

**CCS6344 T2510 Assignment 1 Submission**

**Group Name: Group 12**

**Project** **Title : Bakery Ordering System**

**Lecturer Name: Dr. Navaneethan A/L C.Arjuman**

**Prepared By:**

|  |  |
| --- | --- |
| **Student ID** | **Name** |
| 1211112284 | Nur Dania Iman Binti Desman Desa |
| 1211101800 | Tan Jia Jin |
| 1211101775 | Lam Yuet Xin |

YouTube link:

GitHub link:

Contents

[**Task 1 - Preparation of the proposal** 2](#_Toc198559810)

[**Task 2 - Implementation of the application using SQL Database** 4](#_Toc198559811)

[**Task 3 Threat Modelling (STRIDE AND DREAD)** 8](#_Toc198559812)

[**Task 4 PDPA 2010** 8](#_Toc198559813)

[**Task 5 Security Measures Implementation** 10](#_Toc198559814)

# **Task 1 - Preparation of the proposal**

**1.1 Objective of the project**

The aim of this project is to design, develop, and secure a web-based application using the conventional SQL database system in line with comprehensive security best practices and processes for regulatory compliance with PDPA 2010 regulations. The project will present project documentation, including a technical report and presentation.

* Build a web application with proper authentication mechanism
* Imposing security controls on general threats like SQL injection attacks
* An analysis of threats using the STRIDE and DREAD framework
* Working on data handling processes conforming to the PDPA 2010 regulations
* Validation of security through penetration tests
* Preparation of all professional documentation and presentation

**1.2 Design and Implementation of the Application**

The Bakery Ordering System is a web-based application designed to allow customers to conveniently place food orders online. The system emphasizes both security and user experience, ensuring that interactions on both the customer and administrator sides are safe, smooth, and user-friendly.

The key features of the application include:

* User Management: Customers can create accounts and log in securely. Input validation is implemented to prevent security vulnerabilities and protect against automated login attempts, ensuring secure and reliable access.
* Menu and Products: Customers can browse a clearly displayed menu that lists all available food items along with their prices before placing an order.
* Staff Management: Staff members can also create accounts and log in securely. Similar input validation measures are in place to prevent security vulnerabilities and unauthorized access.
* Order Management System: Users can place and manage their orders efficiently through the system.
* Secure Payment System: Payments are handled securely using encryption to protect payment information. Input validation is also implemented to prevent fraudulent or incorrect data entries, maintaining the integrity and security of every transaction.

**1.3 Hardware and software to develop the application**

1. Programming Language: PHP for backend, HTML/CSS/JavaScript for frontend.
2. Database: MSSQL for data storage. SSMS for database management.
3. Operating System: Windows 11/Window 10
4. Server: XAMPP Local Server

**1.4 System design and database design for this application**

System Design

The system is designed using a multi-layered approach:

1. **Application Layer:** This is the backend of the system, responsible for handling user requests, executing business logic, and managing the communication between the user interface (presentation layer) and the database (data layer).
2. **Presentation Layer:** This is the part of the system that users interact with directly. It includes web pages, forms, and other elements of the user interface that allow users to browse, input, and view information.
3. **Data Layer:** This layer uses an MSSQL database to store all necessary data for the application. It includes user accounts, product details, staff records, order histories, and payment information.

Database Design

The database includes the following tables:

1. Customers: To store the customer’s personal details.
2. Products: To store product details.
3. Staff: To store the staff’s personal details.
4. Orders: To store order receipt.
5. Order Items: To store order details placed by customers.
6. Payment: To manage payment details.

**1.5 Secured database using the SQL database system**

To secure the database, the following measures are implemented:

a. **Authentication**:  
Before accessing the database, users must prove their identity by entering a valid username and password. These credentials are checked against those stored in the system to confirm the user is legitimate.

b. **Authorization**:  
Once authenticated, users are only allowed to access the data they are permitted to see or change. This is based on their specific roles and access levels within the system.

c. **Data Encryption**:  
Sensitive data stored in the database is encrypted, meaning it’s converted into a secure format that can’t be easily read by unauthorized users.

d. **Password Hashing**:  
Passwords are converted into a secure hashed format before being stored in the database. This ensures that even if the database is compromised, the original passwords remain protected.

e. **Data Masking**:  
Sensitive details, such as credit card numbers, are partially hidden or masked in the database so that the full information is not visible to unauthorized users.

f. **Input Validation**:  
All data entered by users is checked to make sure it’s in the correct format and doesn’t contain harmful code or malicious input, helping to prevent security threats like injection attacks.

g. **Audit Log**:

It records who accessed the sensitive data, when, and for what purpose, creating an audit trail for accountability and transparency. Log help track and record security-related events, such as database access, login attempts, and DDL operations.

