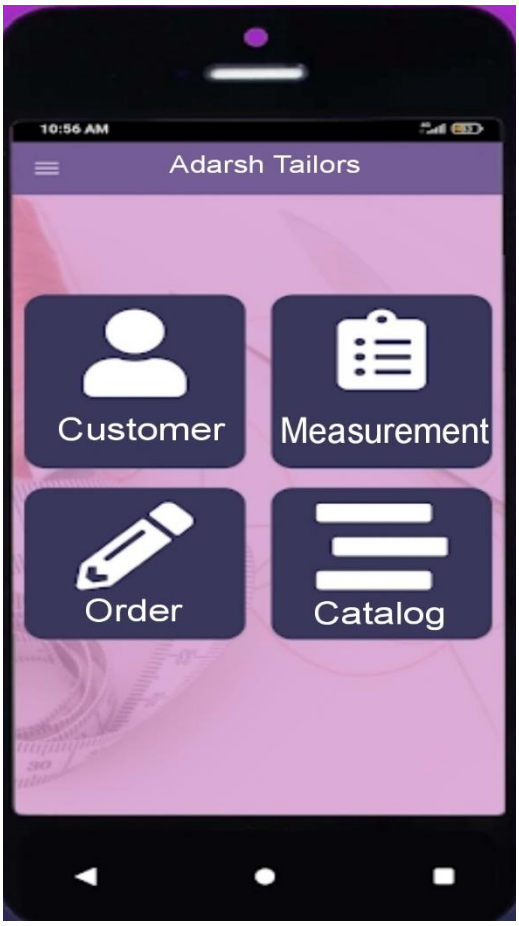
Secure Sign In With Phone No Verification:



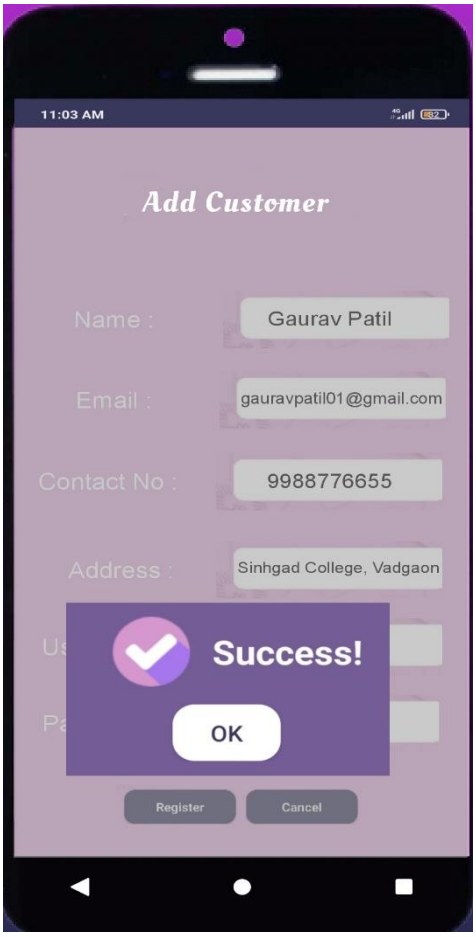
Tailor Profile:



