**Alpha Tailor**

##### Umair Zia Mehran Ahmed

051-21-135215 051-19-132006



Ms. Shaina Laraib

A Final Year Project Report is submitted in Partial Fulfillment of the Requirements for the Degree of

Bachelorof Science in Software Engineering

### Department of Computing & Technology Iqra University

**2025**

##### Certificate

We hereby accept the work contained in this report titled: ***Alpha Tailor***, as a confirmation to the required standards for the partial fulfillment of the degree of Bachelors of Science in Software Engineering

*Internal Examiner External Examiner*

*Project Supervisor Head of Department*

#### Declaration

We, Umair Zia and Mehran Ahmed, solemnly declare that the research work presented in our undergraduate project, as the case, titled **“Alpha Tailor”** is solely our original research work with no significant contribution from any other person or Generative AI Tools. Small contribution/help wherever taken has been duly acknowledged/cited and the complete project/thesis has been written by us in accordance with the latest plagiarism policy declared by HEC and our respective university in-line with the policy for use of Generative AI Tools.

We understand the zero-tolerance policy of the HEC and **IQRA University of Islamabad H-9 Campus**, towards plagiarism. Therefore, we, as Authors of the above titled project/thesis, declare that no portion of our project/thesis has been plagiarized and any material used as reference is properly referred/cited.

We undertake that if we are found guilty of any plagiarism in the above titled project even after award of undergraduate degree, the University reserves the right to withdraw/revoke our degrees and that HEC and the University have the right to publish our names on the HEC/University Website on which names of students are placed who submitted plagiarized projects.

**Umair zia 35215**

**Mehran Ahmed 32006**

#### Acknowledgements

I would like to express my sincere gratitude to my project supervisor, Madam Shaina Laraib for their invaluable guidance and unwavering support throughout the semester. Their expertise and insightful feedback significantly contributed to the success of this project. This project has been a rewarding learning experience, and I am thankful for the encouragement from my family and friends.

#### Abstract

Alpha Tailor is a mobile application designed to modernize the tailoring experience by providing users with a convenient, fully digital platform for ordering custom-fit garments. By allowing users to submit measurements online, browse tailor profiles, visualize designs on a 3D mannequin, and securely process payments, Alpha Tailor eliminates the need for in-person visits and simplifies the process of acquiring personalized clothing. Built with Flutter for cross-platform compatibility and powered by AI for automated measurement detection, the app caters to individuals with busy schedules or limited mobility, delivering a streamlined and accessible solution. Alpha Tailor is technically feasible, financially viable, and aims to set a new standard in digital tailoring through an intuitive, user-friendly experience.

Table of Contents

[Declaration 3](#_bookmark0)

[Acknowledgements 4](#_bookmark1)

[Abstract 5](#_bookmark2)

[Contents 6](#_bookmark2)

[Introduction 13](#_bookmark3)

* 1. [Overall Description 14](#_bookmark4)
     1. [Objectives 14](#_bookmark5)
     2. [Problem Description 14](#_bookmark6)
     3. [Methodology 14](#_bookmark7)
     4. [Product Scope 15](#_bookmark8)
     5. [User Classes and Characteristics 15](#_bookmark9)
     6. [Operating Environment 15](#_bookmark10)
     7. [Assumptions 16](#_bookmark11)
     8. [Dependencies 16](#_bookmark12)
  2. [External Interface Requirements 16](#_bookmark13)
     1. [User Interfaces 16](#_bookmark14)
     2. [Hardware Interfaces 17](#_bookmark15)
     3. [Software Interfaces 17](#_bookmark16)
     4. [Communications Interfaces 17](#_bookmark17)
  3. [User Registration and Authentication 18](#_bookmark18)
     1. [Functional Requirements 18](#_bookmark19)
     2. [Body Measurement Form 19](#_bookmark20)
     3. [Tailor Directory and Reviews 20](#_bookmark21)
     4. [Functional Requirements 20](#_bookmark22)
  4. [Order Placement and Payment Integration 20](#_bookmark23)
     1. [Functional Requirements 21](#_bookmark24)
     2. [3D Visualization of Custom Designs 21](#_bookmark25)
     3. [Functional Requirements 22](#_bookmark26)
     4. [Admin Portal for Service and Tailor Management 22](#_bookmark27)
     5. [Functional Requirements 22](#_bookmark28)
  5. Report Structure [22](#_bookmark28)

[Chapter2: Literature Review 26](#_bookmark29)

* 1. [Introduction 27](#_bookmark30)
  2. [Tailoring and Technology 27](#_bookmark31)
     1. [Traditional Tailoring Practices 27](#_bookmark32)
     2. [The Need for Online Tailoring Services 27](#_bookmark33)
     3. [Market Demand 27](#_bookmark34)
  3. [Innovative Features in Tailoring Apps 28](#_bookmark35)
     1. [3D Visualization 28](#_bookmark36)
     2. [Competitive Analysis 28](#_bookmark37)
  4. [Challenges in Online Tailoring Services 28](#_bookmark38)
  5. [Relevance of Flutter and Firebase 28](#_bookmark39)
     1. [Reliable Backend Solutions 29](#_bookmark40)
  6. [Unique Features for Alpha Tailor 29](#_bookmark41)
  7. [Recommendations 29](#_bookmark42)
  8. [Conclusion 30](#_bookmark43)

[Chapter 3: System Design 32](#_bookmark45)

* 1. [Introduction 33](#_bookmark46)
     1. [Purpose 34](#_bookmark47)
  2. [Design Considerations 34](#_bookmark48)
     1. [Assumptions 34](#_bookmark49)
     2. [Constraints 34](#_bookmark50)
     3. [Design Methodology 34](#_bookmark51)
     4. [Risks and Mitigation 34](#_bookmark52)
  3. [System Architecture 34](#_bookmark53)
     1. [Overview 34](#_bookmark54)
     2. [Subsystems 35](#_bookmark55)
     3. [Design Strategies 35](#_bookmark56)
  4. [Database Schema 35](#_bookmark57)
     1. [Data Migration 36](#_bookmark59)
  5. [High-Level Design 36](#_bookmark60)
  6. [Low-LevelDesign Modules 37](#_bookmark63)
  7. [User Interface Design 38](#_bookmark63)
     1. [Application Controls 38](#_bookmark64)
     2. [System Design 38](#_bookmark65)
     3. [Use Case Diagram 39](#_bookmark67)
     4. [Control Flow Diagram 40](#_bookmark69)
  8. [Class Diagram 41](#_bookmark71)
     1. [Sequence Diagram 42](#_bookmark73)
     2. [System Sequence Diagram 44](#_bookmark75)
     3. [State Diagram 44](#_bookmark77)

[3.9 Summary 46](#_bookmark79)

[Chapter 4: Implementation 47](#_bookmark80)

* 1. [Overview 48](#_bookmark81)
     1. [Key aspects of the implementation include 48](#_bookmark82)
  2. [Development Methodology 48](#_bookmark83)
     1. [The methodology consists of the following phases 48](#_bookmark84)
  3. [Tools and Technologies 48](#_bookmark85)
     1. [Key technologies and tools include 48](#_bookmark86)
  4. [Summary 49](#_bookmark87)
     1. [Login Screen 49](#_bookmark87)
     2. [Sign up Screen 50](#_bookmark89)
     3. [Home Screen 51](#_bookmark91)
     4. [App Splash Screen 52](#_bookmark93)
     5. [OnBoarding screens 52](#_bookmark95)

[Chapter 5: Testing 54](#_bookmark97)

* 1. [Testing Phone Number/Email Verification 56](#_bookmark99)
  2. [Testing New Order Creation (Tailoring) 56](#_bookmark101)
  3. [Testing Measurement Entry and Editing 56](#_bookmark103)
  4. [Summary 57](#_bookmark107)

[Chapter 6:Conclusions and Future Work 58](#_bookmark108)

* 1. [Contributions 59](#_bookmark109)
     1. [Contribution 2 59](#_bookmark110)
  2. [Findings 59](#_bookmark111)
  3. [Future Work 59](#_bookmark112)
     1. [Further System Designs 59](#_bookmark113)

