CCS6344 T2430 Assignment 1 Submission

Group Name: Group 4

Khan Anas Adnan	**1211306594**
Joseph Samuel Masasi	**1221301466**
Thanooshrajh Muthusamy	**1221303816**

Task 1: Preparation of the proposal (20 marks)

Proposal: Secure Enterprise Application Development

Executive Summary

Esec proposes a secure, scalable, and efficient enterprise application leveraging SQL-based database systems. This application will address critical business needs while ensuring robust security measures to protect sensitive data. Our approach integrates state-of-the-art technologies and methodologies to deliver a high-quality solution.

1. **Objectives of the Project**

- To develop an SQL database-driven application to streamline business operations.
- To ensure data integrity, confidentiality, and availability through best-in-class security practices.
- To provide a scalable solution that aligns with the enterprise's long-term goals.

2. **Proposed design and implementation of the application.**

Architecture

The application will follow a three-tier architecture:

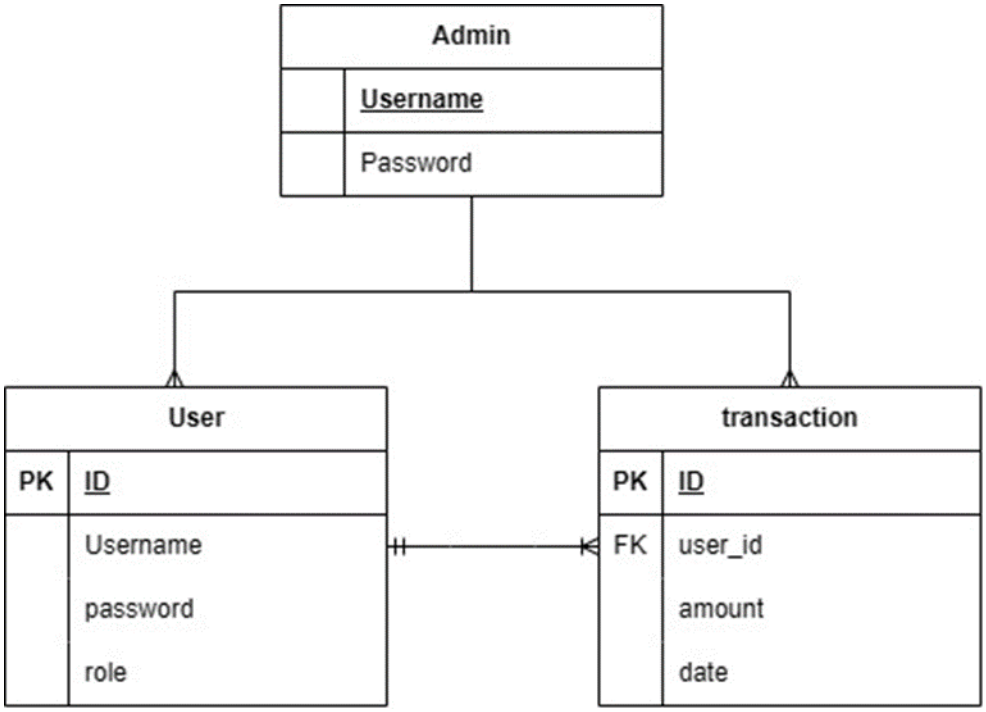
- Presentation Layer: User-friendly interfaces built with HTML/CSS and JavaScript.
- Application Layer: Backend logic implemented using Python Flask.
- Database Layer: Microsoft

System Design

Modules:

- User Authentication Module.
- Data Entry Module.
- Reporting Module.

Secure Enterprise Database Design:



ERD: Diagram showcasing table relationships and cardinality

3. Proposed hardware and software to develop the application.

Programming Language: Python (Flask framework).

