



**SCHOOL OF BUSINESS AND ECONOMICS  
DEPARTMENT OF BUSINESS INFORMATION TECHNOLOGY**

**COURSE: SYSTEMS ENGINEERING, DATABASE AND JAVA PROGRAMMING CODE**

**Project Name:**

**ONLINE TAILOR BOOKING MANAGEMENT  
SYSTEM**

**Reg No:222008235**

**Submitted to: Dr. BUGINGO Emmanuel**

**Date: 08/03/ 2024**

## Table of contents

### Contents

Table of contents.....	1
1. Introduction.....	4
2. Statement of the problem .....	4
2.1 Description of the existing system .....	4
2.2 Problems of the current system .....	4
2.3 How the proposed system will work .....	4
2.4 Describe other alternatives.....	5
3. Software requirement.....	5
3.1 User Requirements .....	5
3.1.1 Functional requirement .....	6
3.1.2 Non-function requirement.....	6
3.1.3 Portability requirement.....	7
3.1.4 Reliability requirement.....	7
3.1.5 Usability requirement.....	8
3.1.6 Space requirement .....	9
3.1.7 Organization requirement.....	9
3.1.8 Implementation requirement .....	10
3.1.9 External environment requirement .....	10
3.1.10 Privacy requirement .....	11
3.1.11 Safety requirement .....	13
3.2 System requirements .....	14
3.2.1 Minimum End-user Hardware Requirements .....	15
3.2.2 Minimum End-user Software Requirements .....	16
3.3 Software specifications .....	17
4. DATA FLOW DIAGRAM (LEVEL 0, LEVEL 1).....	19
4.1 DFD LEVEL 0.....	19
4.2 DFD LEVEL 1 .....	20
5. Feasibility study .....	21
5.1 Technical Feasibility .....	21
5.2 Financial Feasibility .....	22
5.3 Market Feasibility .....	23

5.4 Economic Feasibility.....	24
5.5 Legal and Regulatory Feasibility .....	25
5.6 Operational Feasibility .....	26
6. Description of the selected Process Model .....	27
6.1 Cause of your selection .....	27
7. Conclusion.....	28
8. References .....	29

# CHAPTER 1: SYSTEM ENGINEERING

## 1. Introduction

Online tailoring management system is a system aimed to assist in management of tailoring activities within the industry. It will provide online services to customers such as: measurement submission to their tailors, check whether their garments are finished and also help in proper keeping of records. This will ensure availability of right information, information safety, easy storage, access and retrieval.

The study aims at building a computerized tailoring management system that would be more effective and efficient than the existing manual system.

## 2. Statement of the problem

. Considering the above section, there are many problems associated with the existing manual system, they include the following:

- ❑ Duplication in records of the customers.
- ❑ There is a problem of storage of the taken measurements i.e. can easily get lost.
- ❑ Information retrieval from these sources is not easy

### 2.1 Description of the existing system

The existing system was found to be completely manual, i.e. customers' information is captured in books, there also required to walk to the tailor shop to get their measurements taken. Customers also go to the tailor shops to check on the progress of their garments.

### 2.2 Problems of the current system

Currently customers have to walk to the tailor shops to get their measurements taken for the tailoring of their garments. Their details are taken and kept on papers. Customers too need to move from their offices to go and check for the clothes whether there complete or not. This is time consuming and costly. Due to the manual systems in use, the whole process tends to be slow. Customers too have no prior information on cost of netting their garments.

### 2.3 How the proposed system will work

The Online Tailoring Management System will permit to register and deliver measurements to

the tailor for the next process to follow.

It also maintains clients' information and generating various reports about the tailor shop. The main users of the project are clients and system Administrator.

It also enables customers to check the status of their garments i.e. if ready or not for collection.

The system provides information about the cost, the fabric type the customer want his/her dress knit from, the duration a customer wants the dress finished, the type of material to be used, quantity in terms of pairs needed and most importantly, the system computes the total cost and avails that information to the customer.

However, online payment has not been achieved, but the customer is expected to pay either via mobile money transfer services like m-pesa, pesapal or cash when they come to pick their clothes

#### 1.4 Describe other alternatives

There are several alternatives to an online tailor booking management system, depending on the specific needs and preferences of the tailor or tailoring business. Here are a few alternatives worth considering:

- ❑ Manual Appointment Scheduling: Some tailors may prefer a more traditional approach by managing appointments manually. This could involve maintaining a physical appointment book or using a spreadsheet or calendar software to record bookings. While this method is simple and cost-effective, it lacks the automation and convenience provided by dedicated management systems.
- ❑ Generic Calendar Software
- ❑ Custom-Built Software
- ❑ Online Scheduling Tools
- ❑ Tailor-Specific Software
- ❑ Online Marketplace Platforms

## 2. Software requirement

- ❑ Browsers: Microsoft Internet Explorer, Firefox
- ❑ Server: Xampp
- ❑ Operating System: Windows XP, 7, 8 and Linux.
- ❑ Back end: MySQL.
- ❑ Front end: PHP Script and scripting is done using JavaScript

### 3.1 User Requirements

It is very important to get users of the system fully involved such that the problem of change

management does not arise. The system is expected to be:

- ☐ Easy to learn and use
- ☐ Improve on the efficiency of information storage and retrieval
- ☐ Produce results faster i.e. measurements submission or checking clothe status, therefore reducing on time wasted during to and fro travelling.
- ☐ Provides attractive interfaces with easy navigation throughout the system
- ☐ Faster, flexible and convenient.
- ☐ A system that stores data and produces reports timely and accurately

- 16 -

### 3.4.3.2 Functional Requirements

Functional requirements capture the intended behavior of the system. This behav

#### 3.1.1 Functional requirement

Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. Therefore the proposed system is able to:

- ☐ Capture customer information, store it and make it available at the time of need.
- ☐ Present the users with a real-time display on the garments status.
- ☐ Generate reports accurately and timely
- ☐ Search and display customer information details
- ☐ Computes the total cost of a garment depending on the selected fabric, type of material, quantity and duration and avails that information to the customer

#### 3.1.2 Non-function requirement

Non-functional requirements are requirements which specify criteria that can be used to judge the operation of a system, rather than specific behaviours. This is contrasted with functional requirements that specify specific behaviour or functions. Systems must exhibit software quality attributes, such as accuracy, performance, cost, security and modifiability plus usability, i.e. easy to use for the intended users. NFRs help to achieve the functional requirement of a system. Thus the proposed system does the following:

- ☐ The system has high performance and reliability level. The mean time between failures,

mean time to repair, and accuracy are very high.

☑ The system has user-friendly interfaces. This ensures the ease with which the system can be learned or used. The system can allow users to install and operate it with little or no training.

☑ Handles growing amounts of work in a graceful manner as can be readily enlarged i.e. the ease, with which the system can be modified to handle a large increase in users, workload or transactions.

☑ The system prevents unauthorized access to the system with user authentication via login on s

### 3.1.3 Portability requirement

If portability is a requirement for the tailor booking management system, it means that the system should be accessible and functional across multiple devices and platforms. Here are some considerations for meeting the portability requirement:

**Responsive Web Design:** Develop the booking management system with a responsive web design approach. This ensures that the system adapts and displays properly on different screen sizes, including desktop computers, laptops, tablets, and smartphones. Users should be able to access and use the system seamlessly regardless of the device they are using.

**Mobile Applications:** Create dedicated mobile applications for popular platforms like iOS and Android. These applications can provide a more optimized and user-friendly experience for customers and tailors using smartphones or tablets. The mobile apps should offer the same functionality as the web version, allowing users to book appointments, manage orders, and track progress on the go.

**Cloud-Based Infrastructure:** Host the booking management system on a cloud-based infrastructure. This allows users to access the system from any device with an internet connection, without the need for local installations. Cloud hosting also offers scalability, data backup, and the ability to synchronize data across multiple devices.

**Cross-Platform Compatibility:** Ensure that the booking management system is compatible with multiple web browsers, such as Chrome, Firefox, Safari, and Edge. This ensures that users can access the system regardless of their preferred browser. Additionally, consider compatibility with different operating systems, including Windows, macOS, and Linux, to cater to a wider range of users.

#### 3.1.4 Reliability requirement

reliability is a requirement for the tailor booking management system, it means that the system should consistently perform its functions accurately and without interruptions. Here are some considerations to meet the reliability requirement:

- **Robust Infrastructure:** Ensure that the system is built on a reliable and robust infrastructure. This includes using high-quality servers, network equipment, and database systems that can handle the expected workload and provide good uptime and performance. Implement redundancy measures, such as backup servers or failover mechanisms, to minimize the impact of any hardware or network failures.
- **Data Backup and Recovery:** Regularly back up the system's data to prevent loss in case of unexpected events or system failures. Have a reliable backup strategy in place, including both on-site and off-site backups, to ensure that data can be restored quickly. Test the data restoration process periodically to verify its effectiveness.
- **Scalability**
- **Error Handling and Logging**
- **Security Measures**
- **Regular Maintenance and Updates**
- **Monitoring and Alerting.**

#### 3.1.5 Usability requirement

usability is a requirement for the tailor booking management system, it means that the system should be easy to use, intuitive, and efficient for both customers and tailors. Here are some considerations to meet the usability requirement:

**User-Friendly Interface:** Design a clean and intuitive user interface that is easy to navigate and understand. Use clear and concise labels, instructions, and visual cues to guide users through the booking process. Organize information logically and present it in a visually appealing manner.

**Simple Booking Process:** Streamline the booking process to minimize the number of steps and reduce complexity. Customers should be able to easily select the desired service, choose an available time slot, and provide necessary details without confusion or unnecessary delays.



**Clear Feedback and Status Updates:** Provide timely and clear feedback to users during the booking process. Inform customers about the availability of time slots, successful bookings, and any additional information or requirements. For tailors, the system should provide updates on new bookings, changes to appointments, and completed orders.

**Responsive Design:** Ensure that the booking management system is responsive and accessible across different devices and screen sizes. The interface should adapt seamlessly to various screen resolutions, whether customers are using desktop computers, laptops, tablets, or smartphones. This allows for a consistent user experience regardless of the device being used.

**Customization Options:** Provide customization options that cater to the specific needs and branding of the tailor or tailoring business. This can include options to add a logo, choose color themes, or personalize the booking confirmation messages. Tailors should also have the ability to define their available services, pricing, and appointment durations.

### 3.1.6 Space requirement

If space is a requirement for the tailor booking management system, it could refer to the physical space needed to accommodate the system infrastructure or the storage space required for data and files. Here are some considerations related to space requirements:

- Hardware Infrastructure
- Data Storage
- Cloud-Based Storage
- File Management
- Workspace for Users
- Backup and Archiving

### 3.1.7 Organization requirement

If organization is a requirement for the tailor booking management system, it means that the system should have features and functionalities to help the tailor or tailoring business effectively organize and manage their appointments, customer information, orders, and other relevant data. Here are some considerations related to the organization requirement:

Appointment Management

Customer Management

Order Tracking

Communication and Notifications

Document and File Management

Reporting and Analytics

Workflow Management

Inventory Management

### 3.1.8 Implementation requirement

At this stage, I will begin coding as per the design specification(s). The output of this step is one or more product components built according to a pre-defined coding standard and debugged, tested and integrated to satisfy the system architecture requirements.

here are some general considerations for implementing such a system:

- Technology Stack
- System Architecture:
- Development Approach
- User Interface Design: Create user interface (UI) designs and wireframes to determine the layout, navigation, and visual elements of the system. Consider usability, accessibility, and user experience (UX) principles to ensure a user-friendly and intuitive interface.
- Backend Development
- Frontend Development
- Integration
- Testing and Quality Assurance
- Deployment and Maintenance
- Training and User Adoption

### 3.1.9 External environment requirement

**Internet Connectivity:** The system relies on internet connectivity to operate effectively. A stable and reliable internet connection is essential for customers to access the booking interface, for tailors to manage appointments, and for data synchronization. Ensure that the system can handle different internet speeds and can gracefully handle intermittent or slow connections.

**Browser and Device Compatibility:** The tailor booking management system should be compatible with various web browsers and devices to ensure accessibility for customers and tailors. It should be tested

and optimized to work across popular browsers such as Chrome, Firefox, Safari, and Edge, and on devices such as desktop computers, laptops, tablets, and smartphones.

**Security and Privacy Regulations:** Consider any security and privacy regulations applicable to the tailor's location or industry. Ensure that the system complies with data protection laws and guidelines, including measures for secure data transmission, storage, and access control. Implement appropriate encryption, user authentication, and data anonymization techniques to protect sensitive information.

**Integration with External Services:** If the tailor booking management system integrates with external services such as payment gateways or messaging platforms, ensure that the necessary APIs or integration methods are available and properly implemented. Consider compatibility with different versions and configurations of these external services.

**Customer Accessibility Requirements:** Take into account any specific accessibility requirements for customers, such as compliance with accessibility guidelines like WCAG (Web Content Accessibility Guidelines). Implement features to enhance accessibility for users with disabilities, including keyboard navigation, screen reader compatibility, and alternative text for images.

**Local Language and Localization:** If the tailor booking management system is intended for users in different regions or countries, consider localization requirements. Provide support for multiple languages, date formats, time zones, and currency conversions to cater to a diverse user base.

**Scalability and Performance:** Anticipate scalability requirements based on the expected user load and growth of the tailor's business. The system should be designed to handle increased traffic, appointments, and data volume without compromising performance. Monitor performance metrics and implement optimization techniques to maintain responsiveness and minimize latency.

**External System APIs and Integrations:** If the system integrates with external APIs or third-party systems such as inventory management, accounting, or marketing tools, ensure compatibility and reliability of these integrations. Consider any potential changes or updates to these external systems that may impact the functionality of the tailor booking management system.

### 3.1.10 Privacy requirement

Privacy is an essential requirement for a tailor booking management system to protect the confidentiality, integrity, and security of user data. Here are some privacy considerations to meet the privacy requirement:

**Data Protection and Compliance:** Implement measures to protect user data in accordance with relevant privacy regulations and industry best practices. Understand and comply with applicable data protection laws, such as the General Data Protection Regulation (GDPR) in the European Union or the California Consumer Privacy Act (CCPA) in the United States. Ensure that user data is collected, processed, stored, and transmitted securely and with user consent.