## List of Figures

[Figure 3.5.1 High Level Diagram 36](#_bookmark61)

[Figure 3.5.2 Low level Diagram 37](#_bookmark62)

[Figure 3.5.3 System Architecture Design 38](#_bookmark66)

[Figure 3.5.4 Use Case Diagram 39](#_bookmark68)

[Figure 3.5.5 Control Flow Diagram 40](#_bookmark70)

[Figure 3.5.6 Class Diagram 41](#_bookmark72)

[Figure 3.5.7 Sequence Diagram 42](#_bookmark74)

[Figure 3.5.8 System Sequence Diagram 43](#_bookmark76)

[Figure 3.5.9 State Chart Diagram 44](#_bookmark78)

[Figure 4.3.1 Login Screen 48](#_bookmark88)

[Figure 4.3.2 Signup Screen 49](#_bookmark90)

[Figure 4.3.3 Design Screen 50](#_bookmark92)

[Figure 4.3.4 Splash Screen 51](#_bookmark94)

[Figure 4.3.5 Onboarding Screen 52](#_bookmark96)

## List of Table

[Table 2.1 Comparison Table 31](#_bookmark44)

[Table 3.4.1 Field & Relationships Table 36](#_bookmark58)

[Table 5.3.1 Signup Testing Process 56](#_bookmark98)

[Table 5.3.2 Test Phone Number Verification 57](#_bookmark100)

[Table 5.3.3 Ordering Creation Testing 57](#_bookmark102)

[Table 5.3.4 Test Measurement & Editing 58](#_bookmark104)

[Table 5.3.5 Delivery Date/Time Selection 59](#_bookmark105)

[Table 5.3.6 Test Nearest Tailor Shop Location 59](#_bookmark106)

**Chapter 1 Introduction**

#### Introduction

Alpha Tailor is an innovative mobile application designed to modernize and simplify the traditional tailoring process by providing a fully digital, user-friendly platform for ordering custom-fit clothing. As demand for personalized garments grows, Alpha Tailor addresses the challenges of conventional tailoring—such as time-consuming in-person fittings—by allowing users to submit measurements and customization preferences online, all from the convenience of their homes. Built using Flutter for cross-platform compatibility, the app offers advanced features like interactive measurement tutorials, AI-driven Automatic Measurement Detection, and secure payment options. Users can browse a wide range of tailoring options, review tailor profiles with ratings, and visualize their selected styles on a 3D mannequin. Tailored for individuals with busy schedules or mobility limitations, Alpha Tailor transforms tailoring into a streamlined and accessible digital experience, meeting modern demands for convenience, personalization, and efficiency.

##### Overall Description

Alpha Tailor is a mobile app designed to simplify custom tailoring by offering a fully digital platform for ordering personalized garments. Users can submit measurements, explore tailoring options, review profiles, and visualize styles in 3D—all from home. Built with cross-platform Flutter and AI-driven measurement detection, the app ensures a seamless, accessible experience. Alpha Tailor caters to those with busy schedules or limited mobility, transforming traditional tailoring into a convenient digital service.

##### Objectives

* + - * ***Alpha Tailor*** Offer a fully online tailoring service to avoid in-person visits.
      * Simplify measurement submission is installed in ***Alpha Tailor*** with easy tutorials and AI- based auto-measurement.
      * Allowusers to browse and personalize garment options like fabric and design.
      * Providea list of tailors with ratings and reviews for easier selection.
      * Enable 3D previews of chosen designs for a realistic view.
      * Ensure secure online payments for smooth order management.

##### Problem Description

Traditional tailoring requires time-consuming in-person visits for measurements and fittings, which can be inconvenient or inaccessible for many. Few digital solutions simplify the process of ordering custom-fit clothing, leaving a gap for those seeking convenience and personalization. Alpha Tailor solves this by offering a fully online tailoring experience through a mobile app, allowing users to connect with tailors, submit measurements, and manage orders from home. This

approach makes custom tailoring accessible, efficient, and hassle-free.

##### Methodology

* + - * Conductmarket research to understand customer needs, preferences, and trends in tailoring services to guide app development and marketing strategies.
      * Develop the ***Alpha Tailor*** app using Flutter for cross-platform compatibility, ensuring smooth performance on both Android and iOS.
      * Create a user-friendly interface with easy navigation for submitting measurements, selecting designs, and choosing tailors.
      * Implement secure online payment options for seamless transactions and order management.
      * Integrate AI-based automatic measurement detection to enhance user convenience and accuracy.
      * Collect user feedback and update the app with new features and improvements based on customer needs and technological advancements.
      * Promotethe app through targeted marketing strategies to increase awareness and drive user engagement.

##### Product Scope

***Alpha Tailor*** is a mobile app that simplifies the custom tailoring process by offering a fully digital platform for submitting measurements, browsing designs, and connecting with tailors. It provides a convenient solution for individuals with busy schedules or mobility challenges, allowing them to order personalized clothing without in-person visits. The app aligns with the trend of digital convenience in the fashion industry, aiming to enhance customer satisfaction and streamline tailoring services. Alpha Tailor makes custom tailoring more accessible, efficient, and user- friendly for a broader audience.

##### User Classes and Characteristics

The primary users of Alpha Tailor will be individuals seeking custom-tailored clothing. They will benefit from the app’s easy measurement submission, design selection, and secure order management. Tailors will also use the app to manage customer orders and provide services. Administrators will have access to the back-end to manage tailor listings, customer interactions,

and track app performance.

##### Operating Environment

* + - * **Hardware Platform:** Works on smartphones and tablets, supporting both Android and iOS devices.
      * **Operating System:** Compatible with Android and iOS.
      * **Mobile Development Framework:** Built using Flutter for cross-platform compatibility, ensuring smooth performance on both Android and iOS.
      * **Backend Technologies:** Utilizes Firebase for secure and real-time database management.
      * **Internet Connectivity:** Requires an internet connection for users to submit measurements, place orders, and receive updates.
      * **Security Measures:** Uses secure authentication, including password encryption, to protect user data and ensure privacy.

##### Assumptions

* + - * **Stable Internet Connectivity:** It is assumed that users will have access to a stable internet connection to submit measurements, place orders, and receive updates in real-time.
      * **Device Compatibility:** The app is assumed to be compatible with a variety of smartphones and tablets, ensuring accessibility for a broad range of customers.
      * **Accurate Measurements:** It is assumed that users will provide accurate measurements or use the AI-based measurement detection feature for precision.

##### Dependencies

* + - * **Flutter Framework:** The app’s development depends on the Flutter framework for cross- platform compatibility, ensuring it runs smoothly on both Android and iOS.
      * **Firebase for Backend:** Firebase is used for real-time database management, user authentication, and secure data storage, making it essential for the app's functionality.
      * **AI-based Measurement Detection:** The app's measurement detection feature depends on AI algorithms for accurate body measurement recognition from images.

##### External Interface Requirements

This section specifies the necessary interfaces for user interaction and system integration.

##### User Interfaces

* + - * **Homepage:** Showcases available tailoring services, dress categories, and featured designs.
      * **Measurement Form:** Simple, interactive form for users to input or upload their measurements.
      * **Product Catalog:** Displays a variety of custom clothing options with filters for easy navigation (e.g., dress type, fabric, design).
      * **Tailor Profiles:** List of available tailors with ratings, reviews, and expertise.
      * Design Customization: Allows users to visualize their chosen designs on a 3D mannequin or overlay their own photo.
      * **Order Management:** Section for users to track, manage, and update their orders.
      * **Feedback and Support:** Provides customer support options, FAQs, and a platform for submitting feedback.

##### Hardware Interfaces

* + - * Works on common devices like smartphones and tablets running Android and iOS operating systems. Users can easily access the app on their mobile devices, ensuring a convenient tailoring experience.
      * Responsive design adapts to different screen sizes, providing a smooth and consistent user experience across various devices.
      * Accessible to all users, including those with mobility challenges, enabling them to place orders and submit measurements from the comfort of their homes.
      * Users can conveniently browse, select designs, and manage their orders from their preferred mobile device, enhancing accessibility and satisfaction.