# **Task 2 - Implementation of the application using SQL Database**

**2.1 System Design**

The Bakery Ordering System is an online platform that allows users to conveniently place food orders. The system is designed to support three main roles: **Admin**, **Staff**, **User**,and **Guest**.

* **User Role**: Users can log in to the website and view the menu, which displays available food items and their prices. After selecting their desired items, users proceed to the order form to place their order. Once the order is submitted, payment is made manually at the counter, where staff will update the system with the user's credit card number for processing. When all steps are completed successfully, the user's order is confirmed.
* **User Role (Guest)**: Guest can access in to the website and view the menu which displays only available food items without pricing. If guests want to make an order or discover pricing, they must create an account by directing to sign up page from “Join Us!” button (guest order page) or go to sign up.
* **Admin Role**: Admins can log in to the system to access and manage various aspects of the platform. They can view all orders, monitor payment history, manage products, and access information about both customers and staff members.
* **Staff Role**: Staff members have access similar to the admin, with the ability to view orders, manage products, and see customer information. However, they do not have access to view or manage staff details.
* **HR Role**: HR users can access and manage all staff‑related functions including creating, updating, and deactivating staff accounts, reviewing personnel records. They cannot view or modify customer orders, payment details, or other operational data.
* **System Admin**: The system admin is responsible for installation, configuration, and maintenance of the web and application servers.
* **Database Owner**: The database owner manages the SQL Server instance itself—defining the schema, creating and altering tables, configuring backups, and setting up server‑level security policies. They can grant or revoke database permissions, manage encryption keys, and oversee routine maintenance tasks.