**User Consent and Transparency:** Obtain user consent for the collection and processing of their personal information. Clearly communicate the purpose, scope, and duration of data processing to users. Provide a privacy policy that explains how user data is handled, including information about data retention, sharing with third parties, and user rights. Make the privacy policy easily accessible and understandable to users.

**Secure Data Transmission:** Encrypt sensitive data during transmission over the internet to prevent unauthorized access. Use secure communication protocols such as HTTPS (SSL/TLS) to protect user data when it is transferred between users and the system's servers.

**Strong Authentication and Access Controls:** Implement robust authentication mechanisms to ensure that only authorized individuals can access sensitive user data. Use strong passwords, multi-factor authentication (MFA), or other authentication methods to verify user identities. Employ access controls to restrict data access based on user roles and permissions.

**Data Minimization and Anonymization:** Collect and retain only the necessary user data for the operation of the booking management system. Minimize the collection of personally identifiable information (PII) to reduce privacy risks. Consider anonymizing or pseudonymizing data whenever possible to protect user identities.

**Data Breach Response Plan:** Develop a data breach response plan to address any potential data breaches or unauthorized access incidents. The plan should include steps for identifying and containing the breach, notifying affected individuals, and cooperating with relevant authorities, as required by applicable regulations.

**Third-Party Service Providers:** If the system uses third-party service providers, ensure that they have appropriate privacy and security measures in place. Evaluate the privacy practices of these providers and sign data processing agreements that outline their responsibilities in protecting user data.

**Regular Security Audits and Updates:** Conduct regular security audits and assessments to identify and address any vulnerabilities or risks to user data. Keep the system and its components up to date with security patches and updates to address any known security vulnerabilities.

By addressing these privacy considerations, the tailor booking management system can provide a secure and privacy-respecting environment for users, instilling trust and confidence in the handling of their personal information.

### 3.1.11 Safety requirement

Safety is an important requirement for a tailor booking management system to ensure the protection and well-being of users. While safety requirements can vary based on specific circumstances, here are some considerations to meet safety requirements:

**Data Security:** Implement robust security measures to protect user data from unauthorized access, manipulation, or loss. This includes encryption of sensitive data, strong access controls, regular backups, and secure storage practices. Adhere to industry standards and best practices for data security.

**User Authentication:** Ensure that the system incorporates secure user authentication mechanisms to prevent unauthorized access. This can include features such as strong password requirements, multi-factor authentication (MFA), and secure password storage practices.

**User Privacy:** Safeguard user privacy by adhering to applicable privacy laws and regulations, as discussed in the previous privacy requirement section. Protect personally identifiable information (PII) and provide transparency and user control over their data.

**System Reliability:** Develop and maintain a reliable system infrastructure to minimize system failures, downtime, or disruptions. Implement redundancy measures, backup systems, and disaster recovery plans to ensure continuous availability of the system.

**Error Handling:** Implement proper error handling mechanisms to prevent system crashes or unexpected behaviors. Provide informative error messages that assist users in understanding and resolving any issues that may arise.

**Accessibility:** Ensure that the system is accessible to users with disabilities. Adhere to accessibility standards and guidelines, such as the Web Content Accessibility Guidelines (WCAG), to accommodate users with visual impairments, hearing impairments, mobility limitations, or other accessibility needs.

**User Safety Information:** Provide relevant safety information to users as needed. This could include reminders or instructions related to safe garment handling, care instructions, or any other safety guidelines specific to the tailor's services.

**Secure Payment Processing:** If the system involves online payments, ensure the use of secure and trusted payment gateways. Implement encryption and secure protocols to protect sensitive financial information during payment transactions.

Compliance with Industry Regulations

### 3.2 System requirements

System requirements for a tailor booking management system can vary depending on the specific needs and scale of the tailor's business. Here are some general system requirements to consider:

- **Operating System:** Determine the operating system(s) that the system will support, such as Windows, macOS, or Linux. Consider the prevalence and compatibility of the chosen operating system(s) among the target users.
- **Web Browser Compatibility:** Identify the web browsers that the system will support, such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari. Ensure that the system is compatible with the most commonly used browsers and their different versions.
- **Hardware Requirements:** Define the hardware specifications necessary for the system to run smoothly. Consider factors such as processing power, memory (RAM), and storage capacity. The requirements may vary depending on the complexity and scale of the system.
- **Database Management System:** Choose an appropriate database management system (DBMS) to store and manage the system's data. Consider factors such as scalability, performance, and security. Common DBMS options include MySQL, PostgreSQL, Microsoft SQL Server, or MongoDB.

- **Network Requirements:** Ensure that the system has access to a stable and reliable internet connection for real-time data synchronization, online booking, and communication features. Determine the network bandwidth requirements based on the expected user load and usage patterns.
- **Software Dependencies:** Identify any specific software dependencies or libraries required for the system to function correctly. This could include programming frameworks, development tools, or third-party software components.
- **Security Measures:** Define the security measures and protocols that the system will implement to protect user data, such as encryption, secure authentication, and access controls. Consider industry standards and best practices for securing web applications and databases.
- **Scalability and Performance:** Anticipate the scalability and performance requirements of the system based on the expected user load and business growth. Ensure that the system architecture and infrastructure can handle increasing demands without sacrificing performance.
- **Integration Requirements:** Determine any external systems, APIs, or services that need to be integrated with the tailor booking management system. Consider the compatibility, documentation, and availability of the required integrations.
- **Localization and Internationalization:** If the system will be used by users in different regions or countries, consider localization and internationalization requirements. This includes supporting multiple languages, date formats, time zones, and currency conversions.
- It's important to gather specific requirements from the tailor or tailoring business to ensure that the system is designed and implemented to meet their unique needs. Conducting thorough requirement analysis and collaborating with a development team or professionals can help ensure that the system meets the specific system requirements.
- **3.2.1 Minimum End-user Hardware Requirements**
- The minimum end-user hardware requirements for a tailor booking management system can vary based on the system's design and technology stack. Here are some general guidelines for minimum hardware requirements:

- Processor: A modern processor with at least a dual-core configuration is recommended. This ensures smooth performance while interacting with the system's interface and handling data processing tasks.
- Memory (RAM): The minimum RAM requirement typically ranges from 2GB to 4GB. Sufficient RAM allows for efficient multitasking and smooth navigation within the system.
- Storage: A minimum of 10GB of available storage space is typically sufficient for the installation of the system and storage of local data. However, the actual storage requirements may vary depending on factors such as the size of the database and the volume of data being processed.
- Display: A monitor with a resolution of 1024x768 pixels or higher is generally recommended to ensure a clear and comfortable viewing experience when using the system's interface.
- Operating System: The system should be compatible with popular operating systems such as Windows, macOS, or Linux. The specific minimum version requirements should be documented based on the chosen operating systems.
- Web Browser: The system should be accessible through commonly used web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari. It is important to specify the minimum browser versions supported by the system.
- [3.2.2 Minimum End-user Software Requirements](#)
- The minimum end-user software requirements for a tailor booking management system can vary depending on the system's technology stack and compatibility. Here are some general guidelines for minimum software requirements:
  - Operating System: The system should be compatible with popular operating systems such as Windows, macOS, or Linux. Specify the minimum supported versions for each operating system.
  - Web Browser: The system should be accessible through commonly used web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari. Specify the minimum browser versions supported by the system.



- **Browser Plugins:** Identify any necessary browser plugins or extensions required for the system to function properly. For example, if the system uses PDF generation, a PDF viewer plugin may be required.
- **Internet Connectivity:** A stable and reliable internet connection is essential for accessing and using the tailor booking management system. Specify the minimum required internet speed and network stability for optimal performance.
- **Additional Software:** Identify any additional software dependencies required by the system. For example, if the system uses a specific database management system, users may need to install and configure the corresponding database client software.
- **Security Software:** Users should have up-to-date security software such as antivirus and firewall programs installed on their systems. These help protect against malware and unauthorized access to user data.
- **Mobile Applications:** If the system has dedicated mobile applications, specify the minimum supported mobile operating systems (e.g., iOS, Android) and their respective minimum versions.

It's important to provide clear and concise instructions to end users regarding the minimum software requirements for accessing and using the tailor booking management system. This helps ensure that users have compatible software configurations and can effectively utilize the system's features.

### 3.3 Software specifications

Software specifications for a tailor booking management system outline the detailed functionality, features, and behavior of the software. Here are some common software specifications to consider:

- **User Roles and Permissions:** Define the different user roles within the system, such as administrators, tailors, and customers. Specify the permissions and access levels associated with each role to control system functionality and data access.

- **Appointment Booking:** Specify the process for customers to book appointments, including selecting services, available time slots, and providing necessary information (e.g., contact details, measurements). Outline any validation rules or restrictions for booking appointments.
- **Calendar and Scheduling:** Define the calendar and scheduling functionality, including the ability to view and manage appointments. Specify how the calendar is displayed, how appointments are organized, and how conflicts or overlapping bookings are handled.
- **Customer Management:** Outline the features for managing customer information, including the ability to add, edit, and search for customers. Specify the required customer details, such as contact information, measurements, preferences, and order history.
- **Order Tracking:** Define the order tracking capabilities, such as the ability to track the progress of orders from start to completion. Specify the stages or statuses that an order can go through, such as measurements taken, fabric selection, in-progress, and completed.
- **Notifications and Reminders:** Specify the types of notifications and reminders that users will receive. This can include appointment confirmations, reminders, or updates on order progress. Outline the communication channels used, such as email, SMS, or push notifications.
- **Online Payments:** Describe the online payment functionality, including the integration with payment gateways. Specify the supported payment methods, such as credit cards or online wallets, and any additional requirements, such as generating invoices or receipts.
- **Reporting and Analytics:** Define the reporting and analytics capabilities of the system. Specify the types of reports available, such as appointment statistics, sales reports, or customer analytics. Outline any customizable reporting options or data visualization features.
- **Integration with External Systems:** Specify any integrations with external systems or APIs, such as inventory management or messaging services. Describe the required functionality and data exchange between the tailor booking management system and these external systems.

- **Security and Privacy:** Define the security measures implemented in the system, such as encryption, access controls, and data privacy practices. Outline the steps taken to protect user data and comply with relevant privacy regulations.

These software specifications provide a detailed roadmap for the development and implementation of the tailor booking management system, ensuring that the desired functionality and features are accurately captured and delivered.

### 3. DATA FLOW DIAGRAM (LEVEL 0, LEVEL 1)

Data flow diagrams (DFDs) were used to illustrate the flow of information in a system.

They are hardware independent and do not reflect decision points. They demonstrate the information and how it flows between specific processes in a system. They provide one kind of documentation for reports. These diagrams help to show how data moves and changes through the system in a graphical top-down fashion. They also help to give graphical representation of the system's components, processes and the interfaces between them.

When it came to conveying how data flows through systems (and how that data was transformed in the process), DFDs were the method of choice over technical descriptions for three principal reasons:

- DFDs are easier to understand by technical and non-technical audiences.
- They provide a high-level system overview, complete with boundaries and connections to other systems.
- They provide a detailed representation of the system components.

The diagram below shows the flow of data through the proposed system. It depicts the flow of information and the transformation that is applied as data moves from input to output

This figure defines the boundary between the system, or part of a system, and its environment, showing the entities that interact with it, this diagram is a high level view of a system.

A Level 0 Data Flow Diagram (DFD) provides a high-level overview of the system and its major processes. It represents the interactions between external entities and the system as a single process. Here is an example of a Level 0 DFD for a tailor booking management system:

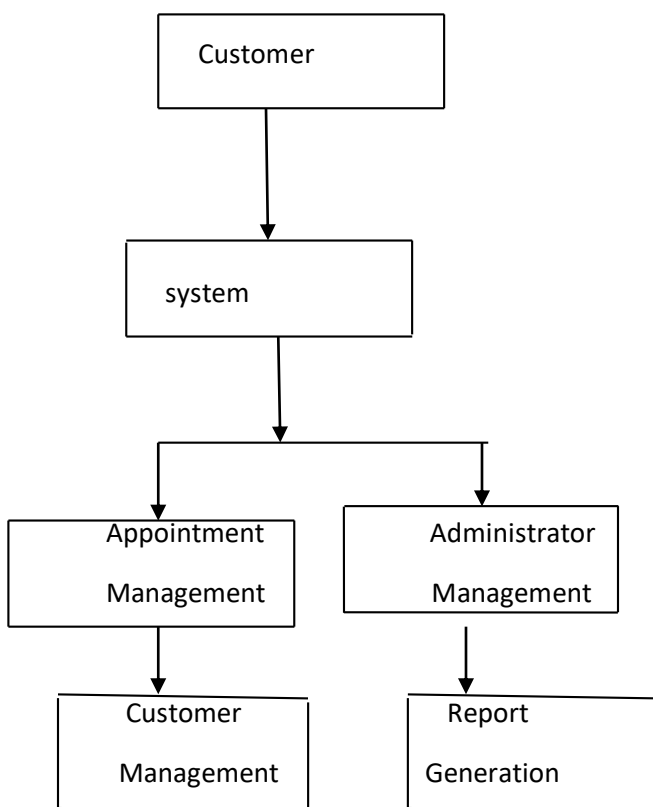
#### 4.1 DFD LEVEL 0

**In the Level 0 DFD:**

The "Customer" represents the external entity that interacts with the system to book appointments, view orders, and access customer-related information.

The "System" is the main process of the tailor booking management system. It receives inputs from customers, processes them, and generates outputs. It handles functionalities such as appointment booking, order tracking, and customer management.

The "Administrator" represents the external entity responsible for managing the system, including tasks like managing appointments, customer information, and system configurations.



The "Customer" represents the external entity (customer) interacting with the system.

The "System" process is the main process of the tailor booking management system.

The system is further broken down into several sub-processes:

"Appointment Management" handles the processes related to appointment booking, scheduling, and notifications for customers.