##### Software Interfaces

* + - * Utilizes Firebase for real-time database management, user authentication, and cloud storage of product information and user data.
      * Employs the Flutter framework for cross-platform development, allowing seamless operation on both Android and iOS devices.
      * Connectsto inventory management systems to provide up-to-date product availability and synchronize stock levels in real-time.

##### Communications Interfaces

* + - * **Mobile App Interaction:** The primary interface for users to interact with the e Commerce app will be through mobile devices. The app will utilize secure HTTPS protocols for data

exchange between the app and the server, ensuring encrypted communication over the internet.

* + - * **Real-Time Data Sync**: The application will leverage real-time data synchronization capabilities through Firebase, allowing for immediate updates to product availability and user interactions without requiring manual refreshes.
      * **User Authentication:** The app will implement secure authentication protocols to manage user login sessions, ensuring that all user credentials and personal information are transmitted securely.

##### Use Registration and Authentication Description and Priority

This feature allows users to create an account and securely log in to the Alpha Tailor app. The priority of this feature is **High**, as it is critical for personalizing user experience, managing orders, and securing payment data. The registration process should be simple and straightforward to encourage adoption.

* **Benefit:** 9
* **Penalty:** 3
* **Cost:** 5
* **Risk:** 4

**Stimulus/Response Sequences**

* **User Action:** User selects “Sign Up” or “Log In” option.
* **System Response:** Displays form to enter email, phone number, or use social media login.
* **User Action:** User submits information or authenticates via social login.
* **System Response:** The system checks the input and authenticates the user, providing access to the main app screen.

##### Functional Requirements

* + - * **REQ-1:** The app must support account creation via email, phone number, and social media logins (Google, Facebook).
      * **REQ-2**: The app must securely store user credentials and allow easy recovery of forgotten passwords.
      * **REQ-3:** The app must authenticate users via OTP or secure login methods.
      * **REQ-4:** The app should handle invalid inputs, prompting users with appropriate error messages (e.g., “Invalid email address”).

##### Body Measurement Description and Priority

This feature allows users to input body measurements for tailored clothing orders. It is a **High** priority feature, as the accuracy of the measurements directly impacts the quality of the service provided.

* + - * **Benefit:** 10
      * **Penalty:** 5
      * **Cost:** 4
      * **Risk:** 6

**Stimulus/Response Sequences**

* + - * User Action: **User selects a clothing category (e.g., dress, shirt, pants).**
      * System Response: **Theapp displays a digital measurement form specific to the selected category.**
      * User Action: **User enters the measurements via text fields or selects options provided.**
      * System Response: **The system confirms the measurements and allows users to review before submitting.**

**Functional Requirements**

* + - * **REQ-1:** The app must provide a dynamic form that adjusts based on the clothing category selected.
      * **REQ-2:** The app should allow for manual input of measurements (e.g., waist size, sleeve length) and offer visual aids for guidance.
      * **REQ-3:** The app should include anoption for AI-based Automatic Measurement Detection by uploading a picture.
      * **REQ-4:** The system must validate input measurements to ensure they are within a reasonable range (e.g., waist size should be between 20-60 inches).
      * **REQ-5:** The app should allow users to review and modify measurements before final submission.

##### Tailor Directory and Reviews

**Description and Priority**

This feature enables users to browse through a list of available tailors, view their profiles, and read customer reviews. It is a **Medium** priority feature but still crucial for creating trust and offering a choice of services.

* + - * **Benefit:** 7
      * **Penalty:** 4
      * **Cost:** 3
      * **Risk:** 5

**Stimulus/Response Sequences**

* + - * **User Action:** User taps on the “Find Tailors” button in the app.
      * **System Response:** Displays a list of available tailors, sorted by location or rating.
      * **User Action:** User taps on a specific tailor's profile.
      * **System Response:** The system displays the tailor’s profile, reviews, services offered, and pricing.

##### Functional Requirements

* + - * **REQ-1**: The app must display a curated list of tailors, sorted by rating, distance, or specialty.
      * **REQ-2:** The app must allow users to view detailed profiles of each tailor, including services, pricing, and user reviews.
      * **REQ-3:** The app should allow users to filter tailors based on specific criteria (e.g., availability, service type, location).
      * **REQ-4:** The system must integrate a review and rating system where users can rate their experiences and leave feedback.

##### Order Placement and Payment Integration Description and Priority

This feature allows users to place orders for their tailored garments and make secure payments online. It is a **High** priority feature, as seamless payment processing is vital for the app’s functionality.

* **Benefit:** 10
* **Penalty:** 6
* **Cost:** 7
* **Risk:** 5

**Stimulus/Response Sequences**

* **User Action:** User selects the clothing item and enters measurements.
* **System Response:** The system displays the total price for the garment.
* **User Action:** User proceeds to payment, selecting a preferred payment method (credit card, wallet, etc.).
* **System Response:** The system processes the payment securely and confirms the order.

##### Functional Requirements

* + - * **REQ-1:** The app must offer multiple payment methods, such as credit/debit cards, Stripe, or wallet integrations.
      * **REQ-2:** The app should securely handle payment data with encryption protocols.
      * **REQ-3:** The app must display the order summary, including garment type, measurements, price, and payment method, before confirming the order.
      * **REQ-4:** The system should notify users of payment success or failure, and confirm the order via email or in-app notifications.

##### 3D Visualization of Custom Designs Description and Priority

This feature allows users to visualize their custom designs on a 3D mannequin or overlay their

photos to see how the garment would look on them. It is a **Medium** priority feature, enhancing the user experience with interactive design features.

* + - * **Benefit:** 8
      * **Penalty:** 3
      * **Cost:** 6
      * **Risk:** 4

**Stimulus/Response Sequences**

* + - * **User Action:** User selects a clothing design and customizes it.
      * **System Response:** The app provides a 3D view of the selected design.
      * **User Action:** User taps to overlay their own image or adjusts the design.
      * **System Response:** The app updates the visual display with the user’s adjustments.

##### Functional Requirements

* + - * **REQ-1:** The app must provide a 3D view of selected clothing designs.
      * **REQ-2:** The app should allow users to overlay their own photos for a more realistic preview.
      * **REQ-3:** The system must support basic customization options (e.g., color, fabric) with real-time updates to the 3D view.

##### Admin Portal for Service and Tailor Management Description and Priority

This feature enables administrators to manage available tailors, service listings, and customer orders. It is a **High** priority feature to ensure smooth operations and updates for users.

* + - * **Benefit:** 9
      * **Penalty:** 4
      * **Cost:** 6
      * **Risk:** 5

**Stimulus/Response Sequences**

* + - * **Admin Action:** Admin logs into the portal.
      * **System Response:** Displays the admin dashboard with options to manage tailors, services, and orders.
      * **Admin Action:** Admin updates the list of available tailors or services.
      * **System Response:** The system updates the mobile app’s data in real-time.

##### Functional Requirements

* + - * **REQ-1:** The admin portal must allow easy management of tailors, services, and user orders.
      * **REQ-2:** The system should provide real-time updates to the mobile app when any changes are made in the admin portal.
      * **REQ-3:** The app must include a feedback management system where admins can respond to customer reviews.

##### Performance Requirements

* + - * **Response Time**: The app should respond to user actions (e.g., navigating between screens, submitting measurements) within 2 seconds for an optimal user experience.
      * **Load Time**: The app's initial load time should not exceed 5 seconds to avoid user drop- off.
      * **Scalability**: The system should support at least 10,000 concurrent users with minimal degradation in performance.
      * **Data Syncing**: Measurements and orders should sync with the backend in real-time or within 5 seconds of submission to ensure up-to-date information.
      * **Error Handling**: Any errors related to measurements, order processing, or payment should be handled within 3 seconds and display a user-friendly message.
      * **Availability**: The app must maintain 99.9% uptime, ensuring continuous availability for users, especially during peak hours.

