



## CCS6344 T2410 Assignment 1 Submission

**Group 11**

|                                   |            |
|-----------------------------------|------------|
| NUR INQSYIRA BINTI ZAMRI          | 1211103098 |
| DANIEL DANISH BIN SUHAIMI         | 1201102370 |
| MOHAMED ARIQUE BIN MOHAMED AZIYEN | 1211306413 |

### **Task 6 (Click which want to view)**

Presentation - <https://youtu.be/jPQFAgUjRGI>

Slides -

[https://www.canva.com/design/DAGFl46mmcs/XRw\\_Yd38phq6q51m2cIG9A/view?utm\\_content=DAGFl46mmcs&utm\\_campaign=designshare&utm\\_medium=link&utm\\_source=editor](https://www.canva.com/design/DAGFl46mmcs/XRw_Yd38phq6q51m2cIG9A/view?utm_content=DAGFl46mmcs&utm_campaign=designshare&utm_medium=link&utm_source=editor)

## **Task 1 Preparation of the Proposal**

### **i. Objectives**

The AgentHub project aims to revolutionize order management for agents by facilitating seamless communication with manufacturers, streamlining the order process, and empowering agents with tools to track customers effectively. The project's objectives include simplifying agent-manufacturer interactions, ~~providing customer tracking capabilities~~, automating order processing and production management, creating an intuitive user interface, and enabling effortless financial management. By achieving these objectives, AgentHub will bridge the gap between manual order management processes and modern digital solutions, optimizing business operations and improving overall efficiency for agents.

### **ii. Proposed Design and Implementation**

The AgentHub application aims to enhance user experience, streamline order management processes, and integrate with other systems through an intuitive UI, robust data processing logic, and seamless integration. The UI will feature a dashboard, order management screens, ~~and reporting tools~~, while data processing logic will handle purchase management, and customer management. The application will integrate with ERP systems, accounting software, and manufacturing systems for real-time updates on production status and inventory levels. The development methodology will follow an iterative approach with Agile methodologies, regular feedback cycles, and incremental releases. The implementation timeline will be divided into sprints, with milestones and deliverables defined for each sprint. Overall, AgentHub aims to deliver a comprehensive and user-centric solution that empowers agents to streamline order management processes, enhance efficiency, and drive business growth.

### **iii. Proposed Hardware and Software**

#### **a. Programming Language and Database Programmes**

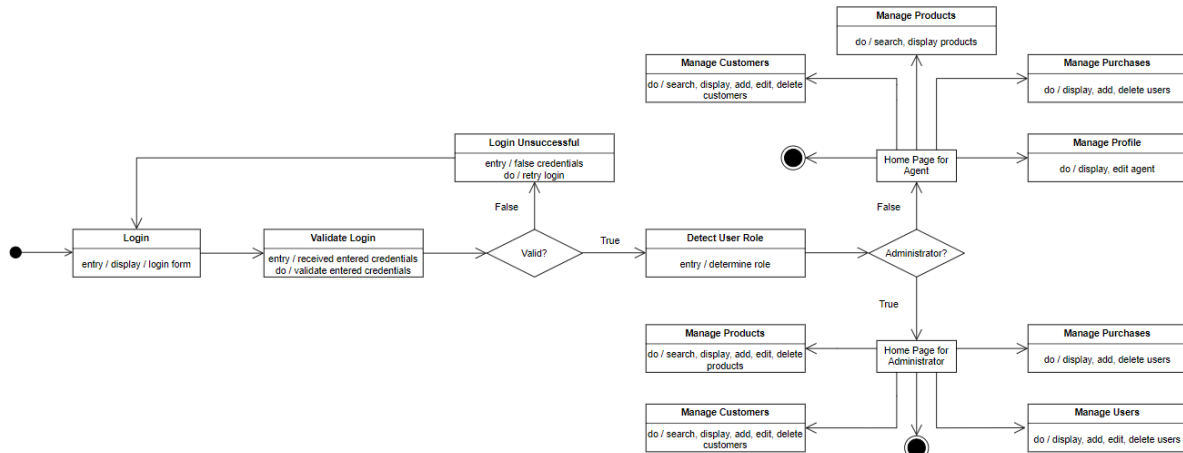
- Programming Language: Java
- Database Management System (DBMS): MySQL (Version: 8.0.25)
- JDBC connectivity: Java Database Connectivity