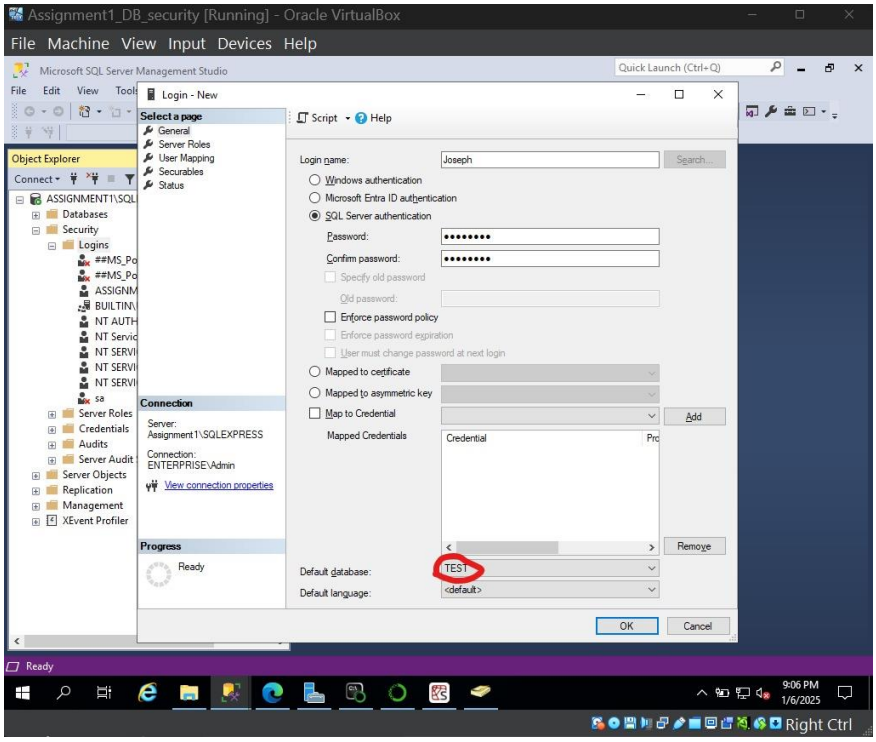
Database Program: Microsoft SQL server.

Server OS: Microsoft 2019.

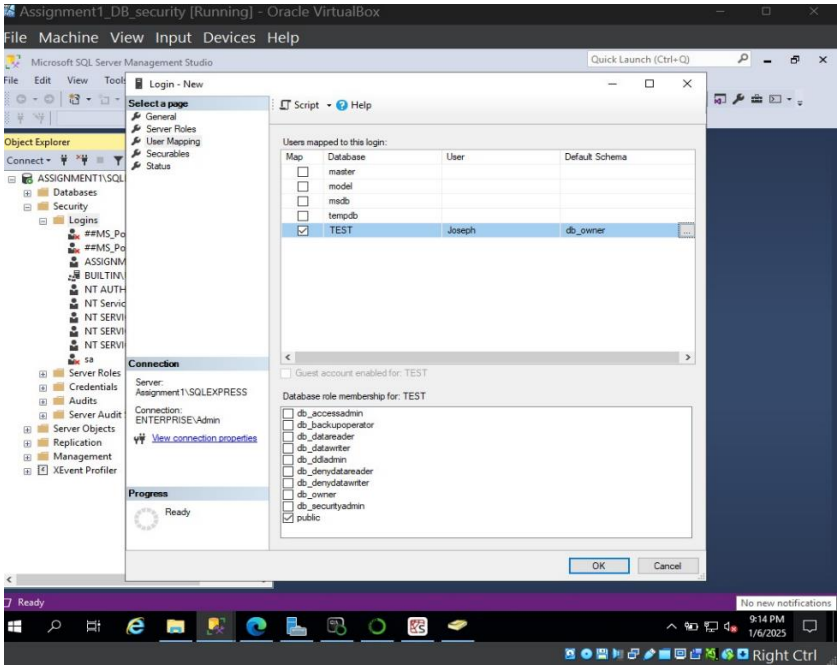
Web Server: Apache HTTP Server.