##### Safety Requirements

* + - * **Data Encryption**: Sensitive personal and payment data, such as body measurements and credit card information, must be encrypted both at rest and in transit (e.g., using AES 256 encryption).
      * **Access Control**: User data must only be accessible to authenticated users and authorized backend processes. Unauthorized access attempts should trigger alarms and be logged for auditing.
      * **Physical Safety**: The app should not allow actions or requests that may result in physical harm to users (e.g., inaccurate measurements that could lead to incorrect garment sizes).
      * **Compliance with Regulations**: The app must adhere to relevant data privacy laws (e.g., GDPR, CCPA) to ensure safety and protection of user data.

##### Security Requirements

* + - * **Authentication**: Users must log in using a secure authentication method (e.g., email/password, social login, or two-factor authentication).
      * **Authorization**: Role-based access control (RBAC) must be implemented to ensure that users can only access appropriate functionalities based on their role (e.g., customer, admin, or tailor).
      * **Payment Security**: All transactions must be processed via a secure payment gateway (e.g., Stripe, PayPal) and comply with PCI DSS standards for handling payment data.
      * **Data Privacy**: Users’ personal data (e.g., measurements, address, payment details) must be stored and managed in compliance with relevant privacy regulations (e.g., GDPR).
      * **Session Management**: The app must automatically log out users after 15 minutes of inactivity to prevent unauthorized access to accounts.
      * **Malware Protection**: The app should not allow the execution of malicious code and should protect against common threats (e.g., man-in-the-middle attacks, phishing).

##### Software Quality Attributes

* + - * **Usability:** The app should have a simple, intuitive interface with a smooth user journey from measurement submission to order completion. User feedback should be collected regularly to ensure usability improvements.
      * **Reliability:** The app should be able to handle expected and unexpected user inputs without crashing or malfunctioning. The app should also automatically recover from any system failure or interruption.
      * **Maintainability:** The app should be designed with modular components to ensure that new features and updates can be integrated easily without affecting other parts of the system.
      * **Interoperability:** The app should be compatible with the latest versions of both Android and iOS operating systems, and it should work seamlessly across various screen sizes and device types.
      * **Testability:** The app should include automated unit tests, integration tests, and end-to-end tests to ensure that new features and updates do not introduce defects.
      * **Portability:** The app should be portable across different Android and iOS devices without

requiring significant changes to the underlying codebase. It should adapt to various screen sizes and resolutions.

* + - * **Scalability:** The app should support scaling up to accommodate increasing user traffic, with backend services able to handle growing data and requests efficiently.
      * **Robustness:** The app should handle error conditions gracefully, providing meaningful error messages and fallbacks when issues arise (e.g., network failure, payment issues).
      * **Adaptability:** The app should allow easy updates for additional tailoring services, measurement features, or UI customizations based on user feedback and market trends.
      * **Availability:** The app should be designed to ensure high availability, even under high traffic conditions, with no downtime exceeding 5 minutes per month for maintenance or updates.

These nonfunctional requirements focus on creating a secure, efficient, and user-friendly experience for both customers and tailors, ensuring the app meets industry standards and user expectations.

#### Report Structure

##### Chapter 1: Introduction

This chapter explains the core idea of Alpha Tailor, which aims to modernize traditional tailoring by offering an online platform. Users can submit measurements and design suits from home, with an overview of project goals, challenges, and scope.

##### Chapter 2: Literature Review

In this chapter, we looked at other tailoring apps that already exist. We found that many of them are missing features like personalized design, live previews, and easy measurement options. Our app Alpha Tailor fills these gaps by using better technology.

##### Chapter 3: System Design

This part explains how the app is designed technically. It shows the system’s structure and main parts of modules. We also explained how data moves in the system and how the database is built, using diagrams. The connection between the front-end what users see and the back-end how it works behind the scenes is also discussed.

##### Chapter 4: Implementation

This chapter tells how we actually built the app using Flutter and Firebase. It explains the technical work like making 3D previews, checking user input with custom forms, and connecting everything with the backend. We also talked about the problems we faced during development and how we fixed them.

##### Chapter 5: Testing

In this chapter we explain how we tested the app to make sure it works well. We did unit testing testing small parts, UI testing testing the user interface, and tested the full process between tailor and customer. The results showed whether the app was working correctly or not.

##### Chapter 6: Conclusion and Future Work

This chapter explains how Alpha Tailor made tailoring easier by helping users take measurements, place orders, and manage profiles online. Both users and tailors found the app useful because it saved time and reduced mistakes. In the future, features like AI-based measurements, language options, mobile apps, and a tailor marketplace will be added.

# Chapter2: Literature Review

##### Introduction

The tailoring industry has traditionally relied on in-person visits for measurements, fittings, and garment customization. However, advancements in mobile technology and online platforms have transformed this space, making tailoring services more accessible and convenient. Research highlights the growing demand for digital solutions in tailoring, driven by an increasing preference for personalized clothing and busy lifestyles. Applications like **Alpha Tailor** aim to bridge the gap between traditional methods and modern convenience, addressing challenges such as accessibility, time constraints, and personalization through innovative technology.

##### Tailoring and Technology

##### Traditional Tailoring Practices

Historically, tailoring has been defined by precise manual measurements and fittings. While this method is reliable, it often requires multiple appointments and significant time investment.

Studies reveal that consumers frequently cite these factors as barriers, creating a demand for alternatives that maintain accuracy while offering convenience.

##### Rise of Digital Platforms

Digital platforms have revolutionized tailoring by enabling remote services. Mobile applications have introduced features such as 3D visualization, virtual fitting rooms, and AI-based tools, which enhance the user experience. Alpha Tailor capitalizes on this trend by integrating advanced technologies, offering virtual measurement options, and realistic previews of custom designs.

##### The Need for Online Tailoring Services

* + 1. **Accessibility and Convenience**

Online tailoring eliminates the need for physical visits, making it ideal for individuals with mobility challenges or busy schedules. Alpha Tailor’s AI-powered tools provide users with accessible solutions, enabling remote measurements and seamless tailoring services from the comfort of their homes.

##### Market Demand

Demand for personalized clothing has surged, particularly among urban customers seeking unique designs and perfect fits without in-person fittings. According to DEF Analytics, the global market

for online tailoring services has grown by 69% in the past five years. By targeting this niche, Alpha Tailor positions itself as a modern, user-friendly platform for custom tailoring.

##### Innovative Features in Tailoring Apps

* + 1. **AI-Based Measurement Tools**

AI-driven tools are revolutionizing tailoring by ensuring accuracy in remote measurements. Alpha Tailor integrates AI and computer vision technology to offer precise, reliable measurement solutions, addressing a key challenge in online tailoring services.

##### 3D Visualization

Research shows that 3D previews enhance customer satisfaction by providing realistic insights into garment designs. Alpha Tailor incorporates this feature, offering hyper-personalized visuals using 3D avatars or overlaid images of users.

##### Competitive Analysis

Several apps provide features aligned with **Alpha Tailor’s** goals:

* + - * **CloudTailor**: Personalized fashion solutions with online measurements and global delivery.
      * **Sartoro**: Custom suits with interactive portals for measurements and adjustments.
      * **Zyod**: Casual and formal wear customization with user-friendly interfaces.
      * **iTailor**: Detailed garment customization, including fabric, style, and thread choices.
      * **eShakti**: Ready-to-wear clothing with tailoring options for perfect fits.

##### Challenges in Online Tailoring Services

Online tailoring services face hurdles related to measurement accuracy, trust, and user adoption. Studies highlight the importance of tutorials, user feedback mechanisms, and visual clarity in addressing these concerns. Alpha Tailor incorporates these elements through AI-powered guidance, interactive tutorials, and a robust feedback system to enhance user trust and satisfaction.

##### Relevance of Flutter and Firebase

* + 1. **Cross-Platform Development**

Flutter enables efficient cross-platform app development, ensuring compatibility with Android and iOS. Research underscores its role in reducing development costs and time, making it a preferred

choice for Alpha Tailor.

##### Reliable Backend Solutions

Firebase provides robust authentication, scalability, and real-time database support. Its integration with Flutter ensures secure and reliable performance, essential for the seamless operation of **Alpha Tailor**.