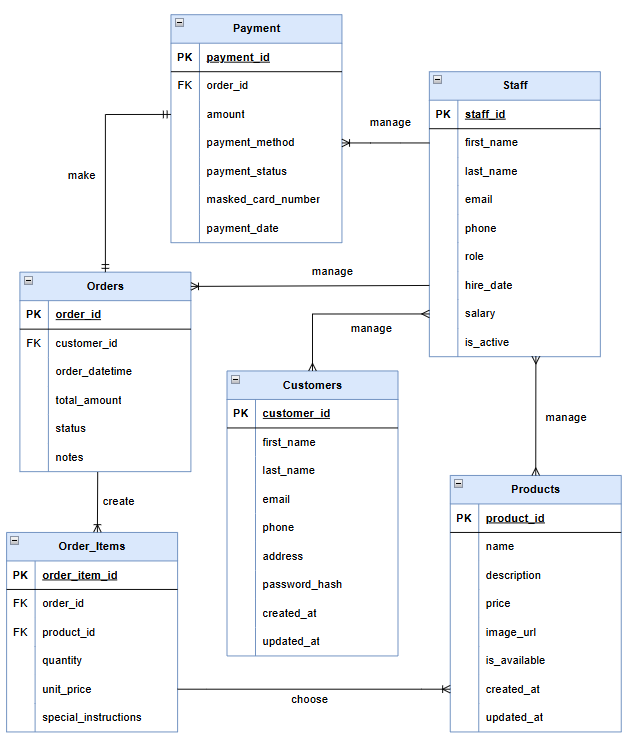


Figure 2.1.0 Bakery Order System Entity Relational Diagram

**2.2 Whole Workflow Application Screenshot**

|  |  |
| --- | --- |
| **Guest(Crumby Café Web Page)** | **Customer(Crumby Café Web Page)** |
| A screenshot of a login form  AI-generated content may be incorrect.  Figure 2.2.1 User can continue as guest | A sign up form with white text  AI-generated content may be incorrect.  Figure 2.2.2 Sign up page |
| A close up of food  AI-generated content may be incorrect.  Figure 2.2.3 Guest’s home page | A screenshot of a login form  AI-generated content may be incorrect.  Figure 2.2.4 Log in page |
| A collage of food on a screen  AI-generated content may be incorrect.  Figure 2.2.5 Guest can view menu without price | A close up of food  AI-generated content may be incorrect.  Figure 2.2.6 User access to Crumby Café home page |
| A screenshot of a computer screen  AI-generated content may be incorrect.  Figure 2.2.7 Sign up to order | A collage of food on a screen  AI-generated content may be incorrect.  Figure 2.2.8 Users can view menu |
| A sign up form with white text  AI-generated content may be incorrect.  Figure 2.2.9 Sign up page | A computer screen shot of a computer screen  AI-generated content may be incorrect.  Figure 2.2.10 Users can create an order. |
| -- | A screenshot of a computer  AI-generated content may be incorrect.  Figure 2.2.11 Users need to make payment at the counter. |
| -- | A screenshot of a computer  AI-generated content may be incorrect.  Figure 2.2.12 Users can view order history. |

|  |  |
| --- | --- |
| **Staff(Admin WebPage)** | **Admin(Admin WebPage)** |
| A screenshot of a computer  AI-generated content may be incorrect.  Figure 2.2.13 Staff can see customer details. | A screenshot of a computer  AI-generated content may be incorrect.  Figure 2.2.14 Admin can see customer details |
| A screenshot of a computer  AI-generated content may be incorrect.  Figure 2.2.15 Staff can see past ordered created by customer. | A screenshot of a computer  AI-generated content may be incorrect.  Figure 2.2.16 Admin can see past ordered created by customer |
| A screenshot of a computer  AI-generated content may be incorrect.  Figure 2.2.17 Staff can see history payment made by customer. | A screenshot of a computer  AI-generated content may be incorrect.  Figure 2.2.18 Admin can see history payment made by customer |
| A screenshot of a computer  AI-generated content may be incorrect.  Figure 2.2.19 Order Items or foods by customer. | A screenshot of a computer  AI-generated content may be incorrect.  Figure 2.2.20 Order Items or foods by customer |
| Staff is not authorized to access Staff page | A screenshot of a computer  AI-generated content may be incorrect.  Figure 2.2.21 Admin can see staff details. |
| A screenshot of a computer  AI-generated content may be incorrect.  Figure 2.2.22 Staff can manage the stock of the product. | A screenshot of a computer  AI-generated content may be incorrect.  Figure 2.2.23 Admin can manage the stock of the product. |

**2.3 Security Measures Implementation**

**Authentication:** Authentication makes sure that only valid users gain access to the system. Customers log in with email and password to place orders. Staff/admin log in with their name and email to process orders and handle product management, whereas only admins have the authorize to access staff page with admin ID and passcode to have full control over the system.

**Authorization:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Authorization levels for both staff and admin can access the dashboard and managing products; the privilege to view and manage staff information is designated solely for admins. Staff do not have permission to view staff details. | |  |  |  | | --- | --- | --- | |  | Staff | Admin | | Dashboard | Checkmark with solid fill | Checkmark with solid fill | | Staff Information | Close with solid fill | Checkmark with solid fill | | Product Management | Checkmark with solid fill | Checkmark with solid fill |   Table 2.3: Authorization |
| Role-based Access Control (RBAC) determines what actions a user can perform after they have been authenticated. We create the sql roles and user to enforce least privilege by structuring permissions around roles. Figure beside shows the granted permission for each roles | Figure 2.3.4 SQL Permission Setup |

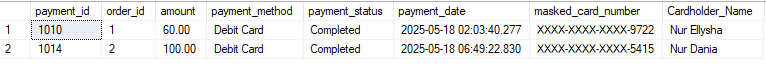
**Hashing:** Used to secure stored data like user passwords by converting them into a non-reversible hash. Enhances security as the original password cannot be retrieved. Any data change results in a completely different hash, helping detect tampering and maintain data integrity.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 2.3.4 Masking Card Number.

**Masking:** The web application employs masking as a security technique to protect sensitive information, as shown in Figure 2.3.4. When displaying credit card numbers, only the final four digits (9722) remain visible, while the first twelve digits are replaced with 'X' characters (XXXX XXXX XXXX 9722). This partial concealment significantly reduces the risk of unauthorized access and potential misuse of card information, thereby preserving the confidentiality of this critical security element.