#### **b. Server OS and Webserver**

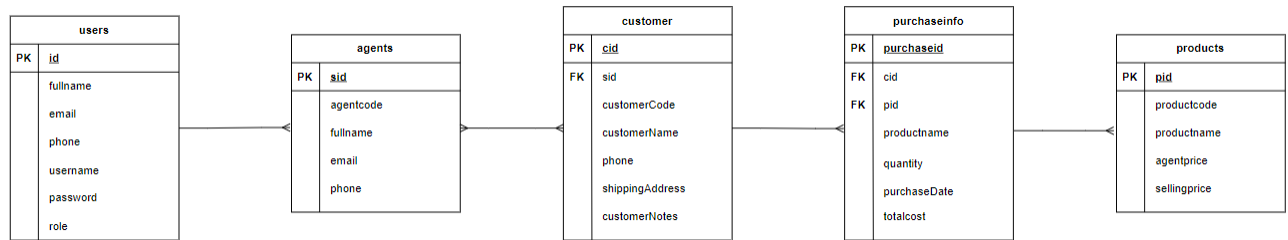
- Server Operating System: Linux (e.g., Ubuntu Server) or Windows Server
- Web Server Software: Apache Tomcat, Jetty, or WildFly

By utilizing Java with MySQL and JDBC connectivity for the backend, and Swing and additional libraries for the user interface, AgentHub will benefit from a robust and efficient architecture.

### **iv. System Design and Database Design**



## 1.1 System Design



## 1.2 Database Design

### v. Securing the Database using Traditional Database System

To secure the "AgentHub" database, we will employ a multi-layered approach with various security measures to protect data confidentiality, integrity, and availability. Here's our plan:

#### Authentication:

- Users will authenticate with unique credentials (username, password).
- Strong password policies will be enforced.

#### Authorization:

- Role-Based Access Control (RBAC) will be implemented.
- Granular permissions will be applied to database objects.

#### Encryption:

- Using hash algorithms called md5 to encrypt the password.

## Access Control Mechanisms:

- Access Control Lists (ACLs) will be implemented.
- Database auditing features will be enabled for security monitoring and compliance.