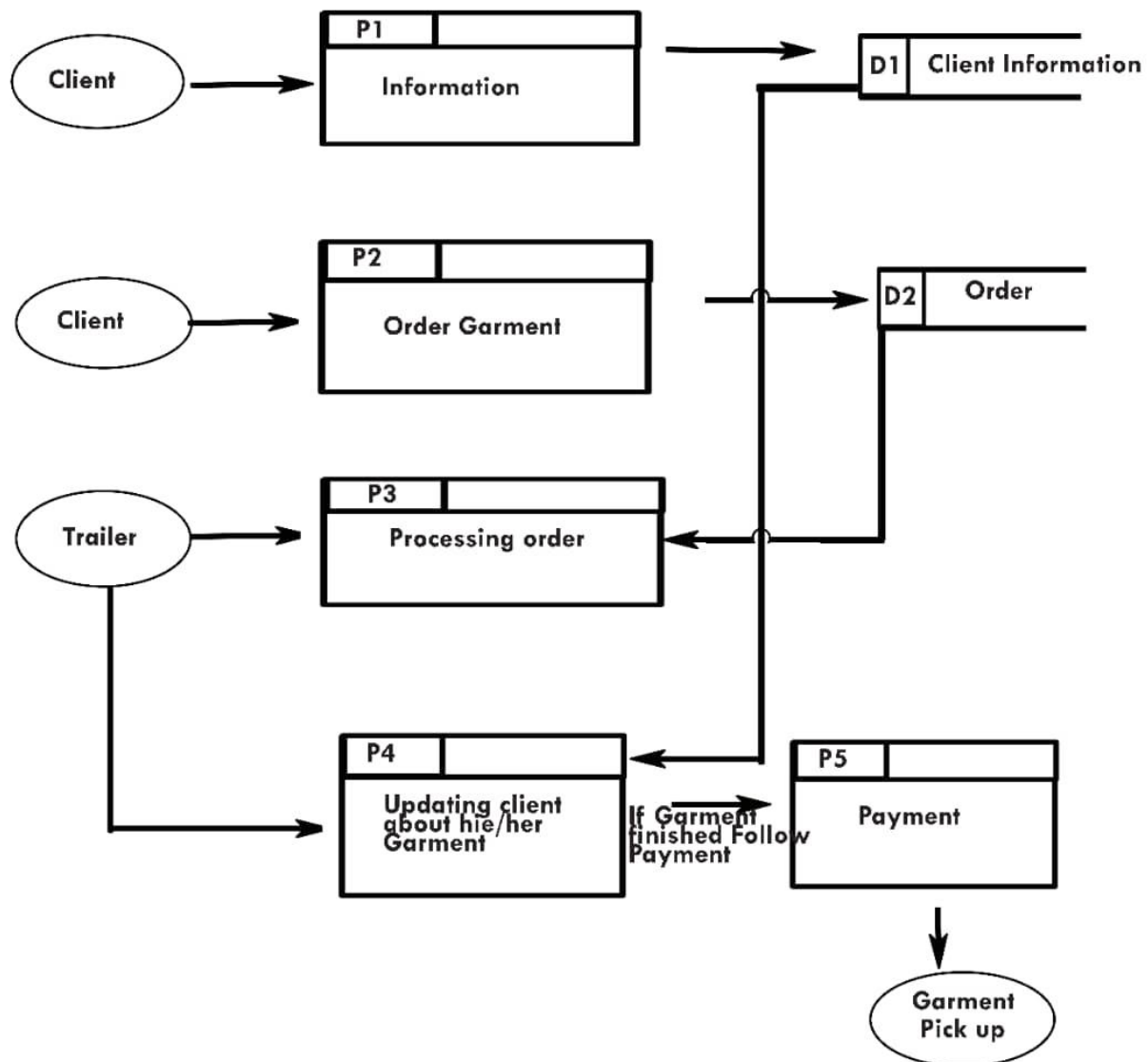
"Administrator Management" handles the processes related to system administration, such as managing appointments, customer information, and system configurations.

"Customer Management" handles the processes related to managing customer information, including adding, editing, and searching for customer records.

"Report Generation" handles the processes related to generating reports, such as appointment statistics, sales reports, or customer analytics.

#### 4.2 DFD LEVEL 1

A Level 1 Data Flow Diagram (DFD) provides a more detailed view of the processes, data flows, and data stores identified in the Level 0 DFD. It breaks down the high-level processes into sub-processes and further elaborates on the data flows between them.



## 5. Feasibility study

Here, I will carry out a study to gain an understanding of the customers' current system and problems experienced in this system through interviews, observations, participations etc. I will use the obtained data to determine the viability of the system being proposed in terms of technical, economic and social feasibilities.

### 5.1 Technical Feasibility

Technical feasibility refers to the assessment of whether a proposed project or system can be successfully implemented from a technical standpoint. It involves evaluating the availability of resources, technology, and skills required to develop and operate the system.

## 5.2 Financial Feasibility

Financial feasibility refers to the assessment of whether a proposed project or system is financially viable and sustainable. It involves evaluating the costs and benefits associated with the development, implementation, and operation of the tailor booking management system. Here are some factors to consider when assessing the financial feasibility:

- **Development Costs:** Evaluate the costs associated with developing the tailor booking management system. This includes expenses related to software development, customization, integration with existing systems, user interface design, and testing.
- **Infrastructure Costs:** Consider the costs associated with acquiring or upgrading the necessary hardware, servers, networking equipment, and storage devices to support the system. Assess whether the existing infrastructure needs to be scaled or if additional investments are required.
- **Licensing and Software Costs:** Determine if there are any licensing fees or costs associated with using third-party software components, libraries, or frameworks in the system. Evaluate the licensing models (e.g., one-time payment, subscription) and ongoing maintenance costs.
- **Operational Costs:** Evaluate the ongoing operational costs associated with running and maintaining the system. This includes expenses such as hosting fees, data storage costs, system monitoring, security measures, and backup and recovery solutions.
- **Return on Investment (ROI):** Assess the potential return on investment for implementing the tailor booking management system. Consider the expected benefits such as increased efficiency, improved customer satisfaction, reduced administrative overheads, and increased revenue from streamlined processes.
- **Cost Savings:** Identify any potential cost savings that the system can bring. For example, the system may automate manual processes, reduce paperwork, or eliminate the need for third-party services, resulting in cost savings over time.
- **Revenue Generation:** Explore potential revenue streams that can be derived from the system. This could include fees charged for booking services, upselling opportunities, or additional revenue-generating features or add-ons.
- **Payback Period:** Estimate the time it will take for the benefits or savings from the system to offset the initial investment and reach breakeven. Consider the timeframe for cost recovery and evaluate whether it aligns with the business goals and expectations.



- **Financial Risk Assessment:** Identify and evaluate potential financial risks associated with the system's implementation and operation. Assess factors such as market demand, competition, regulatory changes, and technology obsolescence that may impact the financial viability of the system.

By carefully assessing the financial feasibility, considering the costs, benefits, and potential returns, you can make informed decisions about the implementation of the tailor booking management system. It allows you to understand the financial implications and determine whether the investment aligns with the business's budget, goals, and long-term sustainability.

### 5.3 Market Feasibility

Market feasibility refers to the evaluation of the potential success and viability of a proposed product or service within a specific market. It involves analyzing various factors to determine if there is sufficient demand, a suitable target market, and a competitive landscape that supports the introduction and sustainability of the product or service. Here are key factors to consider when assessing market feasibility:

- **Target Market Analysis:** Identify and understand the specific market segment(s) that the tailor booking management system will cater to. Analyze their needs, preferences, behaviors, and demographics. Assess the size, growth potential, and profitability of the target market.
- **Market Demand:** Evaluate the demand for a tailor booking management system within the target market. Identify the pain points, challenges, and inefficiencies that the system can address. Conduct market research, surveys, or interviews to gather insights and validate the demand.
- **Competitive Analysis:** Assess the competitive landscape and identify existing competitors or alternative solutions in the market. Analyze their strengths, weaknesses, market share, pricing strategies, and differentiation. Determine how the tailor booking management system can offer unique value and differentiate itself from competitors.
- **Unique Selling Proposition (USP):** Define the unique features, benefits, or value proposition that the tailor booking management system offers to the target market. Determine how it solves customer pain points, improves efficiency, or enhances the tailoring experience. This USP should resonate with the target market and provide a competitive advantage.

- **Market Trends and Opportunities:** Stay abreast of market trends, technological advancements, and emerging opportunities within the tailoring or booking management industry. Identify any emerging customer needs or changing market dynamics that can be capitalized on to gain a competitive edge.
- **Pricing and Revenue Model:** Determine the appropriate pricing strategy and revenue model for the tailor booking management system. Consider factors such as customer willingness to pay, perceived value, competitive pricing, and the financial sustainability of the business model.
- **Marketing and Distribution Channels:** Develop an effective marketing strategy to reach and acquire customers within the target market. Identify the most suitable marketing channels, such as online advertising, social media, industry partnerships, or direct sales. Determine the cost-effectiveness and reach of each channel.
- **Legal and Regulatory Considerations:** Understand and comply with relevant laws, regulations, and industry standards that may impact the implementation and operation of the tailor booking management system. Consider data protection, privacy, consumer rights, and any specific regulations related to the tailoring industry.
- **Risk Assessment:** Identify and assess potential risks and challenges that may affect the market feasibility of the system. Evaluate factors such as market saturation, entry barriers, customer adoption barriers, and potential disruptive technologies.

## 5.4 Economic Feasibility

Economic feasibility refers to the assessment of whether a proposed project or system is financially viable and economically justifiable. It involves evaluating the costs, benefits, and potential return on investment (ROI) associated with the implementation and operation of the tailor booking management system. Here are some key factors to consider when assessing economic feasibility:

- **Cost-Benefit Analysis:** Conduct a comprehensive cost-benefit analysis to evaluate the financial implications of the tailor booking management system. Identify and quantify both the upfront costs (development, hardware, software, infrastructure) and the ongoing costs (maintenance, updates, support). Compare these costs with the anticipated benefits, such as increased efficiency, cost savings, revenue generation, and improved customer satisfaction.

- Return on Investment (ROI): Calculate the projected ROI to determine the financial viability of the system. Assess the payback period, net present value (NPV), internal rate of return (IRR), or other financial metrics to assess the profitability of the investment. Compare the expected returns with the initial and ongoing costs to gauge the financial feasibility.

## 5.5 Legal and Regulatory Feasibility

Legal and regulatory feasibility refers to the assessment of whether a proposed project or system complies with relevant laws, regulations, and legal requirements. It involves evaluating the legal and regulatory framework applicable to the project and ensuring that the system operates within the boundaries set by governing bodies. Here are some key factors to consider when assessing legal and regulatory feasibility:

- Compliance with Laws and Regulations: Identify the laws, regulations, and legal requirements that are relevant to the project or system. This may include industry-specific regulations, data protection laws, consumer protection laws, intellectual property rights, labor laws, or any other applicable legal frameworks.
- Data Protection and Privacy: Assess the requirements and obligations related to data protection and privacy. Determine if the proposed system collects, stores, or processes personal data and ensure compliance with relevant data protection laws, such as the General Data Protection Regulation (GDPR) in the European Union or the California Consumer Privacy Act (CCPA) in the United States.
- Intellectual Property Rights: Evaluate the potential implications on intellectual property rights, including copyright, trademarks, patents, or trade secrets. Ensure that the system does not infringe on the rights of third parties and take necessary measures to protect the system's intellectual property.
- Electronic Transactions and Contracts: Determine if the system involves electronic transactions or contracts. Assess the legal requirements related to electronic signatures, contract formation, and the legal enforceability of electronic records. Ensure compliance with applicable laws and regulations governing electronic transactions.
- Payment Regulations: If the system involves payment processing or financial transactions, consider the legal and regulatory requirements related to payment security, financial regulations, and compliance with payment card industry standards (such as the Payment Card Industry Data Security Standard - PCI DSS).

- **Licensing and Permits:** Determine if the operation of the system requires any specific licenses, permits, certifications, or approvals. Ensure compliance with applicable regulations and obtain the necessary legal authorizations from relevant authorities.
- **Consumer Protection:** Evaluate the legal requirements and regulations related to consumer protection. Ensure that the system provides transparency, fair practices, and protection for consumers. Comply with laws governing refund policies, terms and conditions, warranties, and consumer rights.
- **Jurisdictional Considerations:** Consider the legal and regulatory differences across jurisdictions if the system is intended for use in multiple regions or countries. Ensure compliance with the laws and regulations specific to each jurisdiction where the system will be deployed or accessed.
- **Legal Liability and Limitations:** Assess potential legal liabilities and limitations associated with the system's operation. Implement necessary disclaimers, terms of service, and limitation of liability clauses to protect the organization from legal risks.
- It is crucial to consult legal experts, regulatory authorities, or legal counsel familiar with the specific legal requirements and industry regulations applicable to the project. They can provide accurate and up-to-date guidance on legal and regulatory compliance to ensure the project's feasibility from a legal perspective.

- **Data Protection and Privacy:** Evaluate the legal and regulatory requirements related to data protection and privacy.

- **5.6 Operational Feasibility**

- Operational feasibility refers to the assessment of whether a proposed project or system can be effectively implemented and integrated into the existing operations and processes of an organization. It involves evaluating the practicality, suitability, and impact of the tailor booking management system on day-to-day operations. Here are some key factors to consider when assessing operational feasibility:

- **Process Alignment:** Evaluate how well the tailor booking management system aligns with existing operational processes and workflows. Identify any gaps or potential challenges in integrating the system into current practices. Determine the level of effort required to adapt or reengineer processes to accommodate the new system.
- **Resource Availability:** Assess the availability of resources required to implement and operate the tailor booking management system. Consider factors such as skilled personnel, IT infrastructure, hardware, software, and training requirements. Determine if the necessary resources are accessible and if the organization can adequately allocate them to support the system.
- **System Compatibility:** Evaluate the compatibility of the tailor booking management system with existing hardware, software, and IT infrastructure. Consider factors such as operating systems, databases, networking protocols, and integration capabilities. Determine if the system can seamlessly integrate with existing technologies and if any system upgrades or replacements are required.
- **Training and User Adoption:** Assess the training needs and readiness of the organization to adopt the new system. Identify the knowledge and skills gap that may exist among users and determine the level of training and support required for successful implementation.