##### Unique Features for Alpha Tailor

To differentiate itself, **Alpha Tailor** introduces the following innovative features:

1. **AI-Powered Virtual Tailor**: A step-by-step AI assistant for real-time measurement validation, ensuring precision.
2. **3D Virtual Fitting Room**: Personalized avatars with realistic overlays of selected designs.
3. **Eco-Friendly Options**: Sustainable fabric choices and eco-friendly packaging to appeal to environmentally conscious users.
4. **Tailor Live Chat/Consultation**: Interactive sessions for personalized service and design discussions.
5. **Group Order Management**: A streamlined workflow for bulk orders, ideal for weddings and corporate needs.
6. **Style Recommendations with AI**: Tailored suggestions based on user preferences, events, and fabric choices.
7. **Seamless Returns and Adjustments**: Hassle-free returns with home pickup for resizing or adjustments.
8. **In-App Events and Rewards**: Seasonal contests and loyalty programs to foster engagement.
9. **Offline-to-Online Support**: Partnerships with local tailors for doorstep measurement services.
10. **Voice-Assisted Navigation**: Hands-free navigation for enhanced accessibility, catering to users with disabilities.

##### Recommendations

* + - * **Dynamic Pricing**: Introduce promotions during festive and wedding seasons to attract

bulk orders.

* + - * **Inclusivity**: Expand target audience through gender-neutral marketing.
      * **Visualization Quality**: Prioritize high-quality 3D previews to enhance user trust.
      * **Comprehensive Tutorials**: Support AI tools with interactive guides to minimize errors.
      * **Scalability Testing**: Ensure robust backend infrastructure to handle peak user loads.

##### Conclusion

The literature underscores a shift from traditional tailoring methods to innovative, technology- driven solutions. By combining the precision of traditional tailoring with modern conveniences like AI, 3D visualization, and eco-friendly options, **Alpha Tailor** addresses market demands while overcoming industry challenges. These unique features and strategic refinements position **Alpha Tailor** as a leading solution in the evolving online tailoring industry, promising an exceptional user experience.

**Comparison Table:**

**Table** **2.1 Comparison Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature/Aspect** | **Our Project** | **Tailor Mate** | **Eazy Tailor** |
| Measurement Assistance |  |  |  |
| Target Audience |  |  |  |
| Order Visualization |  |  |  |
| Garment Customization |  |  |  |
| Payment Gateway |  | ✗ |  |
| Tailor Directory |  |  |  |
| Scalability & Backend |  | ✗ |  |
| Specialized Services |  |  |  |
| Customer Feedback |  |  |  |

# Chapter 3: System Design

#### Introduction

The system design defines the structure, components, and implementation strategy for the Alpha Tailor application. Its focus is on creating a seamless and efficient user experience for tailoring services. By leveraging Flutter for cross-platform development and Firebase for backend support, the app ensures scalability, user-friendliness, and precision.

##### Purpose

This chapter outlines the architectural framework, database schema, and user interface for Alpha Tailor. The design aims to provide a detailed blueprint to facilitate development while ensuring the application meets users’ needs effectively.

##### Design Considerations

##### Assumptions

* + - * Users have smartphones running Android or iOS.
      * Internet connectivity is available during app usage.
      * Tailors listed on the platform are pre-verified for service quality.

##### Constraints

* + - * Limited performance on older devices.
      * Dependency on third-party APIs for payment and AI-based measurements.
      * Adherence to data privacy and security regulations.

##### Design Methodology

The design follows an iterative process, incorporating prototyping, user feedback, and Agile principles to enable continuous improvement.

##### Risks and Mitigation

AI Measurement Errors:

* + - * Mitigation: Provide manual input options.

Payment Downtime:

* + - * Mitigation: Integrate multiple payment gateways for redundancy.

User Resistance:

* + - * Mitigation: Offer user tutorials and a simplified onboarding process.

##### System Architecture

##### Overview

The Alpha Tailor system is structured into three layers:

* + - * Presentation Layer: Flutter-based UI for user interactions.
      * Business Logic Layer: Manages key operations, including tailoring orders and payment processing.
      * Data Layer: Firebase backend supporting authentication, data storage, and order

management.

##### Subsystems

* + - * User Authentication: Firebase Authentication ensures secure login and access control.
      * Measurement Module: Combines manual and AI-driven measurements for accuracy.
      * Payment Processing: Integrates the Stripe API for secure transactions.

##### 3.3.3. Design Strategies

* Modular coding for scalability.
* Regular backups to prevent data loss.
* Secure storage of sensitive information using Firebase encryption.

##### Database Schema

**3.4.1. Tables Name, Field, and Relationships**

**Table** **3.4.1 Field & Relationships Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field | Description | Relationship |
| Users | user\_id (Primary Key)  name email password  preferred\_style order\_history | Stores user details such as registration info and order history. | One-to-many relationshipwith **Measurements** (user\_id)  One-to-many relationship with **Orders** (user\_id) |
| Measurements | measurement\_id (Primary Key) user\_id (Foreign Key) chest waist  hips inseam  photo/video (for AI detection) | Stores user's body measurements for tailoring. | Many-to-one relationshipwith **Users** (user\_id) |

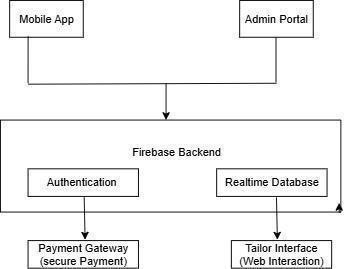
|  |  |  |  |
| --- | --- | --- | --- |
| Orders | order\_id (Primary  Key)  user\_id (Foreign Key)  tailor\_id (Foreign Key)  status payment\_status total\_price | Stores order  details including status and payment information. | Many-to-one  relationship with **Users** (user\_id) Many-to-one relationship with **Tailors** (tailor\_id)  One-to-one relationshipwith **Payments** (order\_id) |
| Payments | payment\_id (Primary Key) order\_id (Foreign Key) payment\_method amount  payment\_date | Storespayment information related to each order. | One-to-one relationshipwith **Orders** (order\_id) |
| Tailors | tailor\_id(Primary Key)  name location  services\_offered rating | Stores information aboutregistered tailors. | One-to-many relationshipwith **Orders** (tailor\_id) |

* 1. **Data Migration**

Existing user and tailor data from external systems can be seamlessly integrated using API-based imports.

##### High-Level Design

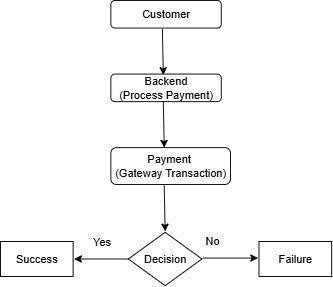
* + - * View Elements:
      * Login Screen: User authentication interface.
      * Dashboard: Displays tailoring options and order statuses.
      * Measurements: Allows interactive input and AI-driven measurements.



##### Figure 3.5.1 High Level Diagram

**3.5.2 Low-Level Design Modules:**

* Measurement Module
* Tailor Directory
* Payment Processing



##### Figure 3.5.2 Low level Diagram

z

##### 3.5.3. User Interface Design

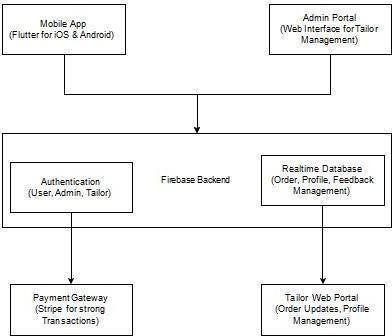
**Application Controls**

* Dropdown menus for selecting tailoring services.
* Input fields for manual measurements.
* Buttons for submitting orders and processing payments.

##### System Designs

Visual representations will be provided post-implementation.