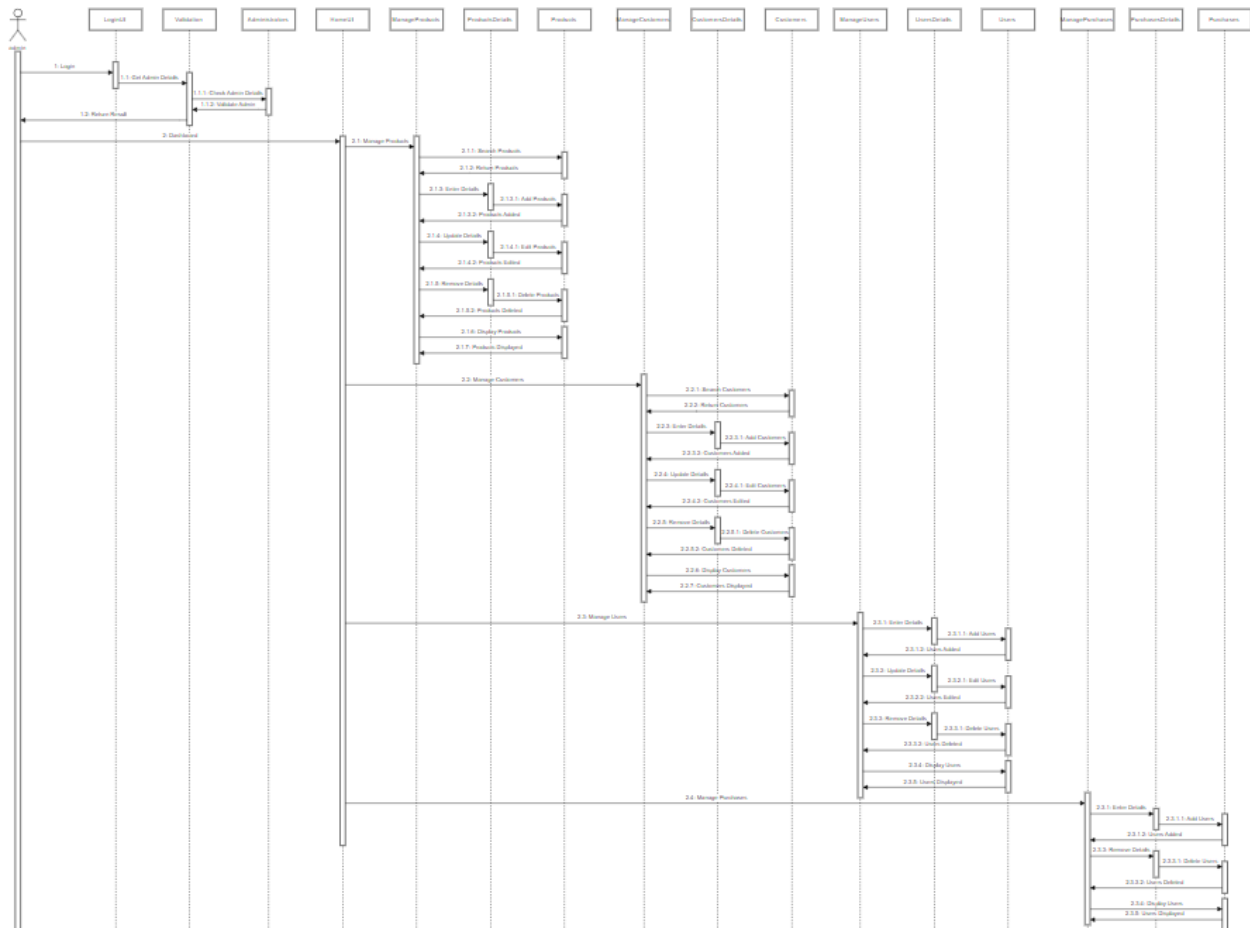
## Backup and Disaster Recovery:

- Regular backups will be implemented for data integrity and availability.
- A comprehensive disaster recovery plan will be developed.

These measures aim to protect the AgentHub database from unauthorized access, data breaches, and data loss. Regular monitoring and updates will be conducted to adapt to evolving security threats and compliance requirements.

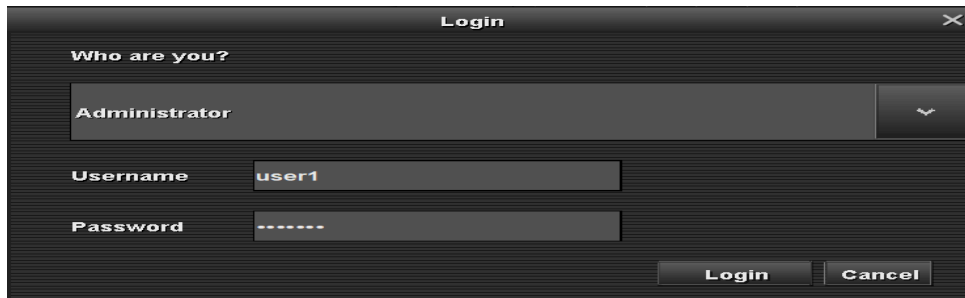
## Task 2 Implementation of the application using SQL Database