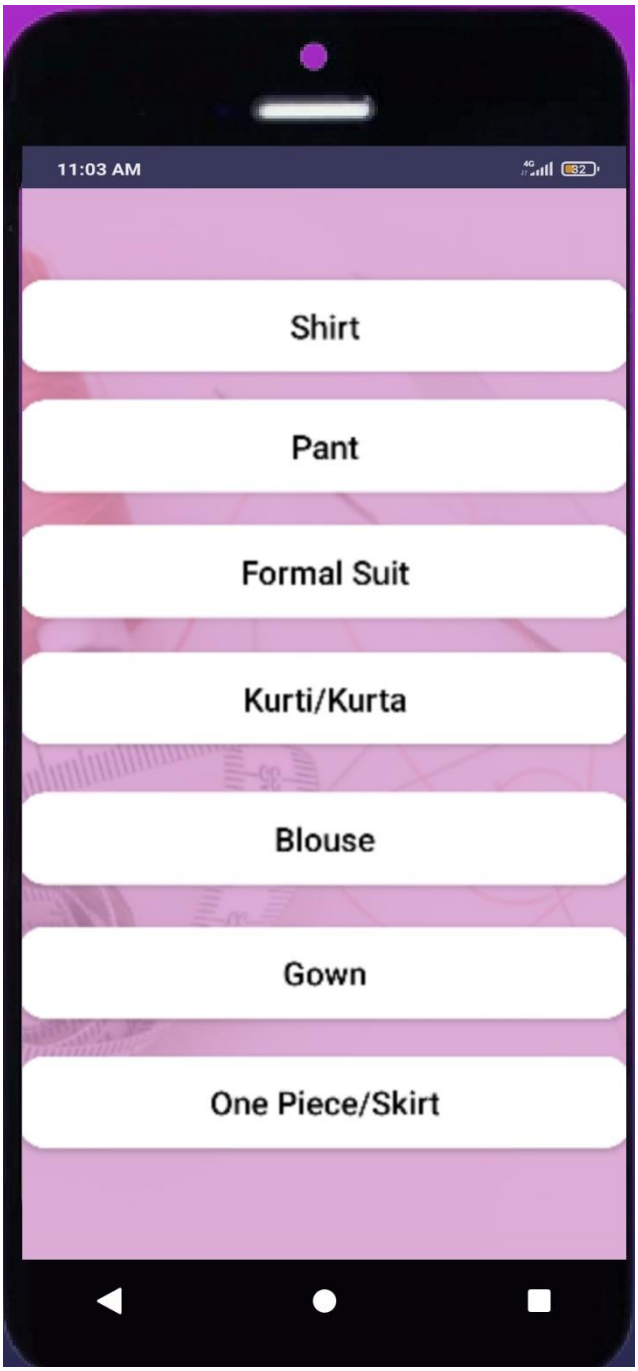
Dashboard:



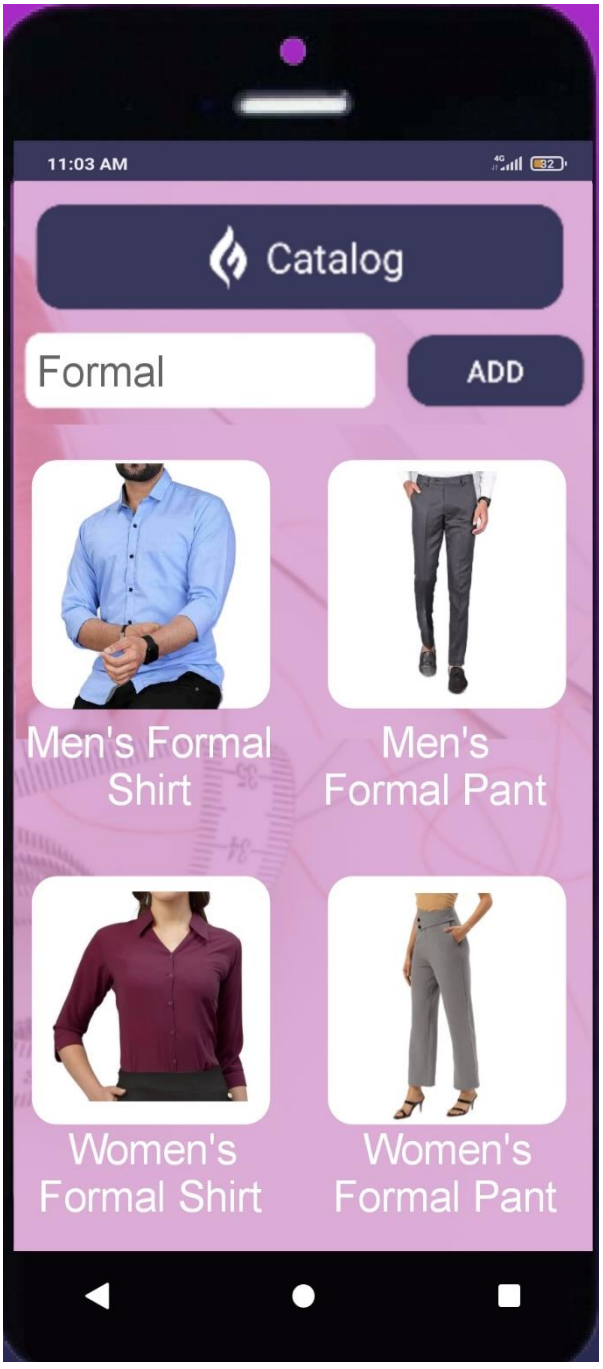
Adding Customer Info:



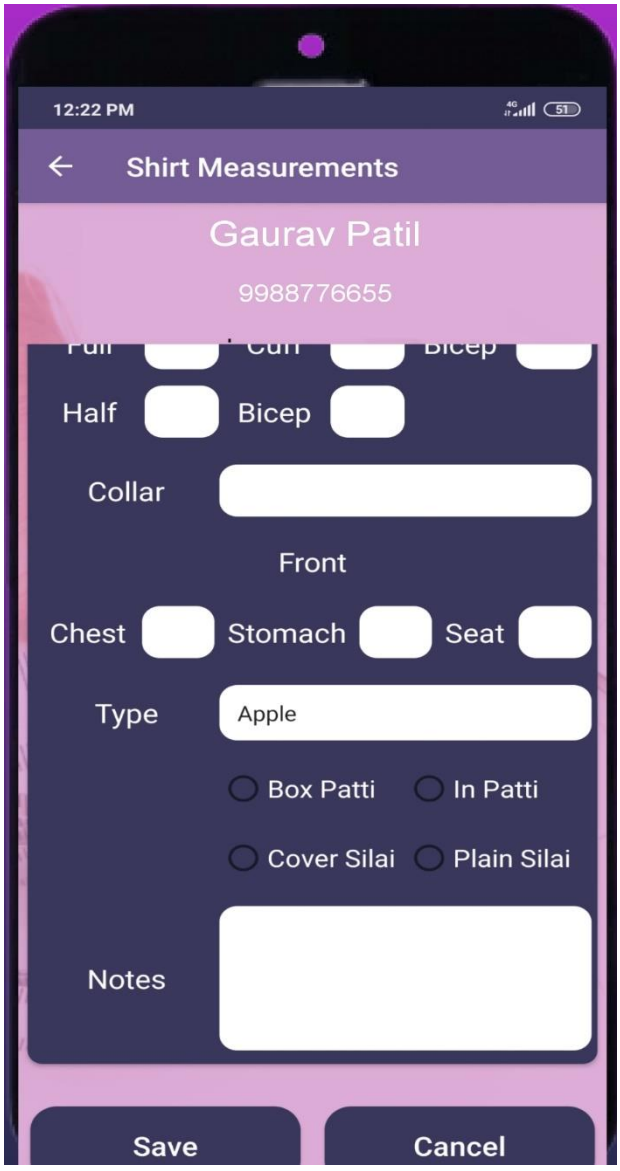
List of Product:



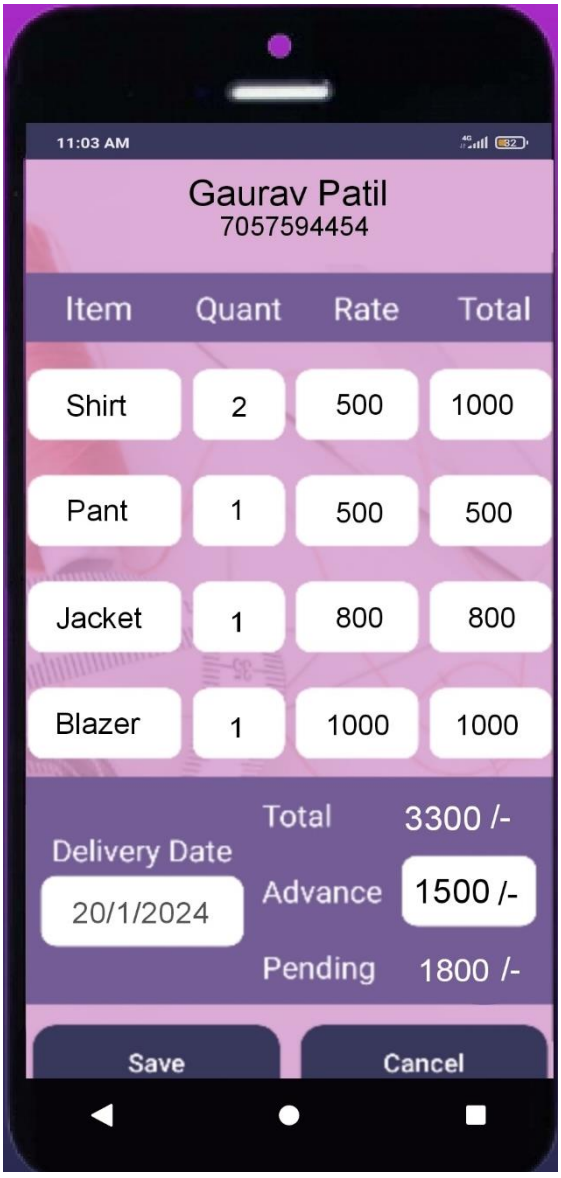
Catalog:



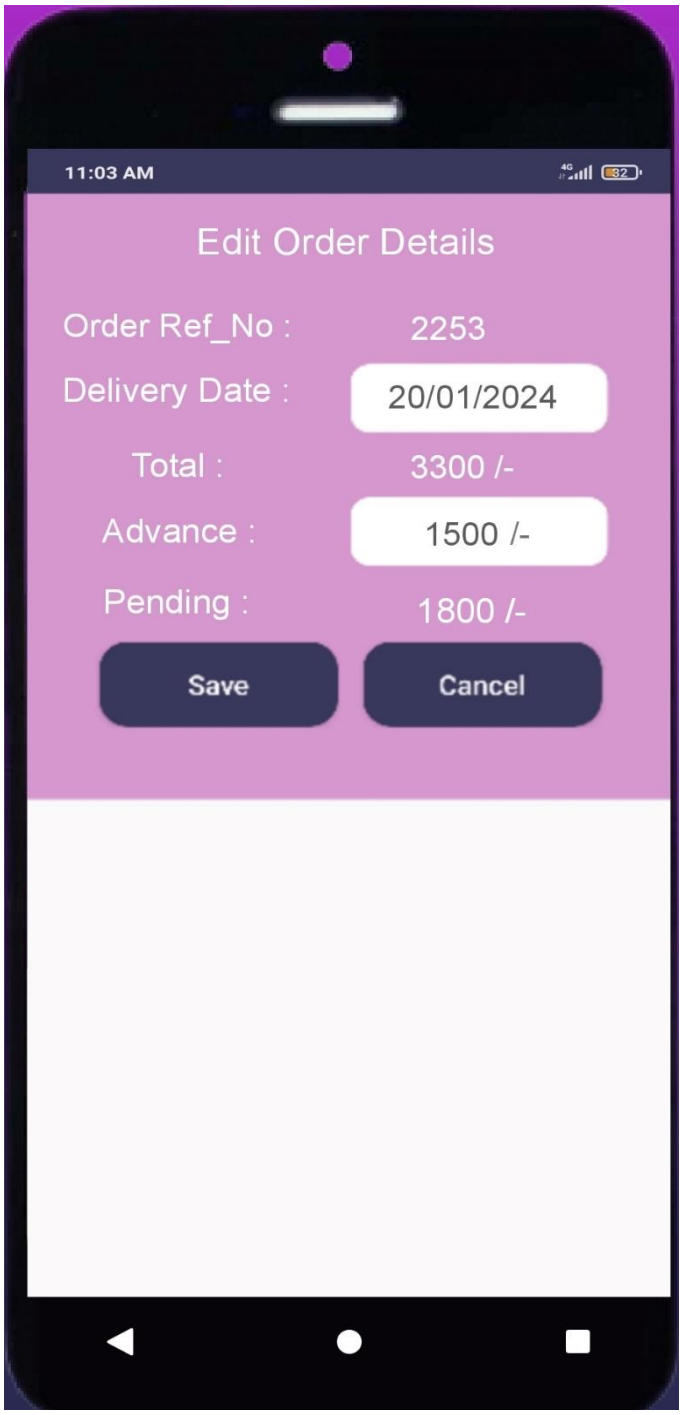
Measurement:



Order:

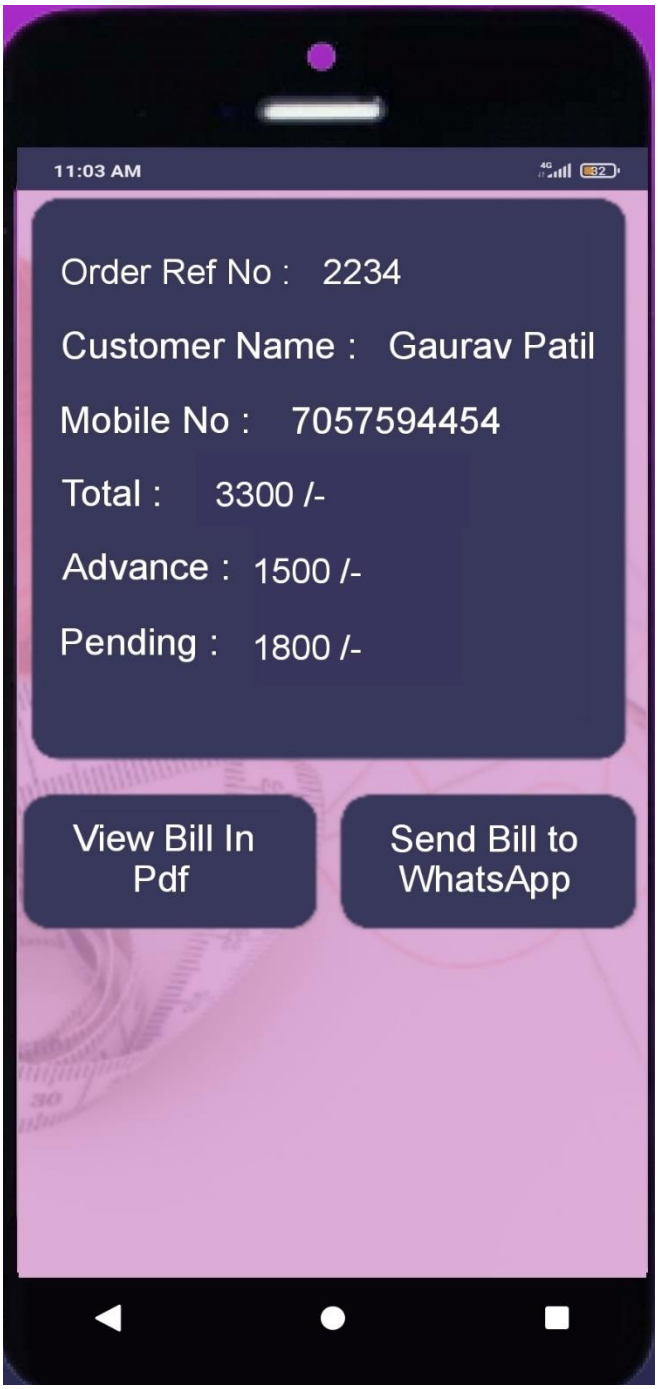


Edit Order Details:

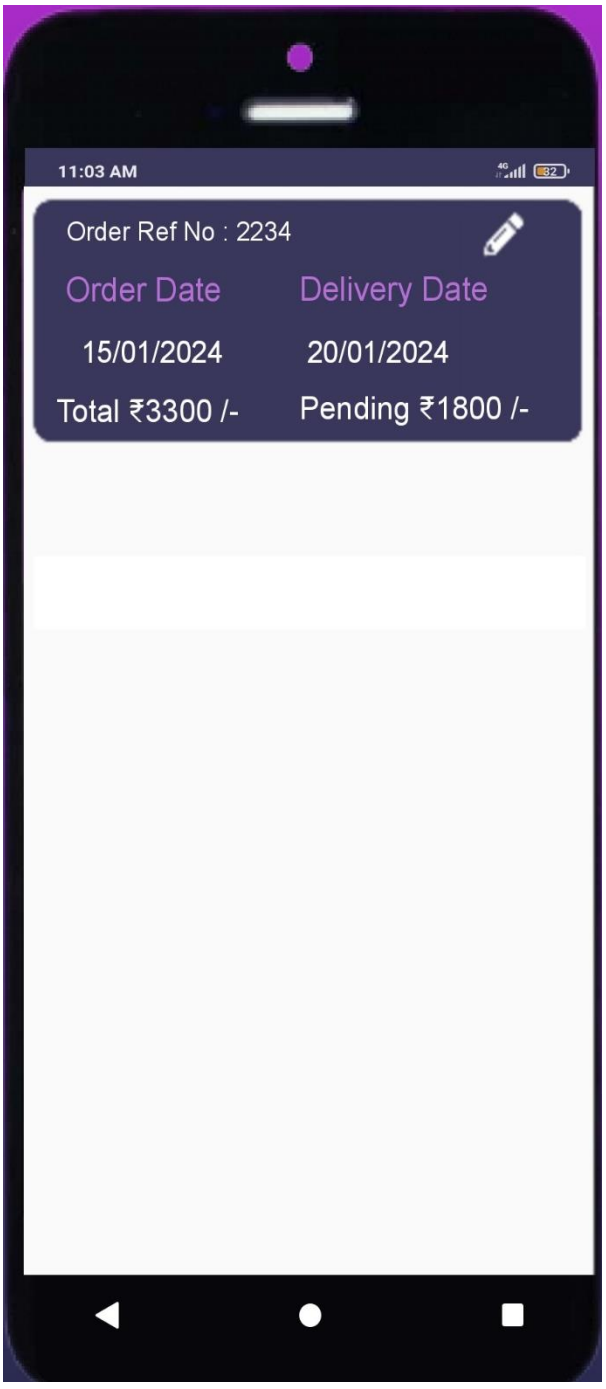


Payment:

Billing Receipt:



Delivery Status:



CHAPTER 4: CODING SAMPLE CODE

Activity_login.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/parent_layout"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".activities.LoginActivity"
    android:background="@drawable/bg">