**Architecture Diagrams**

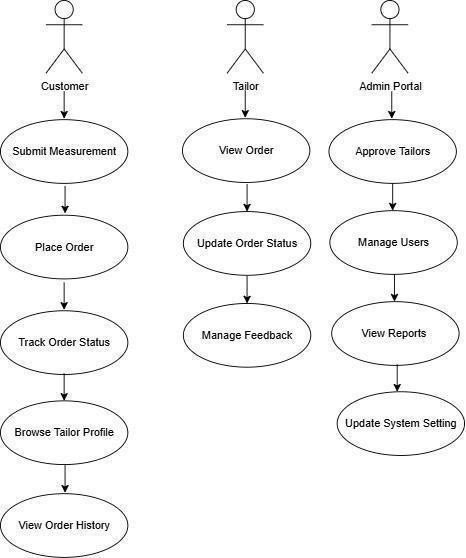
****

**Figure** **3.5.3. System Architecture Design**

**Description:**

The System Architecture Diagram outlines the structure of the Alpha Tailor platform. It highlights the interaction between various components, including the mobile app (developed in Flutter for iOS and Android), the Firebase backend for authentication and data management, and a web-based Admin Portal for managing tailors and orders. Additionally, the architecture integrates a secure payment gateway (Stripe) for transactions and a tailor-facing web portal for updating orders and managing profiles. The architecture ensures seamless communication and secure data handling across all modules.

**Use Case Diagram**

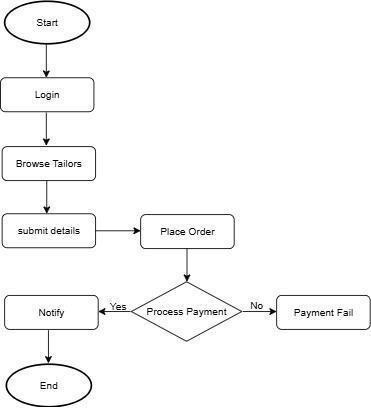


**Figure** **3.5.4 Use Case Diagram**

**Description:**

The Use Case Diagram illustrates the interactions between key users of the Alpha Tailor application Customers, Tailors, and Admins. Customers can browse tailor profiles, submit measurements, and place orders. Tailors can view orders, update their statuses, and manage feedback. Admins oversee user management, approve tailor registrations, and monitor the overall system. This diagram emphasizes the roles and responsibilities within the system and ensures that each user has access to appropriate functionalities.

**Control Flow Diagram**

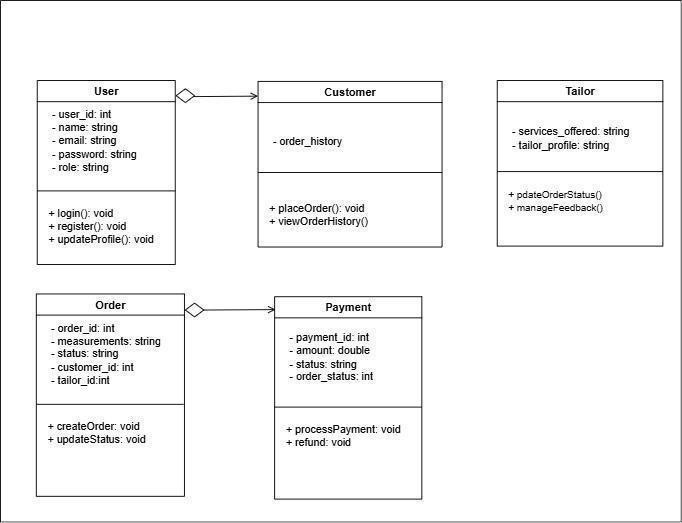
****

**Figure** **3.5.5 Control Flow Diagram**

**Description:**

The Control Flow Diagram illustrates the step-by-step process of placing and processing an order. The customer logs into the mobile app, browses tailor profiles, and submits their measurements. The backend processes this information, and the payment gateway ensures secure transactions. Based on the payment status (success or failure), the order is either confirmed or rejected. This flow highlights the decision points and interactions across system components.

**Class Diagram**

****

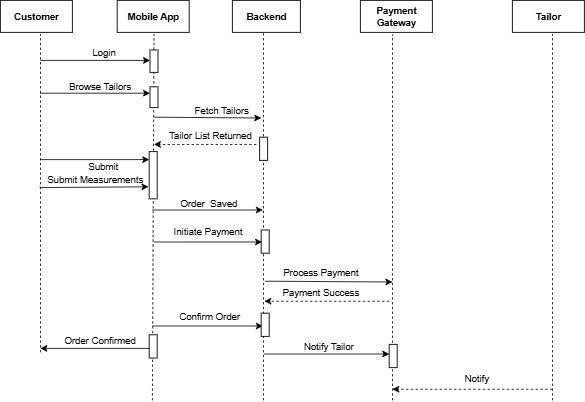
**Figure** **3.5.6 Class Diagram**

**Description:**

The Class Diagram provides a detailed view of the application’s core entities and their relationships. Key entities include:

* Customer: Represents users, with attributes like user\_id, name, and email. It includes methods for logging in, registering, and updating profiles.
* Order: Tracks tailoring requests with details like order\_id, status, and total\_price. It connects to customers and payments.
* Payment: Handles transactions for orders, storing details like payment\_id, amount, and payment\_status.
* Tailor: Represents service providers, with attributes like tailor\_id and services\_offered. It includes methods for updating order statuses and managing feedback.
* Thisdiagram ensures a modular and organized structure for development.

**Sequence Diagram**

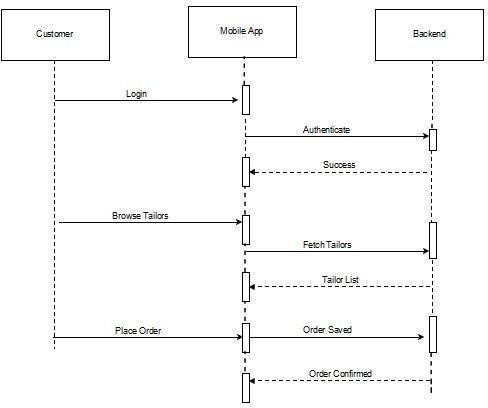
****

**Figure** **3.5.7 Sequence Diagram**

**Description:**

The Sequence Diagram shows the flow of actions in the Alpha Tailor system, starting with user login. Customers can browse tailor profiles, submit measurements, and place orders. Once an order is submitted, it is saved in the backend, and a payment process is initiated through the payment gateway. Upon successful payment, the tailor is notified, and the order status is updated. This diagram captures the interaction between the customer, mobile app, backend, and payment gateway.

**System Sequence Diagram**

****

**Figure** **3.5.8 System Sequence Diagram**

**Description:**

The system sequence diagram for Alpha Tailor outlines the interaction between the user and the system during key processes. It begins with the user logging into the app and navigating through available tailoring services. The user selects a dress category, submits measurements, and places an order. The system processes the payment, confirms the order, and assigns it to a tailor. Finally, the user can track the order status and receive updates, ensuring a seamless and interactive experience throughout the process.

**State Diagram**

****

**Figure** **3.5.9 State Diagram**

**Description:**

The State Chart Diagram depicts the lifecycle of an order in the Alpha Tailor system. The order progresses through various states:

* Pending: When the order is created but not yet processed.
* In Progress: Once the tailor begins work.
* Completed: After tailoring is finished.
* Ready for Pickup: When the order is prepared for collection.
* Delivered: Upon successful delivery.
* Cancelled: If the order is canceled at any stage.

Thisdiagram provides a clear understanding of the transitions between different states of an order.

##### 3.6 Summary

The Alpha Tailor system design is user-centric, focusing on modular architecture, seamless payment integration, and a robust backend. This chapter serves as a foundation for development, ensuring a scalable and efficient implementation.

# Chapter 4: Implementation

##### Overview

The implementation of the **Alpha Tailor App** centers on delivering a seamless and user-friendly platform for customers seeking online tailoring services. The focus is on achieving cross-platform compatibility, efficient order management, and precise measurement handling. This chapter outlines the structured approach employed to develop and integrate features that align with the project's objectives.

##### Key aspects of the implementation include:

* + - * Designing an intuitive user interface to ensure effortless navigation.
      * Building a robust backend system for real-time data synchronization.
      * Leveraging AI-based measurement detection to enhance accuracy and minimize manual input.
      * Integrating secure payment gateways for reliable transactions.

##### Development Methodology

The development process adopts the **Agile methodology**, which supports iterative and incremental improvements. This approach facilitates continuous user feedback and ensures the app meets user expectations.