### i. Design



### 2.1 Administrators Design

### ii. Application Creation = screen snapshot



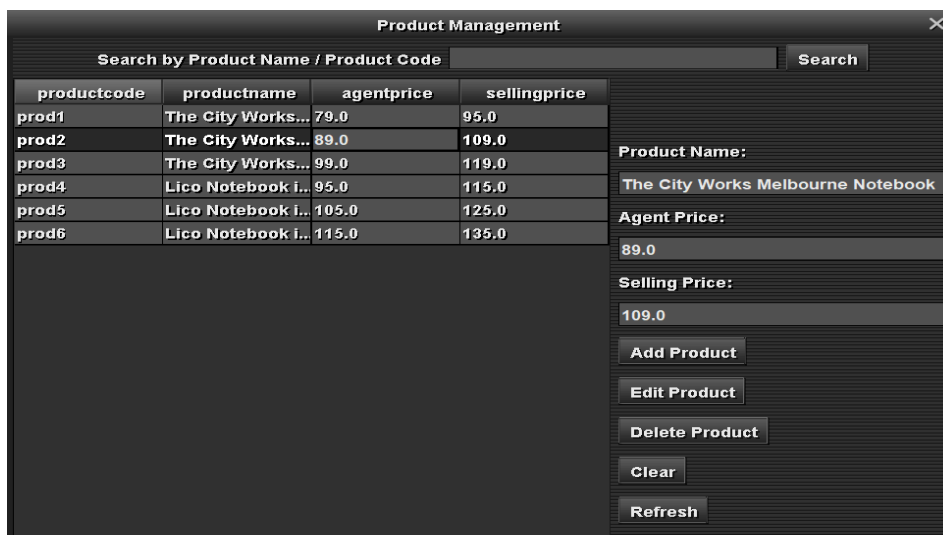
A login window titled "Login" with a close button (X) in the top right corner. The window contains a "Who are you?" section with a dropdown menu currently showing "Administrator". Below this are fields for "Username" (containing "user1") and "Password" (containing "\*\*\*\*\*"). At the bottom right are "Login" and "Cancel" buttons.

### 2.2.1 Login Form Screen



A dashboard window titled "Agent Hub - A Simple Order Management System for Agent". It features a sidebar with "Backup/View Audit Logs", "Backup", and "View Audit Logs". The main content area displays a welcome message: "Welcome, user1 to AgentHub!" followed by "We're excited to have you explore our Order Management System for Agent :)". Below this is a section "Please select your business:" with a numbered list: "1. Manage Profile", "2. Manage Products", "3. Manage Purchases", "4. Manage Customers", and "5. Manage Users". At the bottom left are "Back" and "Logout" links.

### 2.2.2 Dashboard Screen



A "Product Management" window with a close button (X) in the top right. It includes a search bar labeled "Search by Product Name / Product Code" and a "Search" button. Below the search bar is a table with the following data:

| productcode | productname        | agentprice | sellingprice |
|-------------|--------------------|------------|--------------|
| prod1       | The City Works...  | 79.0       | 95.0         |
| prod2       | The City Works...  | 89.0       | 109.0        |
| prod3       | The City Works...  | 99.0       | 119.0        |
| prod4       | Lico Notebook i... | 95.0       | 115.0        |
| prod5       | Lico Notebook i... | 105.0      | 125.0        |
| prod6       | Lico Notebook i... | 115.0      | 135.0        |

To the right of the table are input fields for "Product Name:" (containing "The City Works Melbourne Notebook"), "Agent Price:" (containing "89.0"), and "Selling Price:" (containing "109.0"). Below these are buttons for "Add Product", "Edit Product", "Delete Product", "Clear", and "Refresh".

### 2.2.3 Products Management Screen

Purchase Management

Search by Customer Name / Product Code

Search

| purchaseid | productname             | customerName | quantity | purchaseDate        | totalcost |
|------------|-------------------------|--------------|----------|---------------------|-----------|
| 19         | The City Works Tokyo... | Mr. Nava     | 2        | 2024-05-20 12:24:34 | 190.0     |
| 20         | The City Works Melbo... | Mr. Nava     | 1        | 2024-05-21 03:24:34 | 109.0     |
| 21         | The City Works Malay... | Mr. Nava     | 3        | 2024-05-22 02:24:34 | 357.0     |

Purchase Date:

Customer's Name:

Select Customer's Name

Product Code:

Agent Price:

Quantity:

Add Purchase

Delete Purchase

Clear

Refresh

2.2.4 Purchases Management Screen

Customer Management

Search by Customer Name / Customer Code

Search

| cid | customerName | phone        | shippingAddress     |
|-----|--------------|--------------|---------------------|
| 1   | Mr. Nava     | 011123456789 | Cyberjaya, Selan... |