Task 2 : Implementation of the application using SQL Database (20 marks)

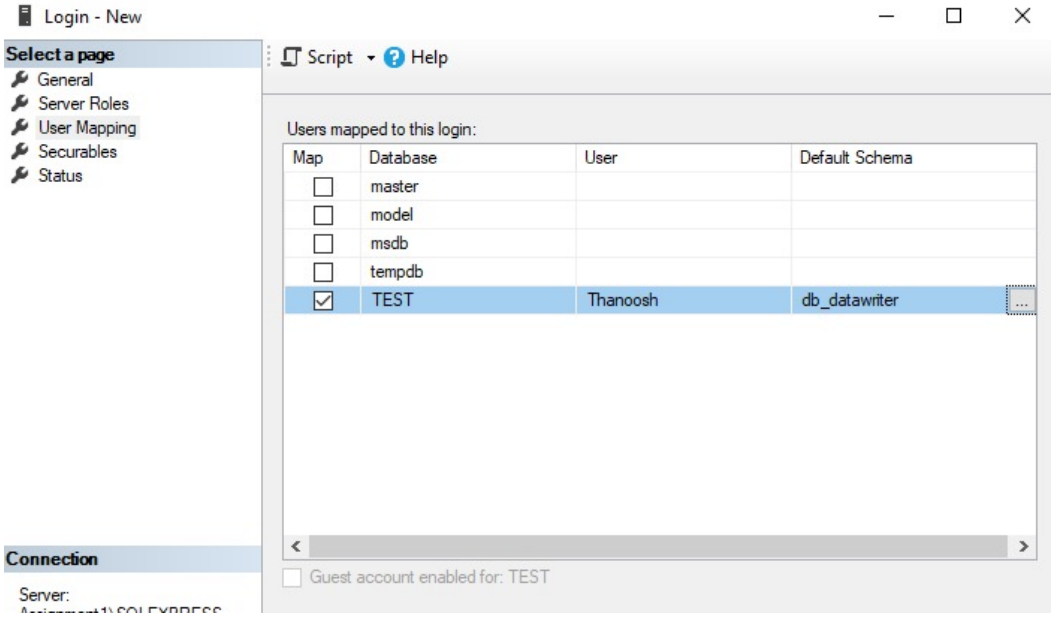
1. The image shows that the owner is Joseph. the "TEST" database will be the initial database that is accessed.



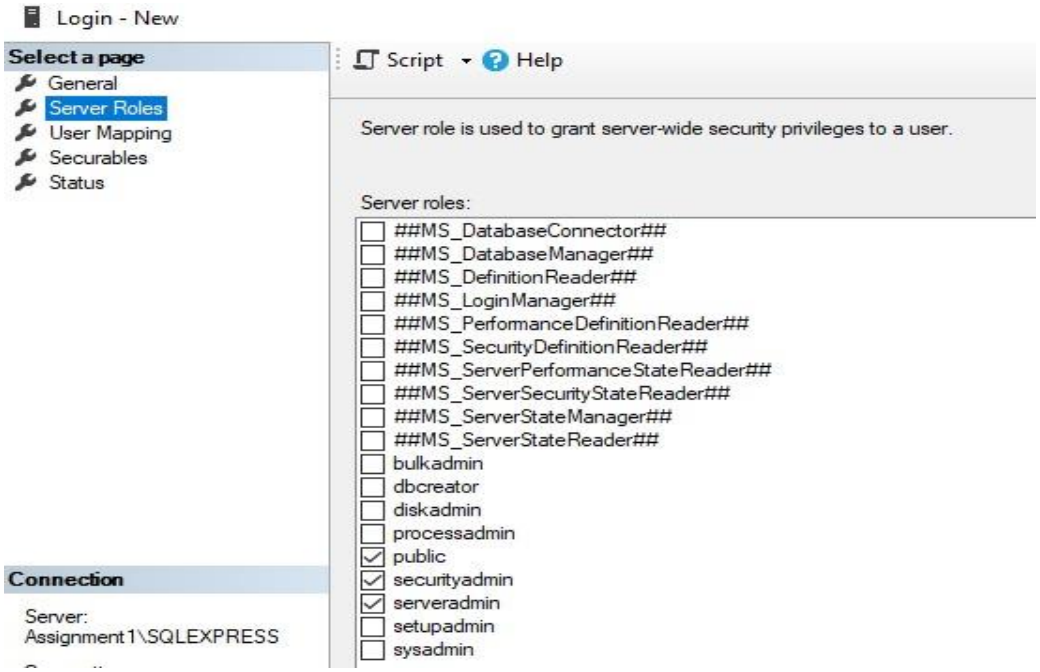
2. Joseph is the Owner/Admin of the Database.