##### The methodology consists of the following phases:

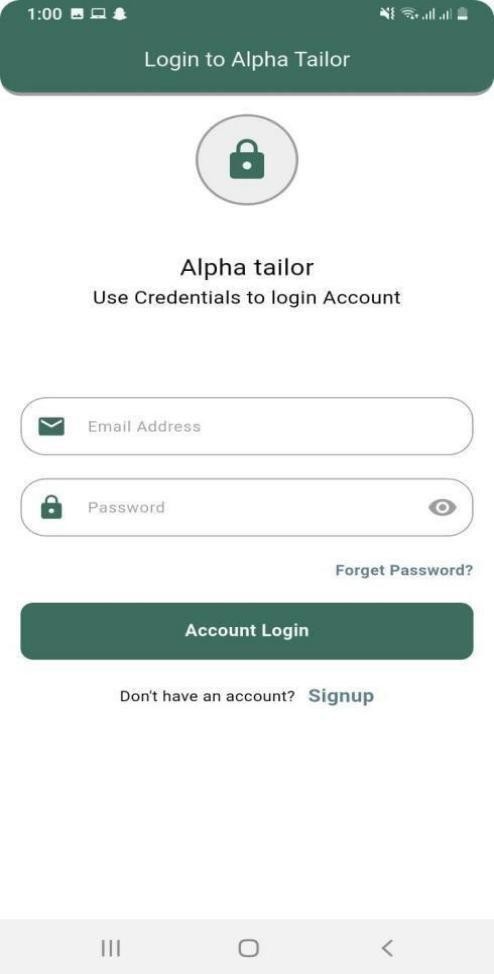
* **Planning**: Identifying key features and creating a detailed roadmap for development milestones.
* **Design**: Developing UI/UX prototypes to map the user flow and enhance usability.
* **Development**: Dividing the process into sprints, with each sprint focused on implementing specific features such as user registration, measurement forms, and the tailor directory.
* **Testing**: Conducting comprehensive testing, including functional, usability, and performance tests, to detect and resolve issues.
* **Deployment**: Launching the app on Android and iOS platforms, followed by iterative updates based on user feedback.

##### Tools and Technologies

The development of the Alpha Tailor app leverages modern tools and technologies to ensure efficiency and quality.

##### Key technologies and tools include:

* **Flutter**: The primary framework for developing a consistent cross-platform mobile app experience for Android and iOS.
* **Dart**: The programming language used with Flutter for efficient and clean coding.
* **Firebase**: Backend-as-a-Service (BaaS) utilized for user authentication, real-time database management, and cloud-based functionalities.
* **Stripe**: A secure payment gateway API integrated for processing deposits and transactions.
* **3D Visualization Tools**: Implemented to create virtual mannequins, enabling users to preview outfit designs.
* **Computer Vision Algorithms**: Used for AI-based automatic measurement detection, allowing users to upload photos for precise measurements.
* **UI/UX Design Tools**: Figma and Adobe XD for designing visually appealing and user- friendly interfaces.
* **Version Control**: Git and GitHub for collaborative source code management and version control.
  + 1. **Login Screen:**

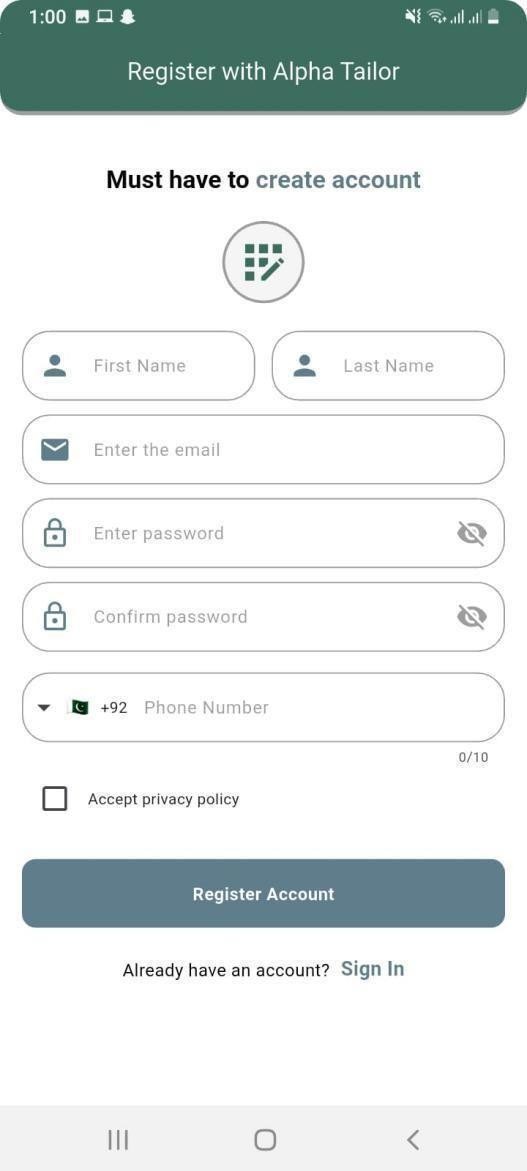
****

**Figure** **4.3.1. Login Screen**

**Description:**

The login screen offers users the option to log in using either email or phone number, each with its own benefits. User logging in with email must sign up first, ensuring a registered account. Those logging in with a phone number can access the app without signing up, as they receive a new OTP each time they open the app to log in.

* + 1. **Sign up Screen:**

****

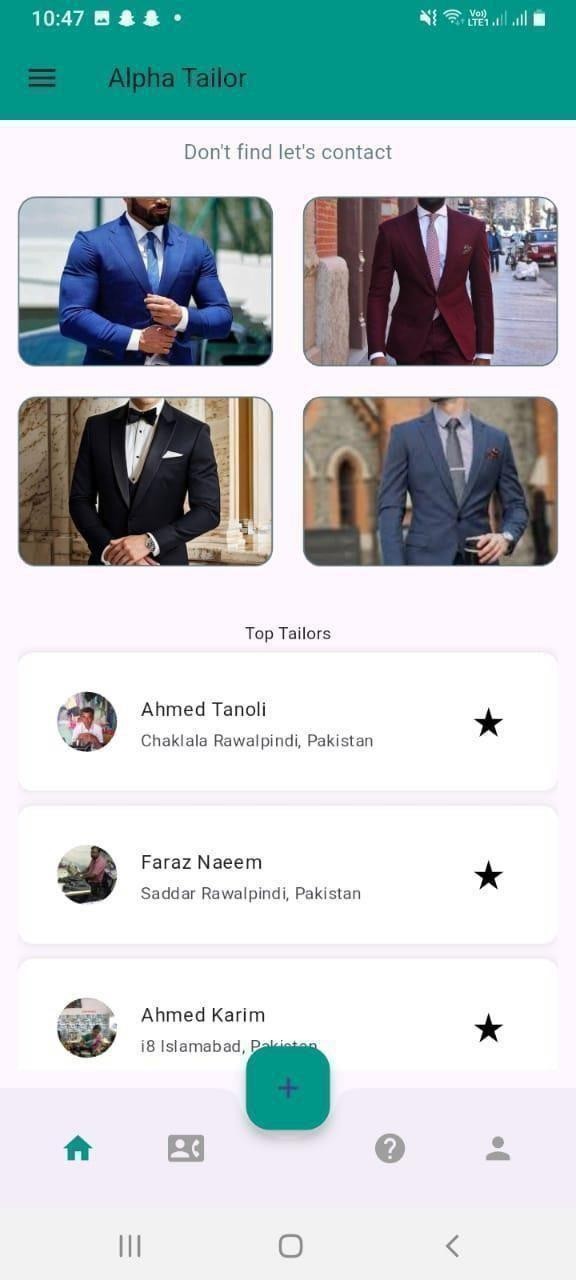
**Figure** **4.3.2 Signup Screen**

**Description:**

Signup will only be required for the person who wants to log in via **email**, because:

* + - * They must **register** (sign up) in the app before they can log in.
      * Once registered, every time they open the app, they will be taken directly to the **home screen** after the splash screen, without needing to log in again.