Customer Name:

Phone:

Shipping Address:

Customer Notes:

Add Customer

Edit Customer

Delete Customer

Clear

Refresh

2.2.4 Customers Management Screen

**User Management**

| fullname       | email          | phone      | username | role         |
|----------------|----------------|------------|----------|--------------|
| Nur Ingsyir... | cira@gmail.... | 0194732486 | user1    | ADMINISTR... |
| Daniel Suha... | dan@gmail....  | 0123456789 | user2    | AGENT        |
| Mohamed A...   | arique@gm...   | 9876543210 | user3    | AGENT        |

Select User Role:  
ADMINISTRATOR ▾

User Fullname:

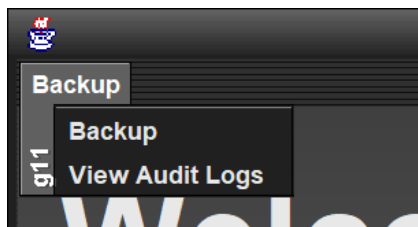
User Email:

User Phone:

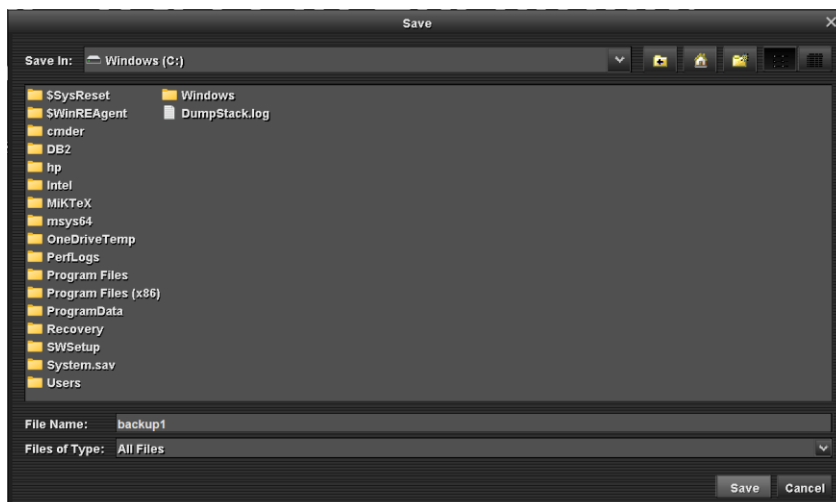
Add User  
Edit User  
Delete User  
Clear  
Refresh

## 2.2.5 Users Management Screen

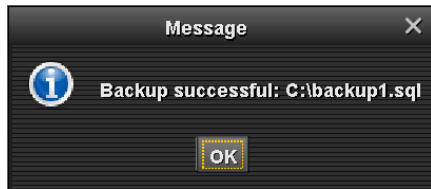
### iii. Security Measures Implemented



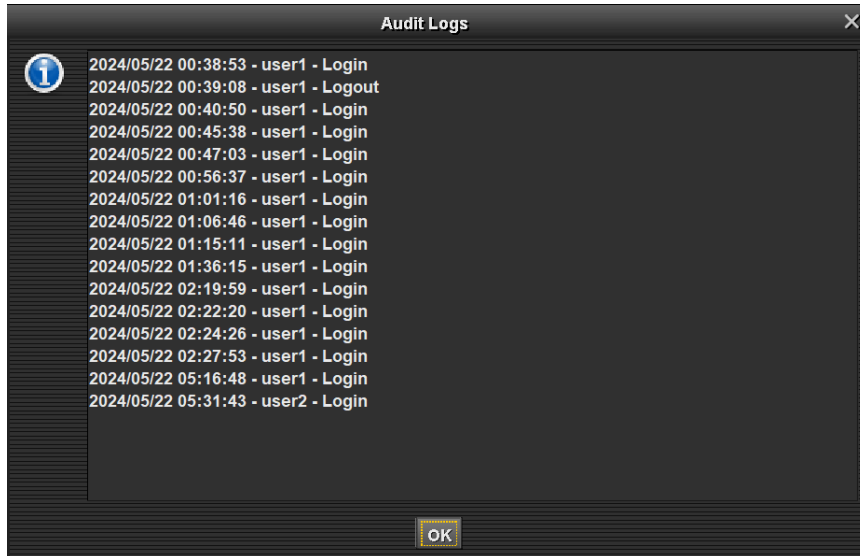
## 2.3.1 Backup / View Audit Logs Menu Bar