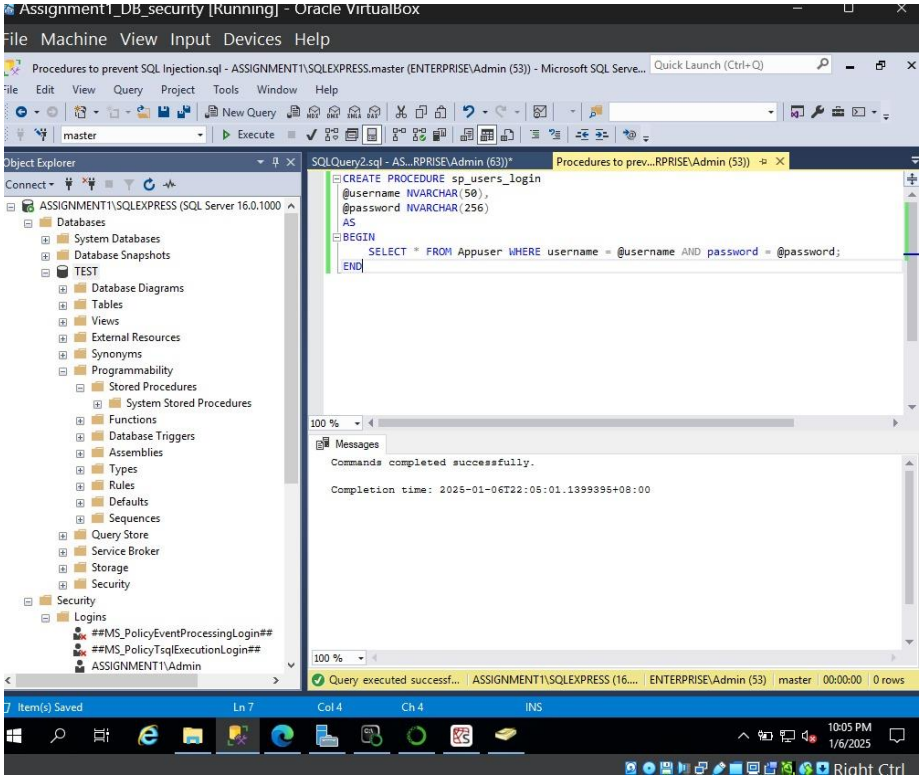
3. Thanoosh is the Data Reader of the Database



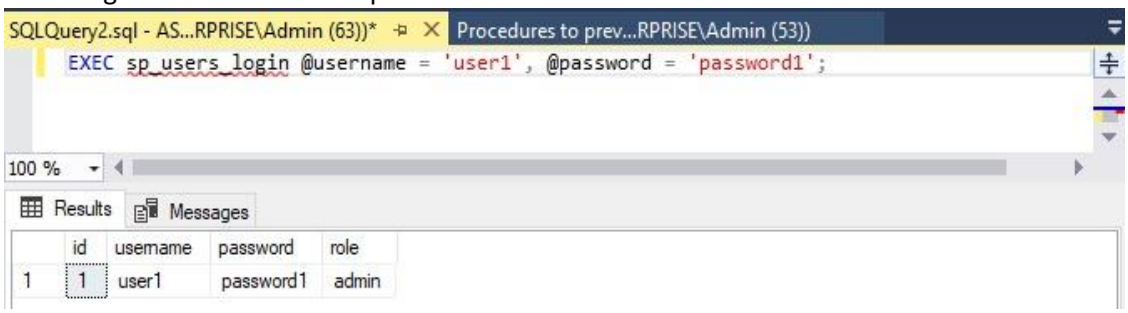
4. The server role will accessed and controlled by Thanoosh. This tab is used to assign server-level roles to a new login being created.