## 6. Description of the selected Process Model

### **Waterfall Model**

It is comprised of the stages that the developer will use when developing the system. It is a sequential model hence the name waterfall. The developer has to finish with one stage before going to the next one. It comprises of the feasibility study, analysis phase, design phase, coding phase, testing phase, implementation phase and finally the maintenance phase. It is a simple model and easy to use and understand. With waterfall development based methodologies, the analysts and users proceed sequentially from one phase to the next. The deliverables from each phase are voluminous and are presented to the project sponsor for approval as the project moves from phase to phase. Once the phase is approved by the sponsor it ends and the next phase begins.

## 6.1 Cause of your selection

The waterfall model is a sequential software development process that follows a linear and systematic approach. It consists of distinct phases, including requirements gathering, design, implementation, testing, deployment, and maintenance. The key reasons for using the waterfall model are:

**Simplicity:** The waterfall model is straightforward and easy to understand. Its linear nature allows for a clear progression from one phase to another, making it ideal for projects with well-defined and stable requirements.

**Clear Documentation:** Each phase in the waterfall model requires detailed documentation before moving to the next phase. This documentation ensures that all requirements, designs, and test plans are thoroughly documented, which can be beneficial for future reference and maintenance.

**Project Planning and Management:** The waterfall model allows for better project planning and management as each phase has specific deliverables and milestones. This enables project managers to estimate project timelines, allocate resources, and monitor progress more effectively.

**Early Detection of Issues:** The waterfall model emphasizes completing each phase before moving on to the next. This sequential approach allows for early detection and resolution of issues within each phase, minimizing the risk of problems being carried over to subsequent phases.

**Well-defined Requirements:** The waterfall model works well when the requirements are well-defined and unlikely to change significantly during the project. It is particularly suitable for projects with stable and predetermined objectives.

**Client and Stakeholder Involvement:** The waterfall model allows clients and stakeholders to have a clear understanding of the project's progress and milestones. Their involvement is typically more prominent during the early phases, such as requirements gathering and design, where they can provide valuable feedback and validation.

**Regulatory and Compliance Requirements:** In certain industries, projects must adhere to strict regulatory and compliance requirements. The waterfall model's emphasis on documentation, traceability, and thorough testing can help ensure compliance with industry standards and regulations.

## 7. Conclusion

The online Tailoring system will ease the work of clients by allowing them to send their measurement online thus cutting on transport expenses and time. It will ease communication between the tailor and the client and also to access each other.

It provides information about the cost, the fabric type the customer want his/her dress knit from, the urgency at which a customer wants the dress finished, the type of material to be used, quantity in terms of pairs needed and most importantly, the system computes the total cost and avails that information to the customer.

Therefore, this system will be more beneficial to implement.

## 8. References

*I have finding the above project through enternet searching and by combining many querries of clients to the tailor other is from readingdiferent books of software engeneering written by diferent engeneer,and from my flied of home where the almost tailor they prefer to use form method in their interprise.*

## CHAPTER 2: DATABASE DESIGN

### DATA BASE OF THE SYSTEM

#### DATABASE AND TECHNOLOGY

**Q1.**

**all entities and their corresponding attributes.**

**Customer:**

Customer ID

Name

Contact

Order History

Preferred Fabric Choices

Payment Information

**Order:**

Order ID

Customer ID

Date and Time of Order

Type of Clothing (shirt, trousers, dress, etc.)

Fabric Selection

Measurements

Order Status (in progress, completed, delivered)

Delivery Date

**Inventory Item:**

Item ID

Item Name (fabric, buttons, zippers, etc.)

Description

Quantity in Stock

Supplier Information

Purchase Price

Selling Price

**Appointment:**

Appointment ID

Customer Details

Date and Time

Assigned Tailor/Staff

Status (scheduled, completed, canceled)

**Supplier:**

SupplierID

Supplier Name

Contact Information

**Invoice:**



Invoice ID

Customer Details

Order Details

Total Amount

Payment Status

Due Date

**Fabric :**

Fabric ID

Fabric Type (cotton, silk, etc.)

Fabric Color

Fabric Pattern/Design

Fabric Price per Meter/Yard

**Employee/Staff:**

Staff ID

Name

Contact Information

Role/Position (tailor, manager, receptionist, etc.)

Working Hours

Tasks/Responsibilities

Date and Time

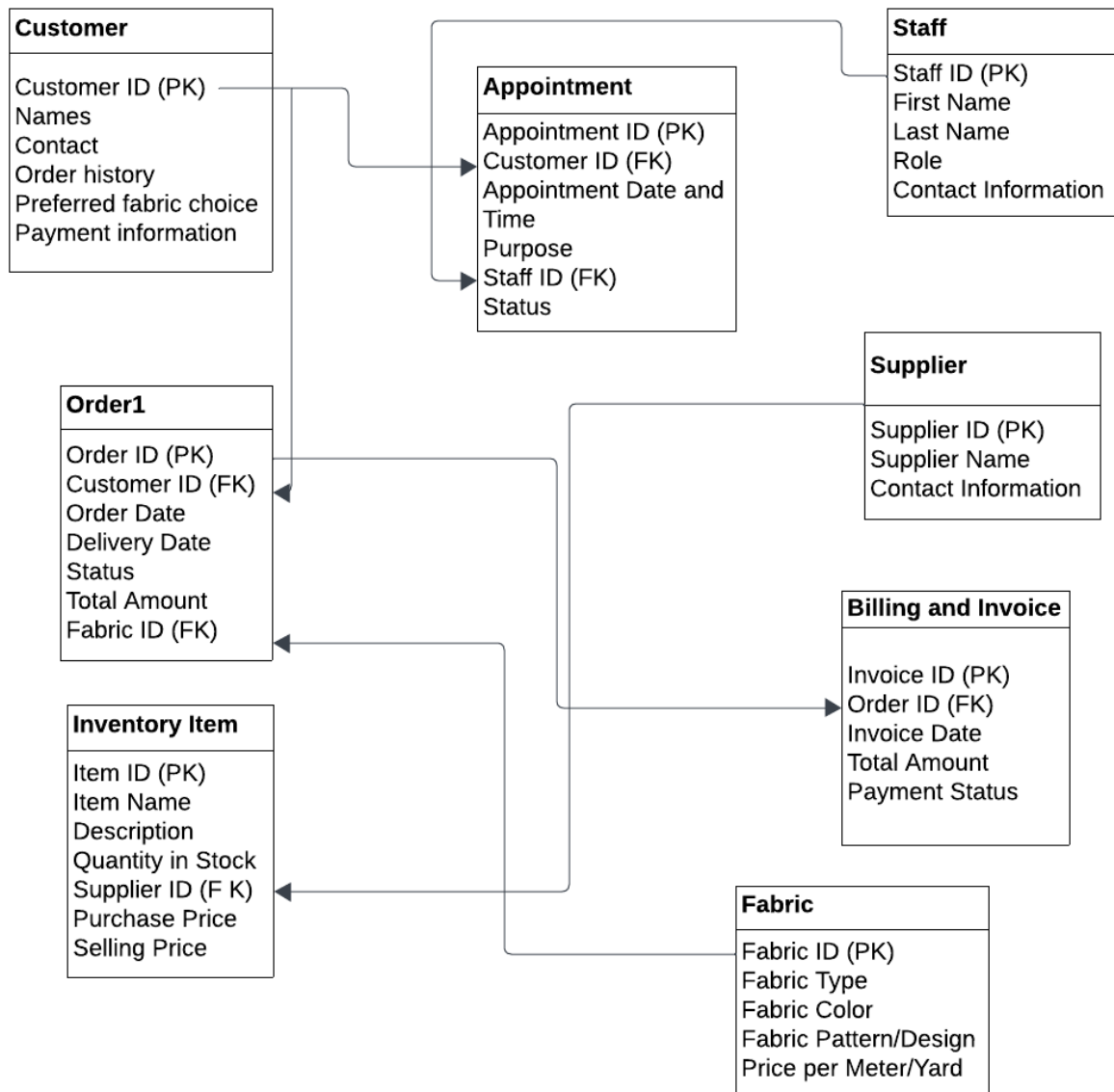
**Q2.An LDM OF Entities.**

**Relationships:**

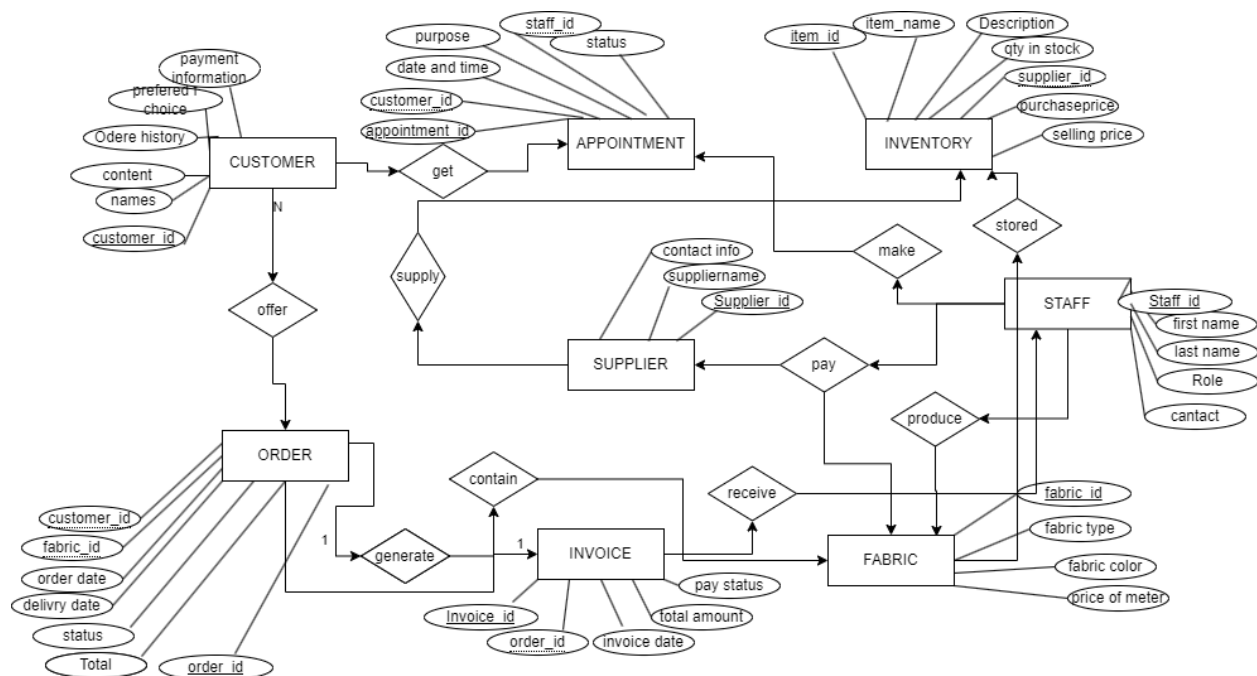
- One Customer can have multiple Measurement Profiles.
- One Customer can place multiple Orders.
- Each Order is associated with one Fabric.
- Each Inventory Item is associated with one Supplier.

- One Appointment is made by one Customer.
- One Appointment is assigned to one Staff.
- One Staff member can have multiple Appointments.
- One Order can generate one Invoice.

### LOGICAL DATA MODEL OF ONLINE TAILORING MANAGEMENT SYSTEM



**Q3. An ERD.**



## SECTION II

**Q1.**CREATE DATABASE online tailoring management system;

**Q2.**

**queries to create all the tables and relationships .**

```
CREATE TABLE customer (`customer_id` INT(34) PRIMARY KEY AUTO_INCREMENT , `names` VARCHAR(50) NOT NULL , `contact` INT(10) NOT NULL , `order history` VARCHAR(100) NOT NULL , `prefered fabric choice` VARCHAR(45) NOT NULL , `payment information` VARCHAR(100) NOT NULL );
```

```
CREATE TABLE `fabric` (`fabric` INT(23) PRIMARY KEY AUTO_INCREMENT , `fabric type` VARCHAR(34) NOT NULL , `fabric color` VARCHAR(23) NOT NULL , `fabric design` VARCHAR(45) NOT NULL , `price per meter` INT(5) NOT NULL );
```

```
CREATE TABLE `staff` (`staff_id` INT(15) PRIMARY KEY AUTO_INCREMENT , `first_name` VARCHAR(16) NOT NULL , `last_name` VARCHAR(15) NOT NULL , `role` VARCHAR(34) NOT NULL , `contact_information` INT(34) NOT NULL );
```

	customer_id	names	contact	order_history	preferred_fabric_choice	payment_informa
<input type="checkbox"/> Edit Copy Delete	5	jefg	234	dej	cotton	wenfo
<input type="checkbox"/> Edit Copy Delete	7	Kamana	786543213	old	silver	bpr
<input type="checkbox"/> Edit Copy Delete	8	samuel	785423123	first	cottle	momo
<input type="checkbox"/> Edit Copy Delete	9	rwema	97654321	12a	blue	ecobank

```
CREATE TABLE `supplier` (`supplier_id` INT(12) PRIMARY KEY AUTO_INCREMENT ,
`supplier_name` VARCHAR(34) NOT NULL , `contact_information` INT(12) NOT NULL );
```

```
CREATE TABLE order1 (order_id INT(20) PRIMARY KEY AUTO_INCREMENT , `customer_id`
INT(34), `order date` INT(50) NOT NULL , `delivery date` INT(45) NOT NULL , `status`
VARCHAR(21) NOT NULL , `total amount` INT(15) NOT NULL , `fabric_id` INT(34) NOT NULL,
FOREIGN KEY (customer_id) REFERENCES customer (customer_id), FOREIGN KEY (fabric_id)
REFERENCES fabric (fabric_id));
```