Figure 2.3.4 Masking Card Number.

**Encryption:** Figure 2.3.3 demonstrates encryption implementation for protecting sensitive information such as `cardholder\_name` is encrypted under the `AES-256-CBC` algorithm with a given key, so that the data is accessible to authorized users with the appropriate decryption key only. The method protects against unauthorized access to the data, supports PDPA compliance, and makes data interception an exercise in futility.

Figure 2.3.5 Encryption of Cardholder Name.

**Decryption -** Only authorized parties who possess the correct decryption key can accessed the original name.



Figure 2.3.6 Decryption of Cardholder Name.

**Input Validation:**

|  |  |  |
| --- | --- | --- |
| A form input validation message is triggered if the First Name field is left empty.  A close up of a box  AI-generated content may be incorrect.  Figure 2.3.5 First Name Validation | email input validation error prompting the user to include an '@' symbol, if is not a valid email format.  A close-up of a sign  AI-generated content may be incorrect.  Figure 2.3.6 Email Validation | password input validation error indicating that the entered password must be at least 8 characters long.  A white rectangular box with black text  AI-generated content may be incorrect.  Figure 2.3.7 Password Validation. |

**Audit Log:**

The SQL Server audit configuration establishes comprehensive monitoring for the Bakery database system by implementing three server audits (Bakery\_DB\_Audit, Bakery\_Login\_Audit, Bakery\_Data\_Audit) and two database audit specifications (Bakery\_HR\_Audit, Sensitive\_Access\_Audit). The server audit specifications track security-related events: *Audit\_User\_Management* records server role, login, and permission changes; *Audit\_Privilege\_Changes* monitors database and object-level modifications; and *Audit\_Failed\_Logins* captures failed authentication attempts. At the database level, *Bakery\_HR\_Audit* logs all staff table modifications by HR personnel, while *Sensitive\_Access\_Audit* tracks admin access to payment and customer data. The figure below shows the example of HR audit to track the recruiting(insert) and firing(delete) of staff.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 2.3.8 HR Audit Log

# **Task 3 Threat Modelling (STRIDE AND DREAD)**

|  |  |
| --- | --- |
| Spoofing Identity | To prevent spoofing:   * Strong user authentication with secure login * Input validation to block fake credentials |
| Tampering with data | To prevent tampering with data:   * Role-based access control (RBAC) ensures only authorized users can modify specific data |
| Repudiation | To prevent repudiation:   * Restrict access to sensitive data using role-based permissions and database-level access controls * Encrypt sensitive data to protect it from unauthorized access or leaks, even if the system is breached |
| Information Disclosure | To prevent information disclosure:   * Data encryption for sensitive data (e.g., credit card numbers) * Data masking for confidential information. |
| Denial of Service | To prevent denial of service:   * Input validation prevents system overload |
| Elevation of Privileges | To prevent elevation of privileges:   * RBAC implementation to restrict access * Regular review and updates to permissions and roles. * No direct user access to admin roles |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Stride Category | Threat Detection | D | R | E | A | D | Threat  Rating |
| S | Customer or attacker logs in as staff/admin | 4 | 3 | 6 | 5 | 6 | 4.8 |
| T | Order manipulation (price/quantity modification) | 3 | 4 | 5 | 4 | 5 | 4.2 |
| R | Customer denies payment or staff denies action | 2 | 2 | 3 | 3 | 4 | 2.8 |
| I | Card info or order history leak | 5 | 3 | 5 | 6 | 6 | 5.0 |
| D | Too many requests prevent new orders | 7 | 6 | 6 | 7 | 6 | 6.4 |
| E | Customer gains admin/staff access | 6 | 4 | 6 | 5 | 5 | 5.2 |