    <TextView
        android:id="@+id/login_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="@color/colorWhite"
        android:layout_marginTop="32dp"
        android:fontFamily="@font/gabriola"
        android:text="@string/login_text"
        android:textSize="@dimen/heading_text"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

```
<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="32dp"
    android:layout_marginTop="64dp"
    android:background="@drawable/custom_card"
    android:orientation="vertical"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/login_text">

    <TextView
        android:id="@+id/text_mobile_sign_in"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:fontFamily="@font/roboto"
        android:text="@string/signin_text"
        android:textColor="@color/colorBlack"
        android:textSize="@dimen/primary_heading_text" />

    <LinearLayout
        android:id="@+id/layout_phone_login"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="20dp"
    android:layout_marginTop="30dp"
    android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginHorizontal="10dp"
        android:drawableLeft="@drawable/ic_phone"
        android:fontFamily="@font/roboto"
        android:text="@string/country_code"
        android:textColor="@color/colorBlack"
        android:textSize="@dimen/login_page_text" />

    <EditText
        android:id="@+id/edit_phone"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:hint="@string/phone_hint"
        android:inputType="phone"
        android:maxLength="10"
        android:textColor="@color/colorBlack"
        android:textColorHint="@color/colorHint"
        android:textSize="@dimen/login_page_text" />

</LinearLayout>
```

```
    <Button
        android:id="@+id/verify_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="10dp"
        android:background="@drawable/custom_button"
        android:fontFamily="@font/roboto"
        android:onClick="onClickLogin"
        android:text="@string/verify"
        android:textAllCaps="false"
        android:textColor="@color/colorWhite"
        android:textSize="@dimen/login_page_text" />

</LinearLayout>


<ProgressBar
    android:id="@+id/login_progress_bar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:visibility="gone" />

<LinearLayout
    android:id="@+id/layout_otp"
    android:visibility="gone"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
```

```
        <EditText
            android:id="@+id/edit_otp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginTop="30dp"
            android:drawableLeft="@drawable/ic_check"
            android:gravity="center"
            android:hint="Enter OTP here"
            android:inputType="number"
            android:maxLength="6"
            android:textColor="@color/colorBlack"
            android:textColorHint="@color/colorHint"
            android:textSize="@dimen/login_page_text" />

        <Button
            android:id="@+id/login_btn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginTop="10dp"
            android:background="@drawable/custom_button"
            android:fontFamily="@font/roboto"
            android:textColor="@color/colorWhite"
            android:textSize="@dimen/login_page_text" />
    </LinearLayout>
  </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

CHAPTER 5: LIMITATIONS OF SYSTEM

- **Technology Dependence:** The app's functionality relies on technology infrastructure, and any disruptions, server downtimes, or technical issues could impact the user experience and service delivery.

- **Internet Connectivity:** Users may face limitations if they have unreliable or no internet connectivity, as the app heavily relies on online features such as fabric selection, design customization, and order placement.

- **Device Compatibility:** The app's effectiveness is dependent on users having compatible devices. Older devices or those with outdated operating systems may not fully support all features, limiting accessibility for some users.

- **Tailor Adoption:** The success of the app depends on tailors adopting and actively using the platform. Resistance or slow adoption by tailors could limit the range of available services for customers.

- **Security Concerns**: There may be concerns regarding the security of user data, especially personal measurements and payment information. Addressing and mitigating potential security risks is crucial to maintaining user trust.

- **User Learning Curve:** Some users, especially those unfamiliar with technology, may find the app challenging to navigate. Providing sufficient user support and tutorials is essential to overcome this limitation.

- **Limited Customization Options:** The app's design interface and customization options may not cater to extremely intricate or unique design preferences, limiting the extent to which users can personalize their garments.

CHAPTER 6: PROPOSED ENHANCEMENTS

- **Tailor Ratings and Reviews:** Implement a rating and review system for tailors, allowing customers to provide feedback on the quality of service, craftsmanship, and overall satisfaction. This builds trust and assists other users in selecting reliable tailors.

- **Custom Fabric Upload:** Allow users to upload images of fabrics they own or have found elsewhere, expanding the fabric selection and customization options.

- **Advanced Fabric Visualization:** Enhance the fabric visualization feature by utilizing advanced imaging technology, providing users with a more realistic preview of how selected fabrics will look on the final garment.

- **Push Notifications for Order Updates:** Implement push notifications to keep users informed about the status of their orders, including updates on measurements, production, and delivery.

- **Customer Loyalty Program:** Introduce a loyalty program that rewards customers for repeat orders, referrals, and active engagement with the app. This encourages customer retention and engagement.

CHAPTER 7: CONCLUSION

- In conclusion, the FabricFit Android Tailoring App presents a transformative solution to modernize and elevate the traditional tailoring experience. With a user-centric approach, innovative features, and a commitment to efficiency, the app aims to revolutionize the way users interact with tailors and customize their garments. The integration of cutting-edge technologies, such as augmented reality, artificial intelligence, and blockchain, positions FabricFit as a pioneer in the online tailoring industry.

- The app addresses common limitations associated with traditional tailoring methods, offering a seamless and convenient platform for users to design, customize, and order garments. The proposed enhancements, including augmented reality fitting rooms, AI-driven style recommendations, and real-time collaboration features, further demonstrate the commitment to continuous improvement and user satisfaction.

- FabricFit not only streamlines the tailoring process but also fosters a sense of community and trust through features like tailor ratings, social media integration, and customer loyalty programs. The attention to security, privacy, and transparency, exemplified by blockchain implementation, instills confidence in users regarding their data and financial transactions.

- As FabricFit evolves, it has the potential to set new standards in the online tailoring industry, providing users with a unique and personalized tailoring experience. The app's success relies on its ability to adapt to user needs, incorporate emerging technologies, and maintain a user-friendly interface. FabricFit stands as a testament to the fusion of technology and fashion, ushering in a new era of convenience, accessibility, and innovation in the world of tailoring.

CHAPTER 8: BIBLIOGRAPHY

REFERENCE BOOKS

1. "Android Programming: The Big Nerd Ranch Guide" by Bill Phillips and Chris Stewart

2. Head First Android Development" by Dawn Griffiths and David Griffiths

3. "Android App Development For Dummies" by Michael Burton

4. "Mastering Firebase for Android Development" by Ashok Kumar S

5. "Firebase Essentials - Android Edition" by Neil Smyth

WEB REFERENCES

1) https://stackoverflow.com/

2) https://app.diagrams.net/

3) https://m2.material.io/develop/android

4) https://developers.google.com/maps/documentation/android-sdk/start

5) https://developer.android.com/