Extra options

supplier_id	supplier_name	contact_information
1	Hirwa	783832520

```
CREATE TABLE inventory item (item_id INT(12) PRIMARY KEY AUTO_INCREMENT, `item name`
VARCHAR(13) NOT NULL , `description` VARCHAR(15) NOT NULL , `quantity in stock` INT(20)
NOT NULL , `supplier_id` INT(14) NOT NULL , `purchasing price` INT(21) NOT NULL , `selling
price` INT(10) NOT NULL ,FOREIGN KEY (supplier_id) REFERENCES supplier (supplier_id)) ;
```

```
CREATE TABLE appointment (appointment_id INT(34) PRIMARY KEY AUTO_INCREMENT ,
`customer_id` INT(21) NOT NULL , `appointment date` INT(9) NOT NULL , `purpose`
VARCHAR(23) NOT NULL , `supplier_id` INT(12) NOT NULL , `status` VARCHAR(41) NOT NULL ,
FOREIGN KEY (customer_id) REFERENCES customer (customer_id), FOREIGN KEY (staff_id)
REFERENCES staff (staff_id));
```

←T→		appointment_id	customer_id	appointment_date	purpose	supplier_id	status
<input type="checkbox"/>	Edit  Copy  Delete	4	2	2.3.2023	dealing	4	pending
<input type="checkbox"/>	Edit  Copy  Delete	5	7	4.4.2021	BASE DRESS	9	PEND

☐ Check all    With selected: Edit Copy Delete Export

```
CREATE TABLE billing and invoice (invoice_id` INT(12) PRIMARY KEY AUTO_INCREMENT ,
`order_id` INT(12) NOT NULL , `invoice issued date` INT(15) NOT NULL , `total amount` INT(14)
NOT NULL , `payment status` VARCHAR(30) NOT NULL , FOREIGN KEY (order_id) REFERENCES
order (order_id));
```

**Q3.**

### Inserting data into the Customer table

```
INSERT INTO `customer` (`customer_id`, `names`, `cantact`, `order history`, `preferred fabric
choice`, `payment information`) VALUES (NULL, 'MANZI', '0786543212', 'scheduled', 'cotton',
'bk');
```

#### **Inserting data into the Order table**

```
INSERT INTO `order1` (`order_id`, `customer_id`, `order date`, `delivery date`, `status`, `total amount`, `fabric_id`) VALUES (NULL, '1', '2022-03-03', '2023-01-12', 'pending', '2300', '1');
```

#### **Inserting data into the InventoryItem table**

```
INSERT INTO `inventory item` (`item_id`, `item name`, `description`, `quantity in stock`, `supplier_id`, `purchasing price`, `selling price`) VALUES (NULL, 'cotton', 'smith cotton', '12', '1', '300', '500');
```

#### **Inserting data into the Supplier table**

```
INSERT INTO `supplier` (`supplier_id`, `supplier_name`, `contact_information`) VALUES (NULL, 'Hirwa', '0783832520');
```

#### **Inserting data into the Staff table**

```
INSERT INTO `staff` (`staff_id`, `first_name`, `last_name`, `role`, `contact_information`) VALUES (NULL, 'Nshuti', 'Elie', 'tailor', '0783832520');
```

#### **Inserting data into the BillingAndInvoice table**

```
INSERT INTO `biling and invoice` (`invoice_id`, `order_id`, `invoice issued date`, `total amount`, `payment status`) VALUES (NULL, '2', '1998-07-09', '20', 'equity bank');
```

#### **Inserting data into the Fabric table**

```
INSERT INTO `staff` (`staff_id`, `first_name`, `last_name`, `role`, `contact_information`) VALUES (NULL, 'Nshuti', 'Elie', 'tailor', '0783832520');
```

#### **Inserting data into the Appointment table**

```
INSERT INTO `appointment` (`appointment_id`, `customer_id`, `appointment date`, `purpose`, `supplier_id`, `status`) VALUES (NULL, '2', '2019-22-04', 'pay', '3', 'finished');
```

#### **Q4.**

#### **queries to display all the information in tables.**

```
SELECT *FROM Customer;
```

```
SELECT *FROM Order1;
```

```
SELECT *FROM Fabric;
```

```
SELECT *FROM Supplier;
```

```
SELECT *FROM Staff;
```

```
SELECT *FROM Appointment;
```

SELECT \*FROM Billing and Invoice;

SELECT \*FROM Inventory Item;

**Q5.**

**query to update information in table of customer.**

UPDATE `customer` SET ,`names`='elie',`cantact`='0783832520',`order history`='scheduled',`prefered fabric choice`='cotton',`payment information`='bk' WHERE customer\_id=1;

**query to update information in table of staff.**

UPDATE `staff` SET  
`first\_name`='Nshuti',`last\_name`='Elie',`role`='Tailor',`contact\_information`='0783832520'  
WHERE `staff\_id`=1;

**Section III.**

**Q1.**

**a view to insert data into tables.**

**Customer view:** create view customer\_view as SELECT \* FROM `customer`;

**Order view:** create view order\_view As select \* from 'order';

**Appointment view:** create view Appointment\_view As SELECT \* FROM `appointment`;

**Fabric view:** create view fabric\_view as select \* from fabric;

**Supplier view:** create view supplier\_view As SELECT \* FROM `supplier`;

**Staff view:** create view Staff\_view As SELECT \* FROM `staff`;

**Billing and Invoice view:** create view billinginvoice\_view As SELECT \* FROM `biling and invoice`;

**Inventory\_item view:** create view inventory\_item\_view As SELECT \* FROM `inventory item`;

**Q2.**

**a view to display all the information in tables.**

**CREATE VIEW Customer AS**

**SELECT**

Customer\_id,names,contact,order history,preferred fabric choice,payment information **FROM**

Customer C;

### Q3

**Create an update view for the Customer table**

**CREATE VIEW** UpdatableCustomer **AS**

**SELECT**

Customer ID ,

Names,

Contact,

Order history,

Preferred fabric choice,

Payment information

**FROM** Customer;

**Create an updat view for the Supplier table**

**CREATE VIEW** UpdatableSupplier **AS**

**SELECT** Supplier ID ,

Supplier Name,

Contact Information

**FROM** Supplier

### Q4.

**A view to deletion data from Customer table:**

**CREATE VIEW** DeleteCustomer **AS SELECT** customer\_id **FROM** Customer;

**A view to deletion data from Order table:**

**CREATE VIEW** DeleteOrder1 **AS SELECT** order\_id **FROM** order1;

### Q5.

**Create one view of your choice that considers sub-query.**

**CREATE VIEW** CustomerOrderSummary **AS SELECT\*** **FROM** Customer **WHERE**  
Customer\_id=(SELECT customer\_id from customer where names = "MANZI");

## Section IV.

### Q1.

a stored procedure to insert data into tables.

#### Stored procedure of customer

```
CREATE PROCEDURE InsertCustomerInfo( IN customer_id INT, IN names VARCHAR(255), IN contact VARCHAR(255), IN order_history TEXT, IN preferred_fabric_choice VARCHAR(255), IN payment_information TEXT ) BEGIN INSERT INTO Customer (customer_id, FirstName, LastName, PhoneNumber, Email, Address, PreferredFabric, PaymentInformation) VALUES ( customer_id, SUBSTRING_INDEX(names, ' ', 1), Assuming names are first name and last name separated by a space SUBSTRING_INDEX(names, ' ', -1), SUBSTRING_INDEX(contact, ',', 1), Assuming contact information is comma-separated SUBSTRING_INDEX(SUBSTRING_INDEX(contact, ',', -2), ',', 1), Email SUBSTRING_INDEX(contact, ',', -1), Address preferred_fabric_choice, payment_information ); END;
```

```
CALL InsertCustomerInfo(
    12345,          customer_id
    'John Doe',     names
    0783238520,    contact
    'Order123,Order456', order_history
    'Cotton Fabric', preferred_fabric_choice
    'Credit Card: 1234' payment_information
);
```

#### Stored procedure of order

```
CREATE PROCEDURE InsertOrderInfo( IN order_id INT, IN customer_id INT, IN order_date DATE, IN delivery_date DATE, IN status VARCHAR(255), IN total_amount DECIMAL(10, 2), IN fabric_id INT ) BEGIN INSERT INTO Order1 (order_id, customer_id, order_date, delivery_date, status, total_amount, fabric_id) VALUES ( order_id, customer_id, order_date, delivery_date, status, total_amount, fabric_id ); END;
```

```
CALL InsertOrderInfo(
    12345,          order_id
    5678,           customer_id
```



'2023-09-15',      order\_date (in 'YYYY-MM-DD' format)  
'2023-09-20',      delivery\_date (in 'YYYY-MM-DD' format)  
'Pending',          status  
125.99,              total\_amount  
9876                  fabric\_id);

**Q2.**

**Create a stored procedure to display all the information in your tables.**

**CREATE PROCEDURE** DisplayAllData()

**BEGIN**

**SELECT \* FROM** Customers;

**SELECT \* FROM** Orders;

**SELECT \* FROM** Appointment;

**SELECT \* FROM** supplier;

**SELECT \* FROM** staff;

**SELECT \* FROM** billing and invoice;

**SELECT \* FROM** Fabric;

**SELECT \* FROM** Inventory item;

**END**

**Q3.**

**A stored procedure to update information in any of the two tables .**

IN p\_supplier\_id INT,

IN p\_supplier\_name VARCHAR(255),

IN p\_contact\_information VARCHAR(255)

)

**BEGIN**

Check if the supplier exists

```

DECLARE supplierCount INT;

SET supplierCount = (SELECT COUNT(*) FROM Supplier WHERE supplier_id = p_supplier_id);

IF supplierCount > 0 THEN
    Update supplier information
    UPDATE Supplier
    SET
        supplier_name = p_supplier_name,
        contact_information = p_contact_information
    WHERE
        supplier_id = p_supplier_id;

    SELECT 'Supplier information updated successfully' AS Message;
ELSE
    SELECT 'Supplier does not exist. No updates were made.' AS Message;
END IF;
END;

//

DELIMITER ;

CALL UpdateSupplierInfo(
    123,          supplier_id of the supplier you want to update
    'Updated Supplier Name', new supplier_name
    'Updated Contact Info' new contact_information
);

Q4.

```

**a stored procedure to delete data in table OF FABRIC according to any simple condition .**

```
CREATE PROCEDURE DeleteFabricByCondition( IN condition_column VARCHAR(255), IN condition_value VARCHAR(255) ) BEGIN Delete data from the Fabric table based on the provided condition DELETE FROM Fabric WHERE condition_column = condition_value; END;
```

```
CALL DeleteFabricByCondition('fabric_type', 'Cotton');
```

**a stored procedure to delete data in table OF Billing and invoice according to any simple condition .**

```
CREATE PROCEDURE DeleteBillingAndInvoiceByCondition( IN condition_column VARCHAR(255), IN condition_value VARCHAR(255) ) BEGIN Delete data from Billing table based on the condition DELETE FROM Billing WHERE condition_column = condition_value; Delete data from Invoice table based on the condition DELETE FROM Invoice WHERE condition_column = condition_value; END;
```

```
CALL DeleteBillingAndInvoiceByCondition('payment_status', 'Unpaid');
```

**Q5.**

**A stored the procedure view customer and order1 that considers sub-query**

```
CREATE PROCEDURE CreateCustomerOrdersView() BEGIN Create a view that shows customer information along with total order count CREATE OR REPLACE VIEW CustomerOrders AS SELECT c.customer_id, CONCAT(c.first_name, ' ', c.last_name) AS customer_name, c.email, c.phone_number, o.total_orders FROM Customers c LEFT JOIN ( SELECT customer_id, COUNT(*) AS total_orders FROM Orders GROUP BY customer_id ) o ON c.customer_id = o.customer_id; END;
```

```
CALL CreateCustomerOrdersView();
```

## **Section V**

### **Q1**

**Create after inserting triggers of Supplier tables .**

```
CREATE TRIGGER AfterInsertSupplier AFTER INSERT ON supplier FOR EACH ROW BEGIN You can include actions to perform after an insert here. For example, you can send notifications or update related tables. This is a placeholder, and you can customize it as needed. INSERT INTO supplier_log (event_description, supplier_id, supplier_name, contact_information) VALUES ('New supplier created', NEW.supplier_id, NEW.supplier_name, NEW.contact_information); END;
```

**Create after inserting triggers of staff tables .**

**CREATE TRIGGER** AfterInsertStaff **AFTER INSERT ON** staff **FOR EACH ROW BEGIN** You can include actions to perform after an insert here. For example, you can send notifications or log the event. This is a placeholder, and you can customize it as needed. **INSERT INTO** staff\_log (event\_description, staff\_id, first\_name, last\_name) **VALUES** ('New staff member added', NEW.staff\_id, NEW.first\_name, NEW.last\_name); **END**;

**Q2.**

**Create after updating triggers of any two tables**

after-update triggers of staff.