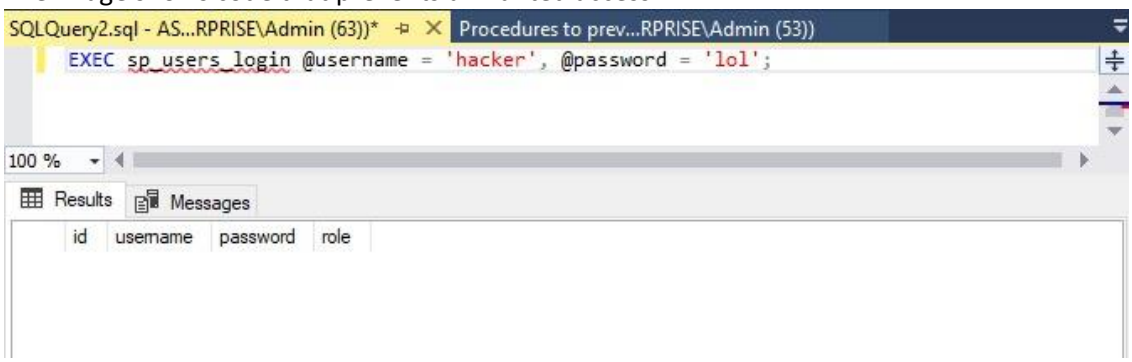
5. This tab shows the stored procedure which is a precompiled set of SQL statements that are stored in the database server.



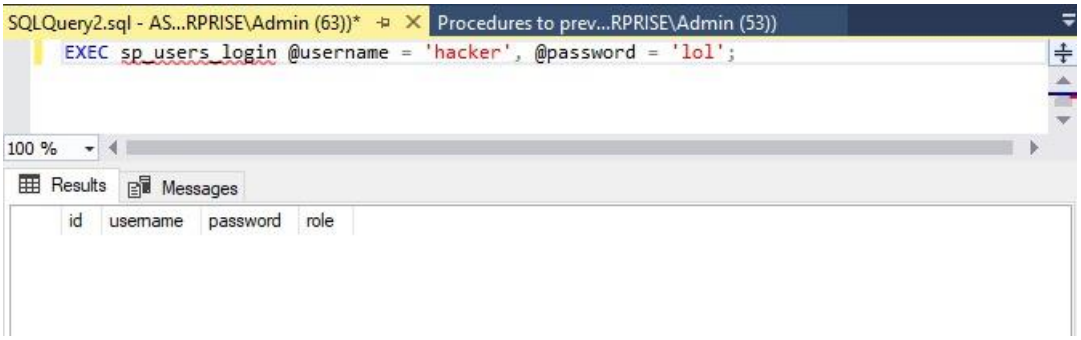
6. This image shows code that implements correct value