## 2.3.2 Backup – Select Local File Destination



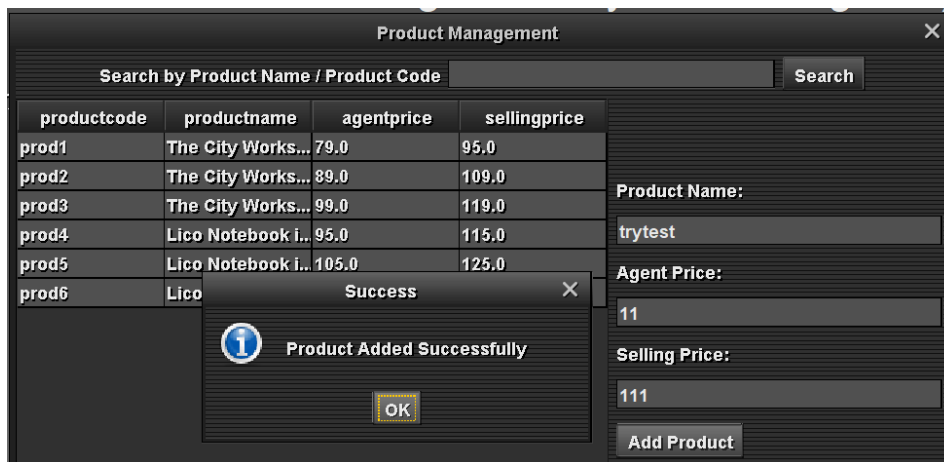
### 2.3.3 Backup – Saved Backup File



### 2.3.4 Backup – Select Local File Destination

#### iv. Application Testing

##### a. Insert New Entry / Delete Old Entries



### 2.4.1 Insert New Product



Product Management

Search by Product Name / Product Code

Search

| productcode | productname        | agentprice | sellingprice |
|-------------|--------------------|------------|--------------|
| prod1       | The City Works...  | 79.0       | 95.0         |
| prod2       | The City Works...  | 89.0       | 109.0        |
| prod3       | The City Works...  | 99.0       | 119.0        |
| prod4       | Lico Notebook i... | 95.0       | 115.0        |
| prod5       | Lico Notebook i... | 105.0      | 125.0        |
| prod6       | Lico Notebook i... | 115.0      | 135.0        |
| prod7       | trytest            | 11.0       | 111.0        |
| prod8       | trytest2           | 111.0      | 1111.0       |

Product Name:

trytest

Agent Price:

11.0

Selling Price:

111.0

Add Product

Edit Product

Delete Product

Success

Product Deleted Successfully

OK

## 2.4.2 Delete Existing Product

Product Management

Search by Product Name / Product Code

Search

| productcode | productname        | agentprice | sellingprice |
|-------------|--------------------|------------|--------------|
| prod1       | The City Works...  | 79.0       | 95.0         |
| prod2       | The City Works...  | 89.0       | 109.0        |
| prod3       | The City Works...  | 99.0       | 119.0        |
| prod4       | Lico Notebook i... | 95.0       | 115.0        |
| prod5       | Lico Notebook i... | 105.0      | 125.0        |
| prod6       | Lico Notebook i... | 115.0      | 135.0        |
| prod8       | trytest2           | 111.0      | 1111.0       |

Product Name:

Agent Price:

Selling Price:

Add Product

Edit Product

Delete Product

Clear

Refresh

## 2.4.3 Updated Product table

### b. Insert Another New Entry

Product Management

Search by Product Name / Product Code

Search

| productcode | productname        | agentprice | sellingprice |
|-------------|--------------------|------------|--------------|
| prod1       | The City Works...  | 79.0       | 95.0         |
| prod2       | The City Works...  | 89.0       | 109.0        |
| prod3       | The City Works...  | 99.0       | 119.0        |
| prod4       | Lico Notebook i... | 95.0       | 115.0        |
| prod5       | Lico Notebook i... | 105.0      | 125.0        |
| prod6       | Lico Notebook i... | 115.0      | 135.0        |
| prod7       | trytest            | 11.0       | 111.0        |