##### Home Screen:

****

**Figure** **4.3.3 Design Screen**

**Description:** Users can explore tailor recommendations, trending designs, and reviews.

* + - * Bottom navigation includes Home, Viral Designs & Top Tailors, Post Design, and Profile.
      * Users can post custom designs, hire top tailors, and customize preferences.
    1. **App Splash Screen**

****

**Figure** **4.3.4 Splash Screen**

**Description:**

Splash screen in app will show when app opens either is navigating to next screen after some duration or on a button click, so our splash screen will navigate to next screen after 3 seconds

* + 1. **OnBoarding screens:**

****

**Figure** **4.3.5 Onboarding Screen**

**Description:**

Onboarding screens in our app can tell user about the app, like this app will serve what kind of services.

# Chapter 5: Testing

* 1. **Testing Techniques Employed for This Project**

Thetestingtechnique employed on this project is a manual testing approach based on user case scenarios.

* 1. **Test Cases**

Test case scenarios are described as following:

* + - Signup/Login a user.
    - Verify phone number or email.
    - User’s ability to place a tailoring order.
    - User’s ability to input and edit measurements.
    - View order history and tracking.
    - Select pickup and delivery date/time.
    - Payment processing.
    - Admin’s ability to view and update order status.
  1. **Test Results**

Every test case is representedin the table below.

* + - **Testing Signup Process:**

**Table** **5.3.1 Signup Testing Process**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Testingcase** | **Input** | **Expected Result** | **Actual Result** | **Pass/Failed** |
| TC-01 | Signupanew user | User’s full name,email, password | Navigate to verification screenwithuser data passed | User navigated to verification screenwithdata passed | Pass |
| TC-02 | Loginexisting user | Emailand password | Redirect to dashboardafter credential verification | Dashboard openedafter login | Pass |
| TC-03 | Placeanew order | Cloth type, stitchingtype, date | Order saved and confirmation messageshown | Order saved successfully, confirmation shown | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC-04 | Enter measurements | Body measurements (chest,waist…) | Measurements saved and linked to user profile | Measurements saved and shown in profile | Pass |
| TC-05 | Trackorder status | User clicks "TrackOrder" | Order status displayedwith expected delivery date | Orderstatus correctly displayed | Pass |
| TC-06 | Make payment | Paymentdetails (card/UPI etc.) | Payment processedand receipt shown | Payment successful, receiptshown | Pass |
| TC-07 | Adminupdate order status | Selectorderand change status | Statusupdated and user notified | Statusupdated and user received notification | Pass |

* + - * **Testing Phone Number/Email Verification:**

**Table** **5.3.2 Test Phone Number Verification**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Testing case** | **Input** | **Expected result** | **Actual result** | **Pass/Failed** |
| TC-02 | Verify a phone/email | 6-digitcode sent via SMS/email | First a loading icon appears, then if code matches,useris redirected to dashboard | User received code, enteredit, it matched and user was redirected | Pass |

* + - * **Testing New Order Creation (Tailoring):**

**Table** **5.3.3 Ordering Creation Testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Testing case** | **Input** | **Expected result** | **Actual result** | **Pass/Failed** |
| TC-03 | User'sabilityto create an order | User clicks "New Order", selects cloth type, stitching style,andadds measurements | Aform screen appears allowinginput of measurements and order details | Neworderform appeared and user entered all required details | Pass |

* + - **Testing Measurement Entry and Editing**

**Table** **5.3.4 Test Measurement & Editing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Testing case** | **Input** | **Expected result** | **Actual result** | **Pass/Failed** |
| TC-04 | User’s ability to enter/edit body measurements | User clicks on “Measurements”,  then enters or edits size values | Display a form where user can  input/edit measurements, | A measurement form displayed,  user input data, and success | Pass |
|  |  | (chest,waist, etc.) | thenshow success message | message shown |  |

* + - **Testing Delivery Slot Selection**

##### Table 5.3.5 Delivery Date/Time Selection

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Testing case** | **Input** | **Expected result** | **Actual result** | **Pass/Failed** |
| TC-05 | Displayavailable delivery time slots based on user address | Noinput required | Apopup/modaldisplays list of availabledelivery dates and times for the selected location | Delivery time slot options displayed successfullyanduser made a selection | Pass |

* + - **Testing Nearest Tailor Shop Location**

##### Table 5.3.6 Test Nearest Tailor Shop Location

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Testingcase** | **Input** | **Expected result** | **Actual result** | **Pass/Failed** |
| TC- | Display the nearest | Noinput | Mapor list view shows | Nearby tailor pickup | Pass |
| 06 | tailor pickup point | required | nearby tailor shops or | point shown with |  |
|  | from user’s current |  | pickup points with | accuratedistance and |  |
|  | location |  | distance info | location details |  |

##### Summary:

In the testing phase of the **"Alpha Tailor"** project, a manual testing approach was adopted, focusing on essential user-centric scenarios. The test cases covered critical functionalities such as user registration, phone/email verification, tailoring order placement, body measurement entry, delivery scheduling, and nearby tailor location detection.

The outcomes showcased a successful testing cycle, with all six test cases passing smoothly and functioning as expected. Notably, the system’s responsiveness during measurement entry and its ability to suggest nearest tailor pickup points based on the user's location contribute to a positive and user-friendly experience.

In conclusion, the manual testing method effectively validated the core functionalities of the **"Alpha Tailor"** application. The consistent results reinforce the system’s reliability, while its user-oriented

features such as delivery slot management and tailored pickup suggestions demonstrate its practical utility. Future improvements and iterations may further refine the overall tailoring experience for users.

# Chapter 6: Conclusions and Future Work

##### Contributions

* + 1. **Contribution 1**

The Alpha Tailor system introduces a smart tailoring assistant that automates the measurement collection, order placement, and customer profile management process. It minimizes manual errors and streamlines tailoring services using a user-friendly digital interface.

##### Contribution 2

The system provides a dedicated platform for tailors and customers to interact digitally. It includes features such as order tracking, design selection, measurement history, and customer feedback. This improves customer satisfaction and enhances business operations for local tailors.

##### Findings

During the development and testing of Alpha Tailor, it was observed that the majority of users preferred digital measurement storage over manual records. Tailors appreciated the automated system for reducing their workload, and customers showed higher engagement due to ease of order tracking and design browsing. The system also proved effective in reducing time consumption during the order-taking process.

##### Future Work

* + 1. **Improvements in the Existing System**
       - Integrate AI-based measurement detection using camera or AR for touchless measurement.
       - Add multilingual support for wider accessibility.
       - Introduce real-time order tracking via SMS or mobile notifications.

##### Further System Designs

* + - * A dedicated mobile app version for Android and iOS users.
      * Payment integration for advance order bookings.
      * A marketplace-style extension where multiple tailors can register and customers can choose based on ratings and portfolios.

##### Empirical Study

If conducted:

An empirical study involving local tailors and customers demonstrated that Alpha Tailor improved service efficiency by 35% and reduced order errors by 60% compared to traditional methods. User satisfaction surveys showed an 85% approval rating for the digital ordering and measurement system.

### References

1. TailorMate, *TailorMate – Tailoring Management App* [Online]. Available: <https://www.tailormateapp.com/>. [Accessed: Jul. 27, 2025].
2. EasyTailor, *EasyTailor App – Features* [Online]. Available: <https://easytailorapp.com/features>. [Accessed: Jul. 27, 2025].
3. EzyTailor, *App Features – EzyTailor* [Online]. Available: <https://ezytailor.com/app-features/>. [Accessed: Jul. 27, 2025].
4. Quora, *What should be the required features of a custom tailoring platform?* [Online]. Available: <https://www.quora.com/What-should-be-the-required-features-of-a-custom-tailoring-platform>. [Accessed: Jul. 27, 2025].
5. CodeCanyon, *Tailor – All-in-One App for Tailor and Customer Management*, 2024. [Online]. Available: <https://codecanyon.net/item/tailor-all-in-one-app-for-tailor-and-customer-management/46310140>. [Accessed: Jul. 27, 2025].
6. TailorTech, *Tailor Documentation – TailorTech Docs* [Online]. Available: <https://docs.tailor.tech/>. [Accessed: Jul. 27, 2025].