**CREATE TRIGGER** AfterUpdateStaff **AFTER UPDATE ON** staff **FOR EACH ROW BEGIN** You can include actions to perform after an update here. For example, you can log the update or send notifications. -- This is a placeholder, and you can customize it as needed. **INSERT INTO** staff\_update\_log (event\_description, staff\_id, new\_first\_name, new\_last\_name) **VALUES** ('Staff information updated', NEW.staff\_id, NEW.first\_name, NEW.last\_name); **END**;

**after-update triggers of supplier.**

**CREATE TRIGGER** AfterUpdateSupplier **AFTER UPDATE ON** supplier **FOR EACH ROW BEGIN** You can include actions to perform after an update here. For example, you can log the update or send notifications. -- This is a placeholder, and you can customize it as needed. **INSERT INTO** supplier\_update\_log (event\_description, supplier\_id, new\_supplier\_name, new\_contact\_information) **VALUES** ('Supplier information updated', NEW.supplier\_id, NEW.supplier\_name, NEW.contact\_information); **END**;

**Q3.**

**Create after deleting triggers of any two tables**

**after deleting triggers table of staff.**

**CREATE TRIGGER** AfterDeleteStaff **AFTER DELETE ON** staff **FOR EACH ROW BEGIN** You can include actions to perform after a delete here. For example, you can log the deletion or send notifications. -- This is a placeholder, and you can customize it as needed. **INSERT INTO** staff\_delete\_log (event\_description, deleted\_staff\_id, deleted\_first\_name, deleted\_last\_name) **VALUES** ('Staff member deleted', OLD.staff\_id, OLD.first\_name, OLD.last\_name); **END**;

**after deleting triggers table of Supplier.**

**CREATE TRIGGER** AfterDeleteSupplier **AFTER DELETE ON** supplier **FOR EACH ROW BEGIN** You can include actions to perform after a delete here. For example, you can log the deletion or send notifications. This is a placeholder, and you can customize it as needed. **INSERT INTO** supplier\_delete\_log (event\_description, deleted\_supplier\_id, deleted\_supplier\_name,

deleted\_contact\_information) **VALUES** ('Supplier deleted', OLD.supplier\_id, OLD.supplier\_name, OLD.contact\_information); **END**;

## **Section VI**

### **Section VI**

**Q1.**

**Create a user with your name as username and your student number as password and grant all privileges to the created user.**

**ANSWER:**

```
CREATE USER 'NSENGIMANA' '@'localhost IDENTIFIED BY '222008235';  
GRANT ALL PRIVILEGES ON *.* TO 'NSENGIMANA' '@'localhost WITH GRANT OPTION;  
FLUSH PRIVILEGES;
```

**Q2.**

**Create a user with your "names\_semi" as username and your student number as password and give him insert, update, and delete privileges to the created user**

```
CREATE USER 'Nsengimana_Olivier' '@'localhost IDENTIFIED BY '222008235';  
GRANT INSERT, UPDATE, DELETE ON online_tailoring_management_system.* TO 'Nsengimana-OLIVER' '@'localhost;  
FLUSH PRIVILEGES;
```

**Q3.**

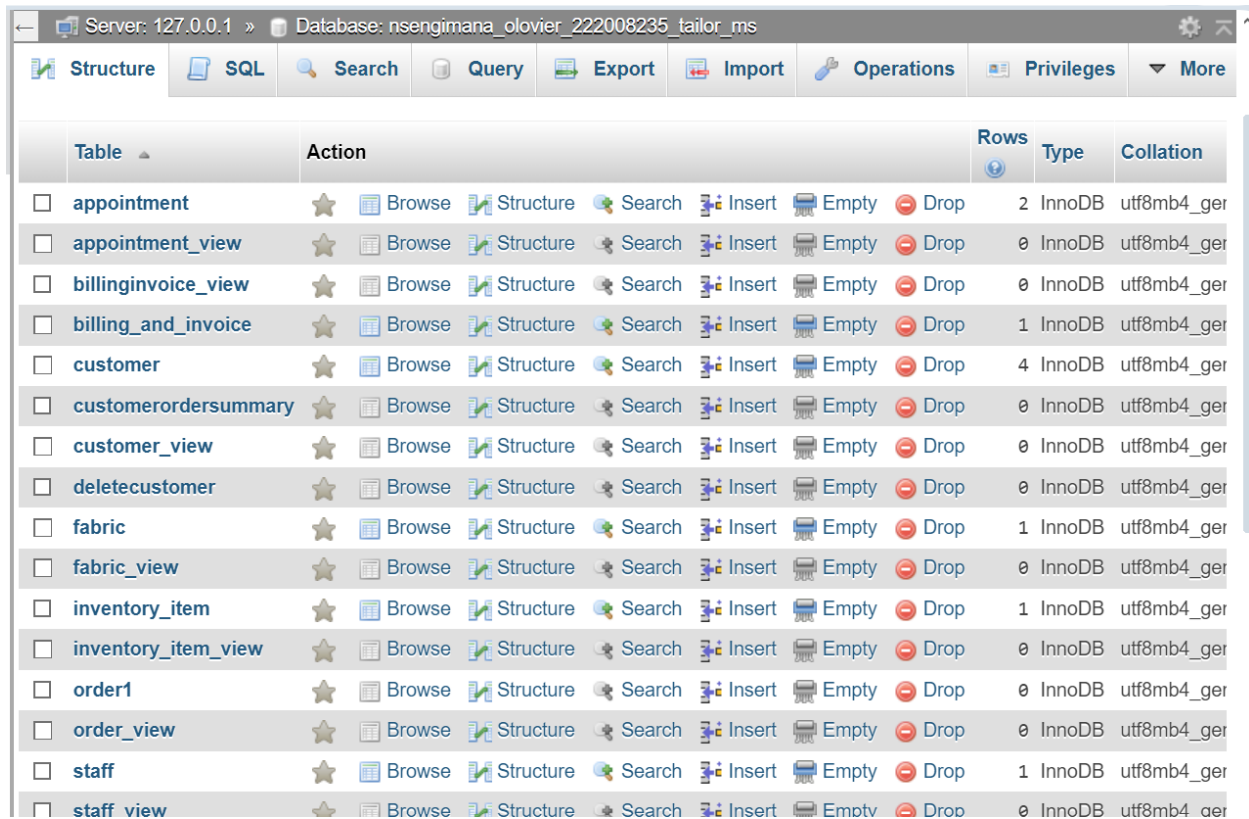
**GRANT insert privileges to the last user you created.**

```
GRANT INSERT ON online_tailoring_management_system.* TO  
'Nsemgimana_Olivier' '@'localhost;  
FLUSH PRIVILEGES;
```

**REVOKE insert privileges to the last user you created.**

```
REVOKE INSERT ON online_tailoring_management_system.* FROM 'Nsemgimana_Olivier' '@'localhost;
```

## FLUSH PRIVILEGES;



The screenshot shows a database management interface with a menu bar (Structure, SQL, Search, Query, Export, Import, Operations, Privileges, More) and a table list. The table list has columns for Table, Action, Rows, Type, and Collation. The tables listed are appointment, appointment\_view, billinginvoice\_view, billing\_and\_invoice, customer, customerordersummary, customer\_view, deletecustomer, fabric, fabric\_view, inventory\_item, inventory\_item\_view, order1, order\_view, staff, and staff\_view. Each table row includes a checkbox, a star icon, and a set of action icons (Browse, Structure, Search, Insert, Empty, Drop).

Table	Action	Rows	Type	Collation
<input type="checkbox"/> appointment	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_ger
<input type="checkbox"/> appointment_view	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_ger
<input type="checkbox"/> billinginvoice_view	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_ger
<input type="checkbox"/> billing_and_invoice	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_ger
<input type="checkbox"/> customer	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_ger
<input type="checkbox"/> customerordersummary	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_ger
<input type="checkbox"/> customer_view	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_ger
<input type="checkbox"/> deletecustomer	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_ger
<input type="checkbox"/> fabric	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_ger
<input type="checkbox"/> fabric_view	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_ger
<input type="checkbox"/> inventory_item	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_ger
<input type="checkbox"/> inventory_item_view	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_ger
<input type="checkbox"/> order1	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_ger
<input type="checkbox"/> order_view	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_ger
<input type="checkbox"/> staff	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_ger
<input type="checkbox"/> staff_view	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_ger

## DATABASE EXPORTS.

Database online tailoring management system

Table structure for table appointment

Column	Type	Null	Default
<b>appointment_id</b>	int(34)	No	
customer_id	int(21)	No	
appointment date	date	No	
purpose	varchar(23)	No	
supplier_id	int(12)	No	
status	varchar(41)	No	

Dumping data for table appointment

1	1	0000-00-00	buy	1	pending
2	2	0000-00-00	pay	3	finished

Stand-in structure for view appointment\_view

Column	Type	Null	Default
appointment_id	int(34)	No	0
customer_id	int(21)	No	
appointment date	date	No	
purpose	varchar(23)	No	
supplier_id	int(12)	No	
status	varchar(41)	No	

Table structure for table biling and invoice

Column	Type	Null	Default
<b>invoice_id</b>	int(12)	No	
order_id	int(12)	No	
invoice issued date	date	No	
total amount	int(14)	No	
payment status	varchar(30)	No	

Dumping data for table biling and invoice

1	2	1998-07-09	20	equity bank
---	---	------------	----	-------------

Stand-in structure for view billinginvoice\_view

Column	Type	Null	Default
invoice_id	int(12)	No	0
order_id	int(12)	No	
invoice issued date	date	No	
total amount	int(14)	No	

payment status	varchar(30)	No
----------------	-------------	----

Table structure for table customer

Column	Type	Null	Default
<b>customer_id</b>	int(34)	No	
names	varchar(50)	No	
contact	int(10)	No	
order history	varchar(100)	No	
prefered fabric choice	varchar(45)	No	
payment information	varchar(100)	No	

Dumping data for table customer

1	MANZI	786543212	scheduled	cotton	bk
---	-------	-----------	-----------	--------	----

Stand-in structure for view customerordersummary

Column	Type	Null	Default
customer_id	int(34)	No	0
names	varchar(50)	No	
contact	int(10)	No	
order history	varchar(100)	No	
prefered fabric choice	varchar(45)	No	
payment information	varchar(100)	No	

Stand-in structure for view customer\_view

Column	Type	Null	Default
customer_id	int(34)	No	0
names	varchar(50)	No	
contact	int(10)	No	
order history	varchar(100)	No	
prefered fabric choice	varchar(45)	No	



payment information	varchar(100)	No
---------------------	--------------	----

Stand-in structure for view deletcustomer

Column	Type	Null	Default
customer_id	int(34)	No	0

Table structure for table fabric

Column	Type	Null	Default
<b><i>fabric_id</i></b>	int(23)	No	
fabric type	varchar(34)	No	
fabric color	varchar(23)	No	
fabric design	varchar(45)	No	
price per meter	int(5)	No	

Dumping data for table fabric

1	fitting	red	fitting inch	150
---	---------	-----	--------------	-----

Stand-in structure for view fabric\_view

Column	Type	Null	Default
fabric_id	int(23)	No	0
fabric type	varchar(34)	No	
fabric color	varchar(23)	No	
fabric design	varchar(45)	No	
price per meter	int(5)	No	

Table structure for table inventory item

Column	Type	Null	Default
<b><i>item_id</i></b>	int(6)	No	
item name	varchar(23)	No	
description	varchar(32)	No	
quantity in stock	int(12)	No	

supplier_id	int(21)	No
purchasing price	int(32)	No
selling price	int(23)	No

Dumping data for table inventory item

1	cotton	smoth cotton	12	1	300	500
---	--------	--------------	----	---	-----	-----

Stand-in structure for view inventory\_item\_view

Column	Type	Null	Default
item_id	int(6)	No	0
item name	varchar(23)	No	
description	varchar(32)	No	
quantity in stock	int(12)	No	
supplier_id	int(21)	No	
purchasing price	int(32)	No	
selling price	int(23)	No	

Table structure for table order1

Column	Type	Null	Default
<b>order_id</b>	int(20)	No	
customer_id	int(34)	Yes	NULL
order date	date	No	
delivery date	date	No	
status	varchar(21)	No	
total amount	int(15)	No	
fabric_id	int(34)	No	

Dumping data for table order1

3	1	2022-03-03	2023-01-12	pending	2300	1
---	---	------------	------------	---------	------	---

#### Stand-in structure for view order\_view

Column	Type	Null	Default
order_id	int(20)	No	0
customer_id	int(34)	Yes	NULL
order date	date	No	
delivery date	date	No	
status	varchar(21)	No	
total amount	int(15)	No	
fabric_id	int(34)	No	

#### Table structure for table staff

Column	Type	Null	Default
<b>staff_id</b>	int(15)	No	
first_name	varchar(16)	No	
last_name	varchar(15)	No	
role	varchar(34)	No	
contact_information	int(34)	No	

#### Dumping data for table staff

1	Nshuti	Elie	tailor	783832520
---	--------	------	--------	-----------

#### Triggers staff

Name	Time	Event	Definition
AfterDeleteStaff	AFTER DELETE		BEGIN -- You can include actions to perform after a delete here. -- For example, you can log the deletion or send notifications. -- This is a placeholder, and you can customize it as needed. INSERT INTO staff_delete_log (event_description, deleted_staff_id, deleted_first_name, deleted_last_name) VALUES ('Staff member deleted', OLD.staff_id, OLD.first_name, OLD.last_name); END
AfterInsertStaff	AFTER INSERT		BEGIN -- You can include actions to perform after an insert here. -- For example, you can send notifications or log the

```
event. -- This is a placeholder, and you can customize it as
needed. INSERT INTO staff_log (event_description, staff_id,
first_name, last_name) VALUES ('New staff member added',
NEW.staff_id, NEW.first_name, NEW.last_name); END
```

```
BEGIN -- You can include actions to perform after an update
here. -- For example, you can log the update or send
notifications. -- This is a placeholder, and you can customize it
```

```
AfterUpdateStaff AFTER UPDATE as needed. INSERT INTO staff_update_log (event_description,
staff_id, new_first_name, new_last_name) VALUES ('Staff
information updated', NEW.staff_id, NEW.first_name,
NEW.last_name); END
```

#### Stand-in structure for view staff\_view

Column	Type	Null	Default
staff_id	int(15)	No	0
first_name	varchar(16)	No	
last_name	varchar(15)	No	
role	varchar(34)	No	
contact_information	int(34)	No	

#### Table structure for table supplier

Column	Type	Null	Default
<b>supplier_id</b>	int(12)	No	
supplier_name	varchar(34)	No	
contact_information	int(12)	No	

#### Dumping data for table supplier

```
1      Hirwa      783832520
```

#### Triggers supplier

Name	Time	Event	Definition
AfterDeleteSupplier	AFTER	DELETE	<pre>BEGIN -- You can include actions to perform after a delete here. -- For example, you can log the deletion or send notifications. -- This is a placeholder, and you can customize it as needed. INSERT INTO supplier_delete_log</pre>

```

(event_description, deleted_supplier_id,
deleted_supplier_name, deleted_contact_information)
VALUES ('Supplier deleted', OLD.supplier_id,
OLD.supplier_name, OLD.contact_information); END

BEGIN -- You can include actions to perform after an insert
here. -- For example, you can send notifications or update
related tables. -- This is a placeholder, and you can
AfterInsertSupplier AFTER INSERT customize it as needed. INSERT INTO supplier_log
(event_description, supplier_id, supplier_name,
contact_information) VALUES ('New supplier created',
NEW.supplier_id, NEW.supplier_name,
NEW.contact_information); END

BEGIN -- You can include actions to perform after an
update here. -- For example, you can log the update or
send notifications. -- This is a placeholder, and you can
AfterUpdateSupplier AFTER UPDATE customize it as needed. INSERT INTO supplier_update_log
(event_description, supplier_id, new_supplier_name,
new_contact_information) VALUES ('Supplier information
updated', NEW.supplier_id, NEW.supplier_name,
NEW.contact_information); END