Product Name:

trytest2

Agent Price:

111

Selling Price:

1111

Add Product

Edit Product

Delete Product

Success

Product Added Successfully

OK

### 2.5.1 Insert Other New Product Entry.

The screenshot shows a 'Product Management' window with a search bar and a table of products. The search bar has a placeholder 'Search by Product Name / Product Code' and a 'Search' button. The table has columns: productcode, productname, agentprice, and sellingprice. The table contains 8 rows of product data. To the right of the table, there are input fields for 'Product Name:', 'Agent Price:', and 'Selling Price:' with values 'trytest2', '111', and an empty field respectively.

| productcode | productname        | agentprice | sellingprice |
|-------------|--------------------|------------|--------------|
| prod1       | The City Works...  | 79.0       | 95.0         |
| prod2       | The City Works...  | 89.0       | 109.0        |
| prod3       | The City Works...  | 99.0       | 119.0        |
| prod4       | Lico Notebook i... | 95.0       | 115.0        |
| prod5       | Lico Notebook i... | 105.0      | 125.0        |
| prod6       | Lico Notebook i... | 115.0      | 135.0        |
| prod7       | trytest            | 11.0       | 111.0        |
| prod8       | trytest2           | 111.0      | 1111.0       |

### 2.5.2 Updated Product Management

#### **Task 3 Threat Modeling**

Below is an explanation of how our application passes this test using the STRIDE a DREAD threat modelling methodology:

##### **i. STRIDE Threat Modelling**

| Possible Risk          | Category |
|------------------------|----------|
| Spoofing               | S        |
| Tampering              | T        |
| Repudiation            | R        |
| Information Disclosure | I        |
| Denial of Service      | D        |
| Elevation of Privilege | E        |

3.1 STRIDE Model

##### **ii. DREAD Threat Modelling**

| Risk                   | Category | D | R | E | A | D | Threat Rating |
|------------------------|----------|---|---|---|---|---|---------------|
| Spoofing               | S        | 8 | 7 | 6 | 9 | 5 | 7             |
| Tampering              | T        | 9 | 7 | 6 | 8 | 5 | 7             |
| Repudiation            | R        | 8 | 7 | 6 | 8 | 5 | 7             |
| Information Disclosure | I        | 9 | 8 | 7 | 9 | 6 | 8             |
| Denial of Service      | D        | 9 | 8 | 7 | 9 | 6 | 8             |
| Elevation of Privilege | E        | 9 | 8 | 7 | 9 | 6 | 8             |

3.2 DREAD Model

| Risk                   | Threat Rating | Countermeasures   |
|------------------------|---------------|---|
| Spoofing               | 7             | Implement strong authentication (including MFA with one-time codes), use digital signatures, validate inputs, encrypt data, use secure protocols, enforce RBAC, implement logging/monitoring, conduct regular security audits, and educate users.                     |
| Tampering              | 7             | Implement integrity checks (using hashes and digital signatures), enforce strict access controls, validate all inputs, use encryption, maintain comprehensive logging and monitoring, conduct regular security audits, and apply secure coding practices.             |
| Repudiation            | 7             | Implement strong authentication, use digital signatures to verify actions, maintain detailed logs, protect logs from tampering, regularly review logs, conduct regular security audits, and enforce access controls.  |
| Information Disclosure | 8             | Encrypt sensitive data, implement strict access controls, use secure communication protocols (like HTTPS), validate inputs and outputs, maintain detailed logs, monitor for suspicious activity, conduct regular security audits, and follow secure coding practices. |
| Denial of Service      | 8             | Implement rate limiting, use load balancers, set resource quotas, employ DDoS protection services, design for scalability, and set appropriate timeouts and retries.  |
| Elevation of Privilege | 8             | Implement strong access controls, validate inputs, regularly audit security and code, apply software patches promptly, monitor and log activities, and follow the principle of least privilege.   |