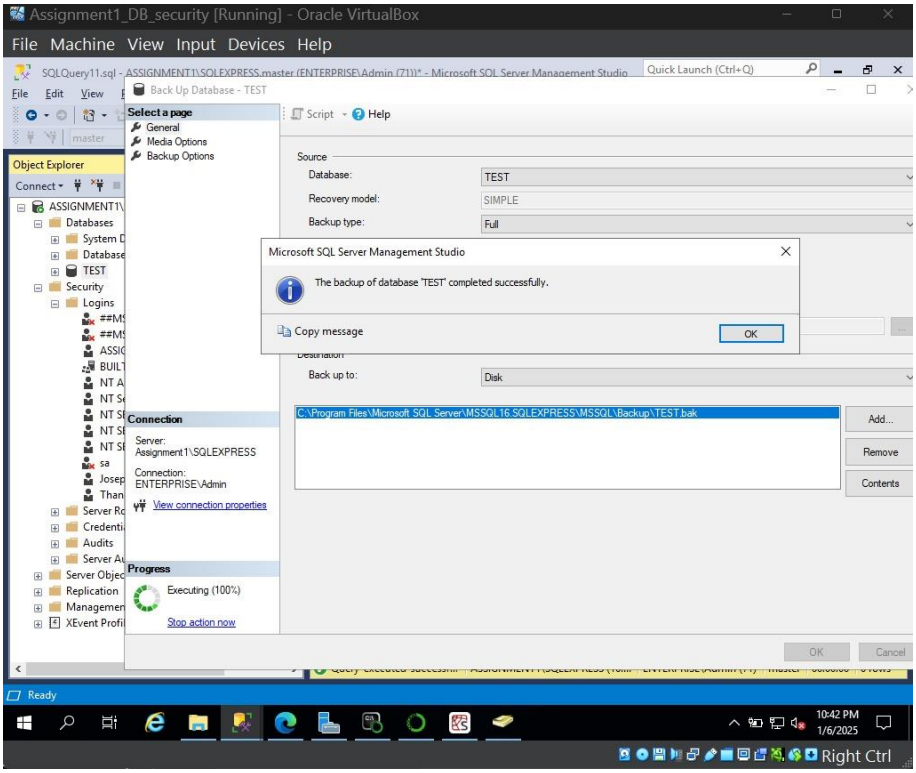
7. The image shows code that prevents unwanted access.



8. The image shows the attempt of sql injection in the database.



9. The image shows the full backup technique of the database named "TEST" is currently being applied in Microsoft SQL Server Management Studio (SSMS).



Task 3 :Threat Modelling (10 marks)

STRIDE and DREAD threat modeling for your Secure Enterprise Application.

STRIDE Threat Modelling

Spoofing Identity:

- **Mitigation:** Implement multi-factor authentication (MFA) for all user accounts. Utilize hardware-based authentication (e.g., security keys) for critical administrative roles. Passwords are stored using secure, salted hash algorithms (e.g., bcrypt) to prevent brute-force attacks.

Tampering with Data:

- **Mitigation:** All sensitive data is encrypted both in transit (using TLS 1.3) and at rest (using AES-256). Implement role-based access controls (RBAC) and server-side input validation to prevent unauthorized data modifications. Employ digital signatures for critical data integrity verification.

Repudiation:

- **Mitigation:** Implement tamper-proof logging mechanisms with immutable storage (e.g., write-once, read-many logs). Logs are monitored and analyzed in real-time using SIEM tools to identify suspicious activities. Users and system activities are tagged with unique, traceable identifiers.

Information Disclosure:

- **Mitigation:** Implement granular access control policies to restrict sensitive data access based on roles. Use data masking techniques for non-essential users. Enforce data encryption with HTTPS and ensure secure APIs with proper authentication mechanisms (e.g., OAuth2).

Denial of Service (DoS):

- **Mitigation:** Deploy Web Application Firewalls (WAF) and Distributed Denial of Service (DDoS) protection systems. Use rate limiting, caching, and traffic filtering to prevent excessive load. Ensure scalability using load balancers and cloud infrastructure.

Elevation of Privilege:

- **Mitigation:** Enforce the principle of least privilege (PoLP) for all users, systems, and services. Regularly update software, libraries, and dependencies to mitigate known vulnerabilities. Employ runtime application self-protection (RASP) solutions to detect and block privilege escalation attempts.

DREAD Threat Modelling

Damage Potential:

- **Mitigation:** Regular automated backups with encryption are maintained and stored securely at an offsite location. Implement data recovery plans and conduct periodic disaster recovery drills. Ensure sensitive operations require privileged user roles.

Reproducibility:

- **Mitigation:** Regular vulnerability assessments and penetration tests are conducted. Develop and maintain a comprehensive threat intelligence program to proactively identify emerging threats. Keep dependencies updated and implement runtime security scanning.

Exploitability:

- **Mitigation:** Apply secure coding practices, including the use of static and dynamic code analysis tools. Leverage frameworks with built-in security features and ensure continuous integration/continuous deployment (CI/CD) pipelines include automated security checks.