# **Task 4 PDPA 2010**

Categorize of Personnel According to PDPA 2010:

|  |  |
| --- | --- |
| **Personnel in Charge** | **Responsibilities** |
| **Data Users** (Data Controllers) : Individuals who collect, store, and process personal data, and have authorize the processing of any personal data. | * Ensure compliance with data protection principles * Implement appropriate security measures * Inform data subjects about data processing activities * Respond to data subjects' requests regarding their personal data |
| **Data Subjects:** Individuals whose personal data is being collected, stored, or processed. | * Right to be informed about the processing of their data * Right to access their personal data and correct inaccurate data * Right to withdraw consent for data processing |
| **Data Processors**: Entities or individuals that process data on behalf of the data user. | * Process data only on the instructions of the data user * Implement appropriate technical and organizational measures to protect personal data * Assist the data user in fulfilling data subject requests |
| **Data Protection Officer (DPO)**: A designated individual responsible for overseeing data protection strategy and compliance. | * Ensure the organization's compliance with the PDPA * Conduct regular data protection audits * Serve as a point of contact for authorities and data subjects * Provide training on data protection |
| **Internal Audit and Compliance Team**: Team responsible for ensuring internal policies and procedures comply with the PDPA. | * Conduct regular audits to ensure data protection practices are followed * Review and update data protection policies * Provide recommendations for improving data protection measures |

**Mapping Data Lifecycle to PDPA 2010 Requirements:**

|  |  |  |
| --- | --- | --- |
| **Stage** | **PDPA Principle(s)** | **Compliance Method** |
| 1. Personal Data Collection  Stage | General Principle,  Notice & Choice Principle | 1.obtaining explicit consent through consent forms  2.providing privacy notices |
| 2.Data Storage  Stage | Security, Retention and Data Integrity  Principle | 1.storing personal data on encrypted servers  2.Role-based access control is implemented and regular security audits are conducted  3.Personal data shall not be kept longer than necessary.  4. Ensure that the data stored is accurate and updated |
| 3.Personal data  usage &  processing stage | Disclosure and  Purpose Principle | 1. No disclosure without consent  2.Disclosure to third parties is subject to Data Processing Agreements (DPA)  3. Data should only be processed for the specified purposes |
| |  | | --- | | 4.Sharing and Disclosure Stage |  |  | | --- | |  | | |  | | --- | | Disclosure  principle | | 1.Personal data shall not be disclosed without consent  2. Follow specific conditions for transferring data to third parties  3. Obtain consent before sharing data with third parties |
| |  | | --- | | 5.Data Retention  and Disposal  Stage | | |  | | --- | | Retention and  Data Integrity  Principles | | 1.data is retained only as long as it is needed and is disposed of under secure measures  2. Take reasonable steps to ensure that personal data is accurate and not misleading  3. Ensure secure disposal of data when no longer needed |
| 6. Data Accuracy  & Updates | Access Principle and Correction Principle | Data subjects must be granted access to their personal data and correction of their data upon formal request. |
| 7.Audit and Review Stage | Security Principle | 1.Role-based access control is implemented to protect personal data from loss, misuse, modification  2.regular security audits are conducted to compliance with PDPA requirements |

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.0.0 Password Validation.

# **Task 5 Security Measures Implementation**

Our team implemented security measures in the Bakery Ordering System's SQL database that comply with PDPA regulations to protect against internal and external threats. First, we established an authentication and least privilege authorization system at every access point. The users such a customer, staff, admin must login with unique credentials that are hashed and stored only in their hashed form. The database strictly follows the Role-Based Access Control (RBAC), where customers can only view menus and place orders, staff can manage orders and inventory, HR handles staff records, and administrators control bakery operations without system-level access. Administrative roles are separated, database owners and system admins have distinct accounts, credentials, preventing privilege escalation and shared superuser credentials that can be abused internally.

On the application side, every query into the database uses parameterized SQL, never concatenated string, so that user‑supplied input is never interpolated directly into DDL statements. In addition, we enforce strict input validation which only allows characters and length‑checked fields are submitted to the database. It stops command injection, cross‑site scripting, and buffer‑overflow attempts at the earliest point.

Internally, we protect sensitive columns such as credit card numbers, cardholder names, personal identifiers through a combination of column‑level encryption, data masking, and auditing. Payment fields are encrypted with AES‑256‑CBC using a securely stored key in SQL server Always Encrypted key vault, so only authorized application roles can decrypt them in memory; meanwhile, on any admin or staff screen, card numbers are masked to “XXXX XXXX XXXX 9722.” We also maintain a detailed audit trail via SQL server audit. Server‑level audits capture login failures, permission‑change events, and schema modifications, while database‑level audits log all SIUD operations on critical data.