### 3.3 Countermeasures

## **Task 4 PDPA 2010**

### **i. Personnel Categorization**

| <b>Category</b> | <b>Explanation</b>   | <b>Personnel Responsible</b> |
|-----------------|--|------------------------------|
| Data User       | Have access to personal data stored in the system and are responsible for ensuring its security and integrity. | Administrators, Agents       |
| Data Subject    | The personal data is collected, processed and stored in AgentHub system. Personal data pertains.               | Customers, Agents            |

#### 4.1 Personnel Categorization

### **ii. Data Lifecycle, PDPA 2010 Requirements, Personnel and Achieved Compliance, Penalties**

| <b>Data Lifecycle</b> | <b>Explanation</b>  |
|-----------------------|---|
| Data Collection       | <ul style="list-style-type: none"><li>- <b>Notice and Choice Principle (Section 7.1)</b></li><li>- <b>Requirements:</b> Allows companies to notify consumers about the reason for data collection and receive consent to collect data</li><li>- <b>Achieving Compliance:</b> Inform customers the purposes and obtain consent</li><li>- <b>Responsible Personnel:</b> Agents</li><li>- <b>Penalties:</b> Fine up to RM 300,000 or imprisonment up to 2 years, or both (Section 5.2)</li></ul> |
| Data Storage          | <ul style="list-style-type: none"><li>- <b>Security Principle (Section 9.1)</b></li><li>- <b>Requirements:</b> Reasonable precaution to guard against unauthorized access to, damage, or disposal of personal data</li><li>- <b>Achieving Compliance:</b> Store data securely using encryption and access control</li><li>- <b>Responsible Personnel:</b> Administrators</li><li>- <b>Penalties:</b> Fine up to RM 300,000 or imprisonment up to 2 years, or both (Section 5.2)</li></ul>     |
| Data Sharing          | <ul style="list-style-type: none"><li>- <b>Disclosure Principle (Section 8)</b></li><li>- <b>Requirements:</b> Must get individuals agreement before revealing their personal information</li><li>- <b>Achieving Compliance:</b> Share data that have been consented by the data subject</li><li>- <b>Responsible Personnel:</b> Agents</li></ul>   |

|               |   |
|---------------|---|
|               | <ul style="list-style-type: none"> <li>- <b>Penalties:</b> Fine up to RM 300,000 or imprisonment up to 2 years, or both (Section 5.2)</li> </ul>  |
| Data Disposal | <ul style="list-style-type: none"> <li>- <b>Retention Principle (Section 10.2)</b></li> <li>- <b>Requirements:</b> Ensure secure disposal of personal data when it's no longer needed</li> <li>- <b>Achieving Compliance:</b> Dispose data securely to prevent unauthorized access</li> <li>- <b>Responsible Personnel:</b> Administrators</li> <li>- <b>Penalties:</b> Fine up to RM 300,000 or imprisonment up to 2 years, or both (Section 5.2)</li> </ul> |

#### 4.2 Data Lifecycle Categorization

### **Task 5 Security Measures Implementation**

#### **Security Measures**

The following security measures are implemented in the Database system design to protect the database from internal and external threats

##### **i. Internal**

Row Level Security:

- Access Control - implementation of RLS is used to impose a fine-grained access control system. This system ensures that users may only access rows in a table that they have been authorized to read, depending on their roles and responsibilities. [08]
- Dynamic Data Filtering- automatically filters the data retrieved by queries, based on the user's role. This ensures that users only have access to authorized data.

Backup and Recovery:

- Implementing regular database backups of the database to protect against data loss caused by human mistakes, system faults, or malicious actions by internal users.
- Securing backups by storing them in a secured manner and encrypting them to prevent unauthorized access.

Logon Triggers:

- **Connection Monitoring:** Logon triggers are employed to oversee and regulate database connections, carrying out predetermined actions such as recording user activity or imposing access restrictions based on specified circumstances.
- **Unauthorized Access Prevention:** Logon triggers can be utilized to deter unauthorized access by implementing supplementary verifications or recording any questionable login attempts.