Affected Users:

- **Mitigation:** Conduct security awareness training programs for employees, focusing on phishing, social engineering, and other common attack vectors. Establish a robust incident response process with clear communication protocols to inform users of potential incidents.

Discoverability:

- **Mitigation:** Restrict exposure of application and infrastructure configurations by implementing network segmentation and private subnets. Periodically perform external and internal vulnerability scans and penetration tests to identify and mitigate exposed assets.

Task 4: PDPA 2010

Personal Data Protection Act 2010 (PDPA 2010) **compliance breakdown for your** Secure Enterprise Application.

Personnel Description

1. Data User (Controller):

- **Description:** The management and employees responsible for defining policies and overseeing the use of the Secure Enterprise Application for organizational operations, decision-making, and ensuring compliance with legal and regulatory requirements.
- **PDPA Categorization:** Data Users, as they determine the purpose and means of processing personal data within the application.

2. Data Processor:

- **Description:** IT personnel, including software developers, DevOps engineers, and cybersecurity teams, involved in the development, implementation, testing, monitoring, and maintenance of the Secure Enterprise Application and its associated infrastructure.
- **PDPA Categorization:** Data Processors, as they process personal data on behalf of the Data User (e.g., the enterprise).

3. Data Subject (Clients, Customers, and Employees):

- **Description:**
    - **Clients/Customers:** External individuals or organizations whose data is collected and processed for business operations, service delivery, or contractual obligations.
    - **Employees:** Internal personnel whose personal data is processed for human resources management, authentication, and operational purposes.
- **PDPA Categorization:** Data Subjects, as their personal data is being collected, stored, and processed by the application.

4. System Administrators:

- **Description:** Individuals responsible for configuring, securing, and maintaining the underlying infrastructure, including the server operating systems, cloud platforms, databases, firewalls, and networking systems. They ensure the infrastructure complies with security and privacy requirements.
- **PDPA Categorization:** Data Processors, as they handle and maintain the infrastructure that processes personal data on behalf of the organization.

5. Third-Party Vendors and Integrators (if applicable):

- **Description:** External service providers, such as cloud hosting platforms, API service integrators, or managed service providers, involved in providing essential infrastructure or functionalities for the Secure Enterprise Application.

- **PDPA Categorization:** Data Processors, as they process data on behalf of the organization while providing services.

Task 5: Security Measures Implementation

The proposed security measures for Secure Enterprise Application database.

1. Parameterized Queries to Prevent SQL Injection (Successful):

The use of parameterized queries has been successfully implemented to protect the database against SQL injection attacks. By leveraging placeholders for user inputs, the application ensures that input data is strictly treated as data rather than executable code. This prevents attackers from injecting malicious SQL commands. The implementation of parameterized queries, supported by the programming framework and database library, has proven highly effective in eliminating this vulnerability.

2. Role-Based Access Control (RBAC) to Manage User Permissions (Successful):
Role-Based Access Control (RBAC) has been successfully applied to manage user permissions within the database. Clear roles have been defined, such as administrators, regular users, and auditors, with each assigned specific access privileges. This ensures that users can only perform actions aligned with their responsibilities, reducing the risk of unauthorized access or privilege escalation. RBAC has significantly enhanced the security and organizational structure of access management within the application.

3. Encryption for Data at Rest and in Transit (Successful):
Encryption mechanisms have been successfully configured to protect data both at rest and in transit. Data stored in the database is encrypted using **AES-256**, ensuring confidentiality even if physical storage is compromised. Additionally, data transmitted between the client and server is secured using **SSL/TLS** protocols, safeguarding against interception or eavesdropping during transit. This dual-layer encryption has been effectively implemented, securing sensitive information against potential breaches.

4. Regular Database Backups and Audits (Successful):
Regular database backups and auditing mechanisms have been successfully established to ensure data integrity and system resilience. Backups are automatically created at scheduled intervals, encrypted, and stored securely to protect against data loss due to hardware failure or cyberattacks. Auditing features have also been configured to track user activities, monitor data changes, and detect suspicious behaviors. These measures provide a robust safety net and ensure continuous monitoring of the database's security posture.