```

Stand-in structure for view [supplier\\_view](#)

Column	Type	Null	Default
supplier_id	int(12)	No	0
supplier_name	varchar(34)	No	
contact_information	int(12)	No	

Structure for view [appointment\\_view](#)

Column	Type	Null	Default
appointment_id	int(34)	No	0
customer_id	int(21)	No	
appointment date	date	No	
purpose	varchar(23)	No	
supplier_id	int(12)	No	

status	varchar(41)	No
--------	-------------	----

Structure for view [billinginvoice\\_view](#)

Column	Type	Null	Default
invoice_id	int(12)	No	0
order_id	int(12)	No	
invoice issued date	date	No	
total amount	int(14)	No	
payment status	varchar(30)	No	

Structure for view [customerordersummary](#)

Column	Type	Null	Default
customer_id	int(34)	No	0
names	varchar(50)	No	
contact	int(10)	No	
order history	varchar(100)	No	
prefered fabric choice	varchar(45)	No	
payment information	varchar(100)	No	

Structure for view [customer\\_view](#)

Column	Type	Null	Default
customer_id	int(34)	No	0
names	varchar(50)	No	
contact	int(10)	No	
order history	varchar(100)	No	
prefered fabric choice	varchar(45)	No	
payment information	varchar(100)	No	

Structure for view [deletecustomer](#)

Column	Type	Null	Default
--------	------	------	---------

customer_id	int(34)	No	0
-------------	---------	----	---

#### Structure for view fabric\_view

Column	Type	Null	Default
fabric_id	int(23)	No	0
fabric type	varchar(34)	No	
fabric color	varchar(23)	No	
fabric design	varchar(45)	No	
price per meter	int(5)	No	

#### Structure for view inventory\_item\_view

Column	Type	Null	Default
item_id	int(6)	No	0
item name	varchar(23)	No	
description	varchar(32)	No	
quantity in stock	int(12)	No	
supplier_id	int(21)	No	
purchasing price	int(32)	No	
selling price	int(23)	No	

#### Structure for view order\_view

Column	Type	Null	Default
order_id	int(20)	No	0
customer_id	int(34)	Yes	NULL
order date	date	No	
delivery date	date	No	
status	varchar(21)	No	
total amount	int(15)	No	

fabric_id	int(34)	No
-----------	---------	----

Structure for view staff\_view

Column	Type	Null	Default
staff_id	int(15)	No	0
first_name	varchar(16)	No	
last_name	varchar(15)	No	
role	varchar(34)	No	
contact_information	int(34)	No	

Structure for view supplier\_view

Column	Type	Null	Default
supplier_id	int(12)	No	0
supplier_name	varchar(34)	No	
contact_information	int(12)	No	

## CHAPTER 3: JAVA PROGRAMING

### Introduction.

So a tailor Shop Management System is a software application designed to streamline and organize various aspects of running a tailor shop or clothing store.

In this chapter I will be describing how powerful general purpose programming language was used to create the analyzed system. Under this

chapter I will explain in full all detail of how everything will function together with database that have been describe above and how it work with full analyzed system.

### Tools used to develop this system in java programming:

**Eclipse IDE:** an integrated development environment used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. It is the second-most-popular IDE for Java development.

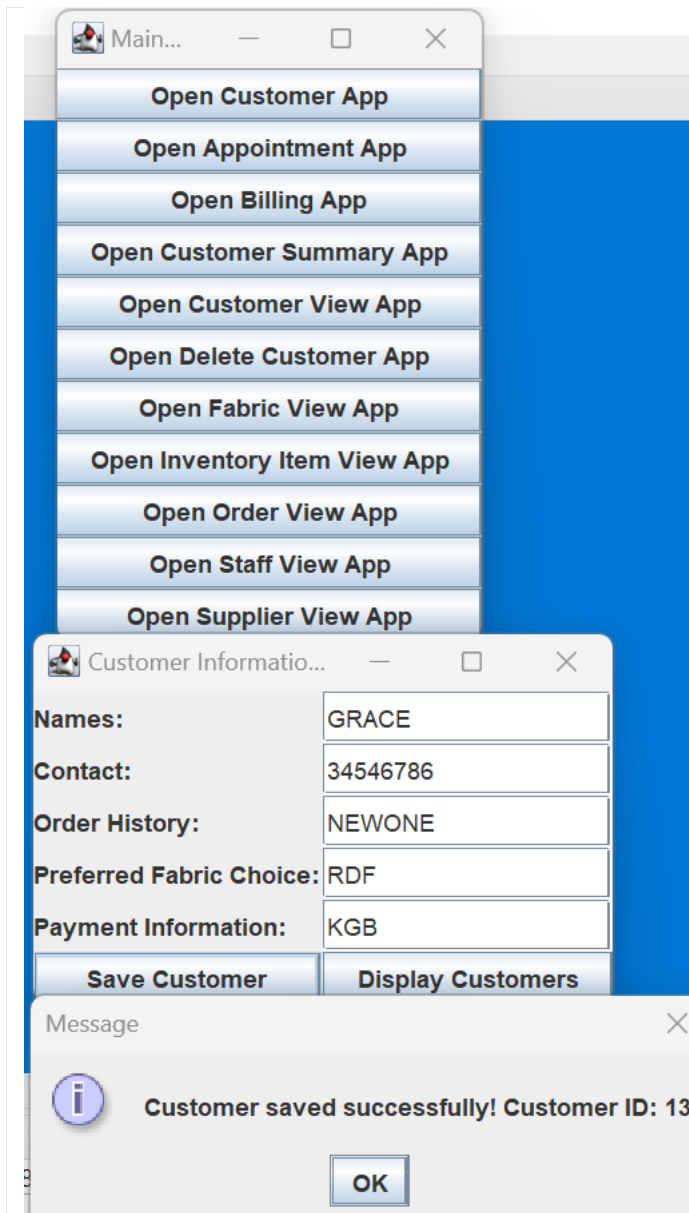


JAR stands for Java Archive. It's a file format based on the popular ZIP file format and is used for aggregating many files into one. Although JAR can be used as a general archiving tool, the primary motivation for its development was so that Java applets and their requisite components.

Let's explore the potential applications and functionalities of such a system:

1. **Customer Management:**

- The "Open Customer App" button likely opens a module for managing customer information.



- Features may include adding new customers, updating customer details, and keeping track of customer orders and preferences.

## 2. **Appointment Management:**

- The "Open Appointment App" button suggests a module for managing appointments.

Appointment Information A...

Customer ID:	2
Appointment Date:	23.1.2023
Purpose:	tai
Supplier ID:	04
Status:	active

Save Appointment    Display Appointments

Message

**i** Appointment saved successfully! Appointment ID: 6

OK

- This could include scheduling appointments for fittings or consultations with customers.

### 3. Billing and Invoicing:

- The "Open Billing App" button indicates a module for handling billing and invoicing.
- Features might include creating invoices for completed orders, tracking payments, and generating financial reports.

Billing Inform...

Order ID:	01
Issued Date:	12.12.1998
Total Amount:	1200
Payment Status:	mobile money

Save Billing    Display Billings

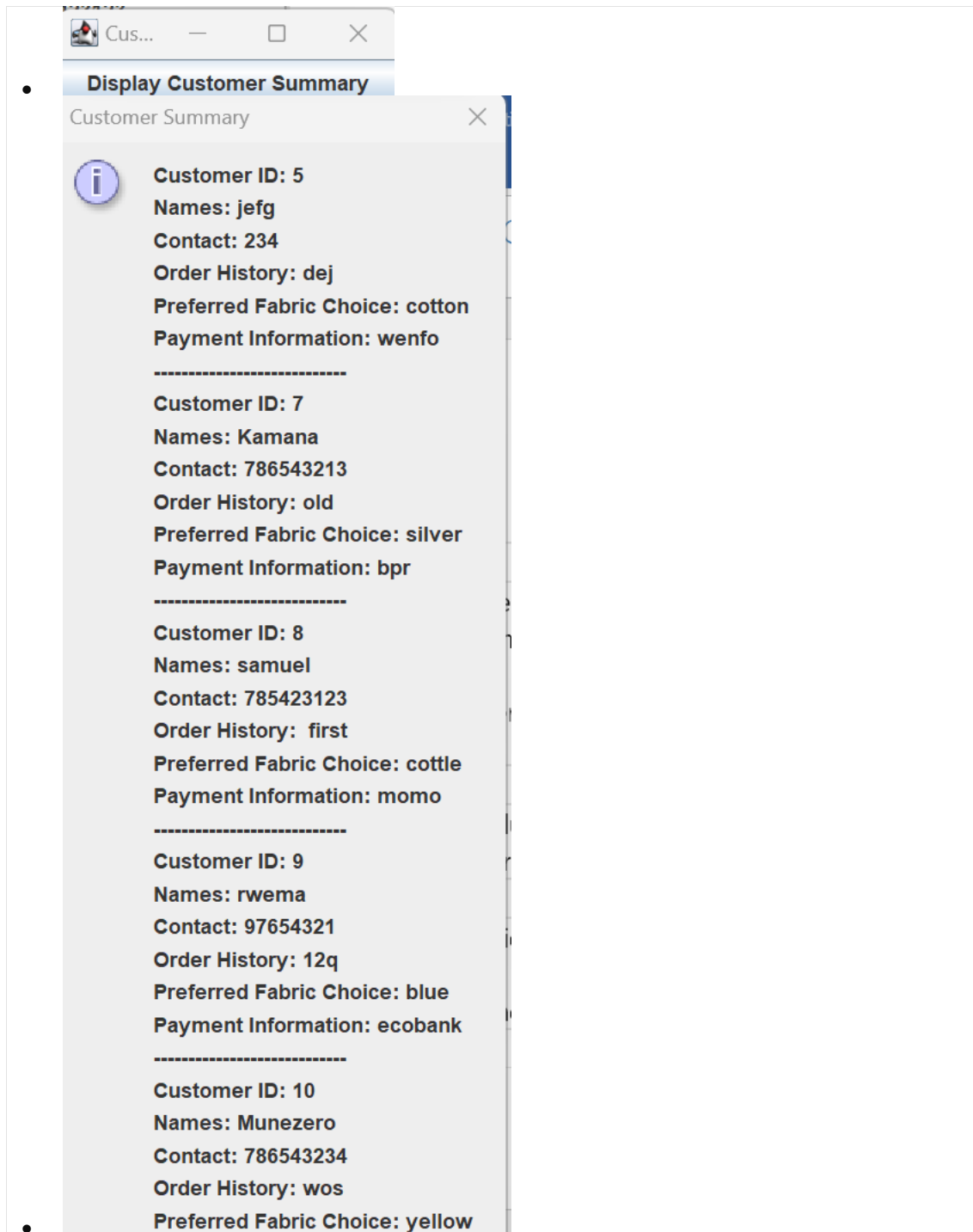
Billing Information

**i** Invoice ID: 1  
Order ID: 2  
Issued Date: 1998-07-09  
Total Amount: 20  
Payment Status: equity bank

OK

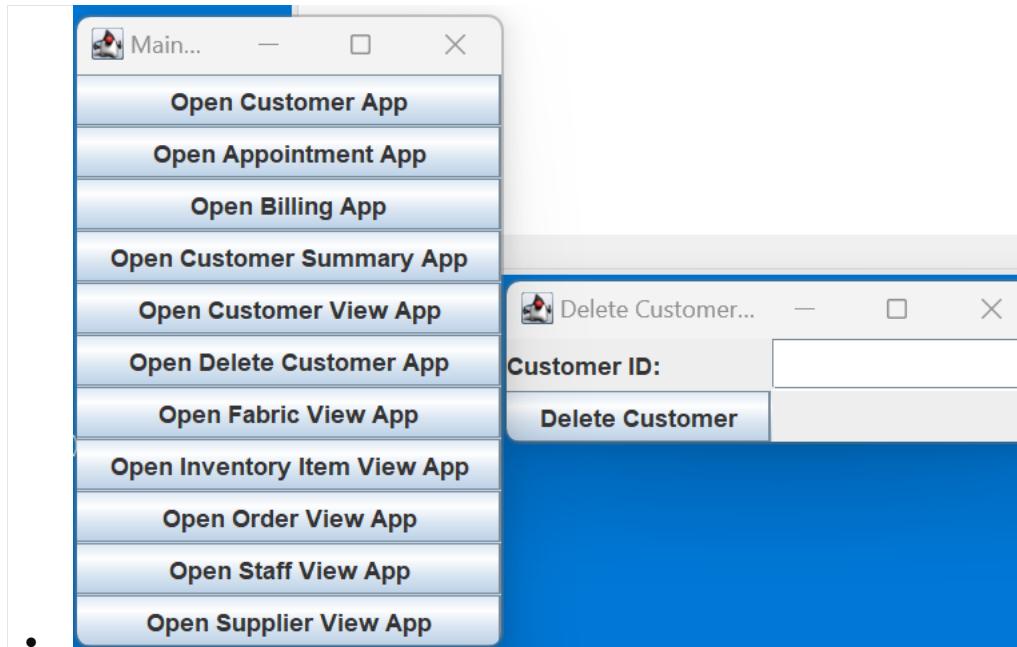
### 4. Customer Summary and Viewing:

- The "Open Customer Summary App" and "Open Customer View App" buttons might provide summary views of customer information and individual customer profiles.
- This could include order history, measurements, and other relevant details.



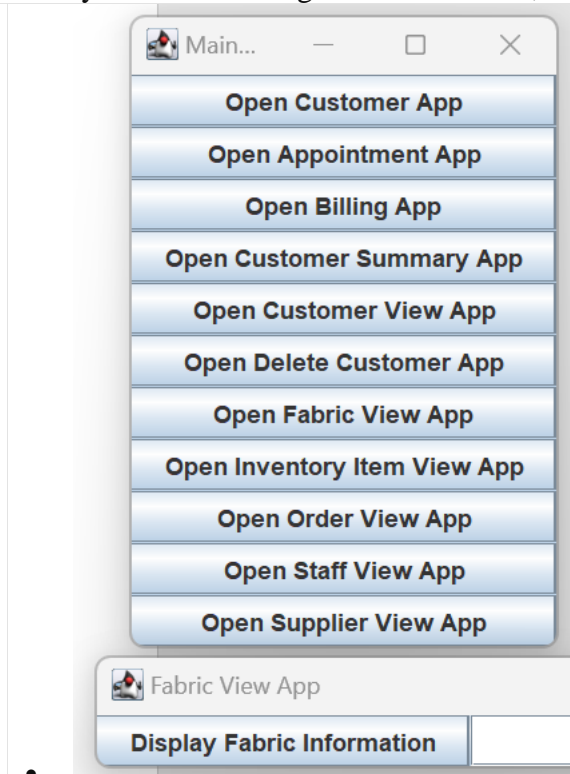
## 5. Customer Deletion:

- The "Open Delete Customer App" button suggests a module for managing customer records, including the ability to delete customer profiles.



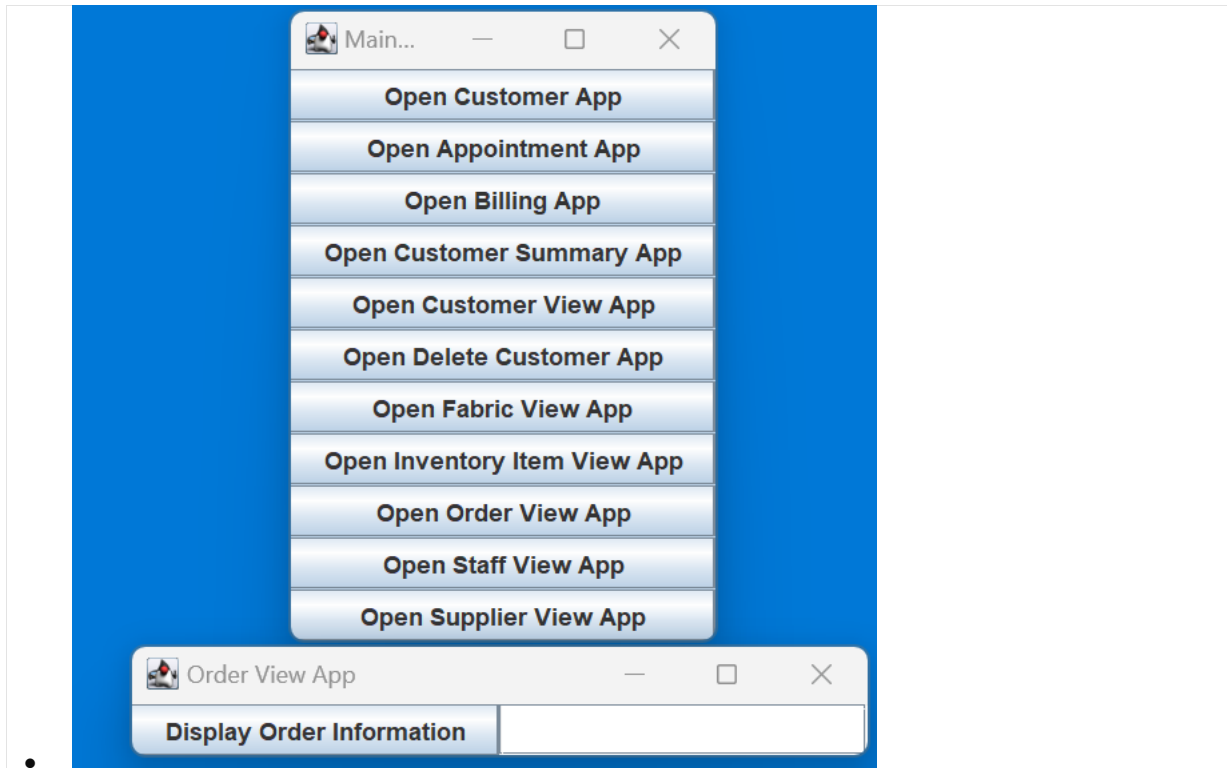
#### 6. Fabric and Inventory Management:

- The "Open Fabric View App" and "Open Inventory Item View App" buttons hint at modules for managing fabric inventory and other items.
- This may involve tracking available fabrics, accessories, and other inventory items.



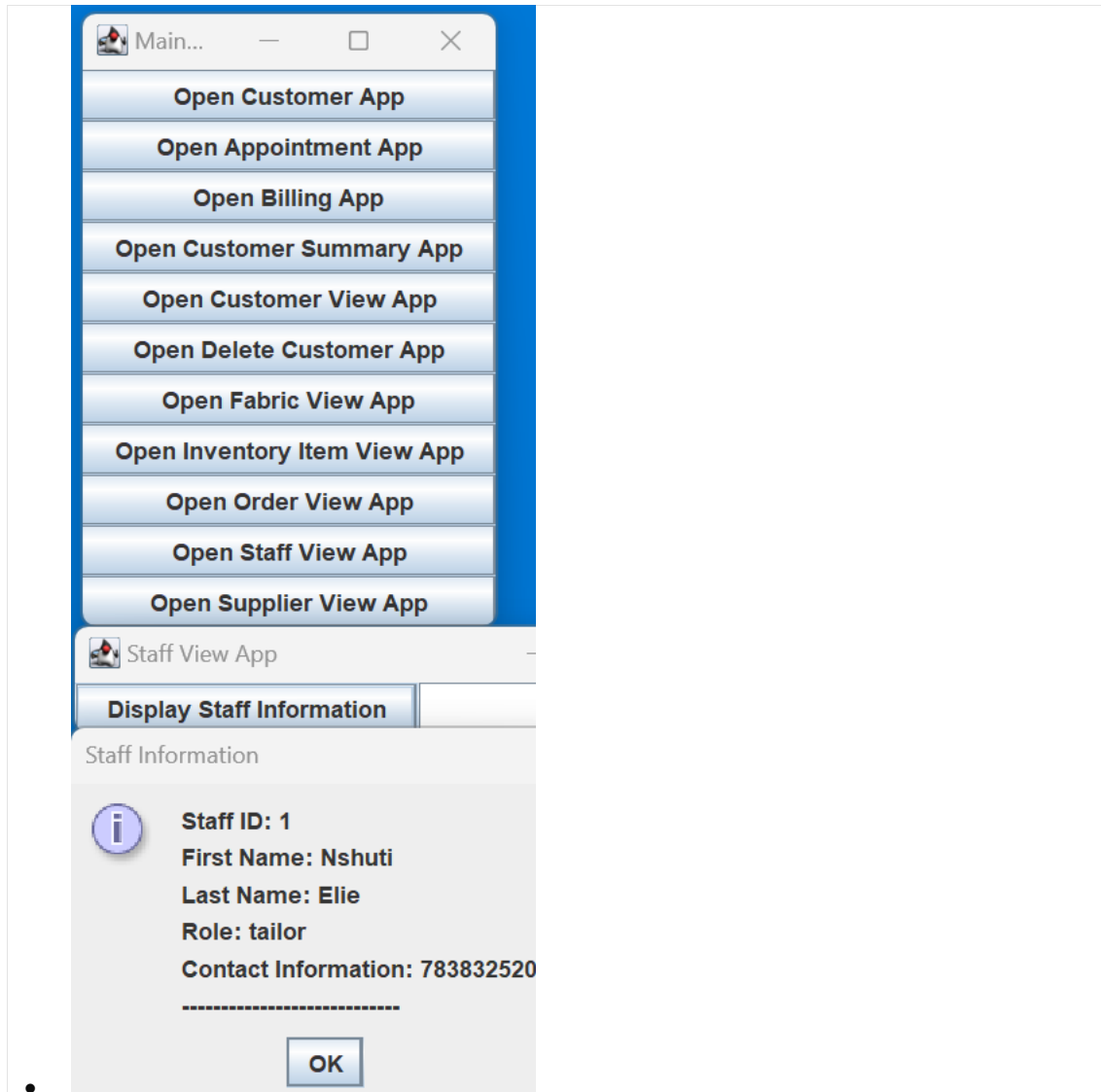
#### 7. Order Management:

- The "Open Order View App" button may lead to a module for tracking and managing customer orders.
- This could include order status, delivery dates, and other order-related information.



8. **Staff Management:**

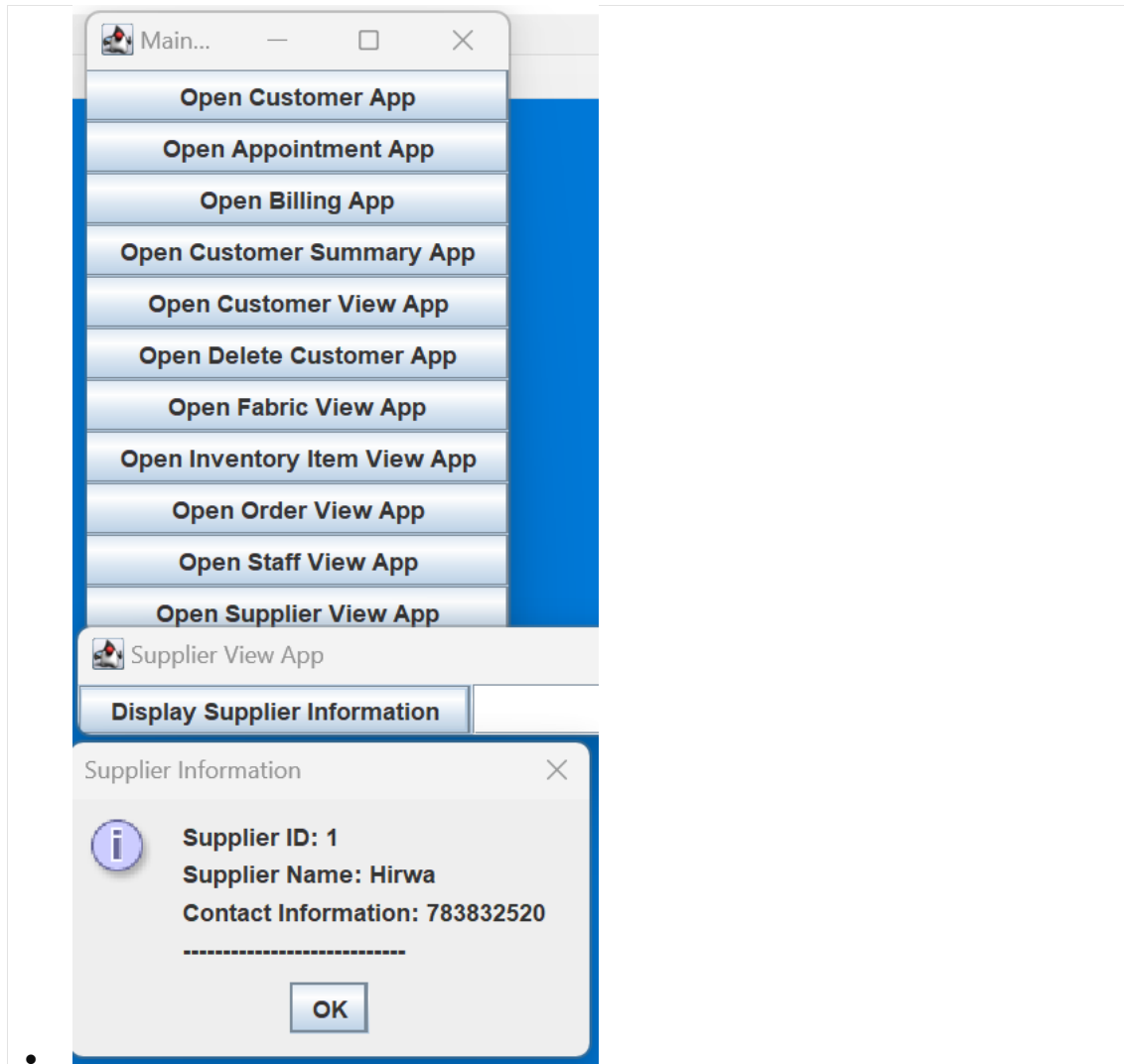
- The "Open Staff View App" button may open a module for managing staff information.



- This could include details about tailors, their work schedules, and other relevant personnel information.

#### 9. **Supplier Management:**

- The "Open Supplier View App" button might open a module for managing relationships with fabric suppliers or other vendors.
- Features may include supplier contact information, order history, and inventory updates.



#### 10. User-Friendly Interface:

- The entire application is built using Java's Swing library, providing a user-friendly Graphical User Interface for easy navigation and interaction.

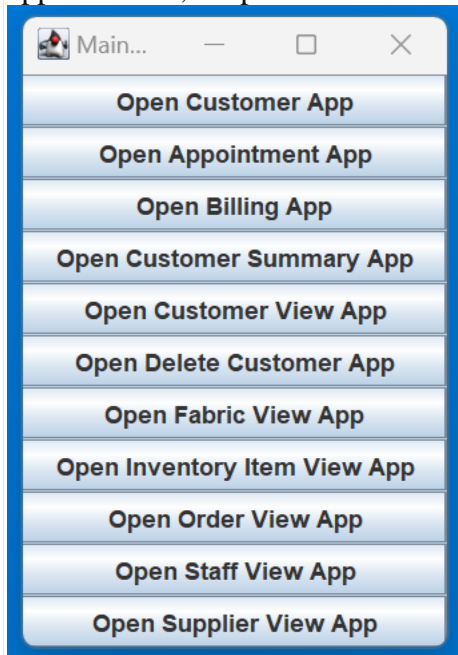
#### 11. Modular Design:

- The system follows a modular design, allowing each functionality to be encapsulated in separate modules or classes.
- This modular approach enhances maintainability and scalability.

, a Tailor Shop Management System like this could significantly improve the efficiency of day-to-day operations in a tailor shop. It helps manage customer information, streamline



appointments, keep track of inventory, and handle various other aspects of



the business in an organized and systematic manner.

The Graphical User Interface provides an intuitive interface for users to interact with the system simply.