Project Proposal[1]: GotoGro-MRM – Solution Design

Table of Contents

[1] This document is by no means a "full project proposal". It has been simplified and customized for the purposes of SWE20001 teaching. The full project proposal includes many other sections which have not been discussed during the first few weeks of SWE20001 teaching.

Solution Direction Description:

Based on the project objectives, scope, and the analyses we conducted in 01P and 02P, both individually and as a group, I find that the most suitable chosen solution direction for Goto Grocery's member management system is to develop a web-based application with a multi-layer architecture. This architecture will consist of the following layers:

- Frontend Layer (User Interface)
- Backend Layer (Application Logic and Functionality)
- Data Storage Layer

This choice is driven by the need for a user-friendly graphical user interface (GUI), the ability to handle real-time interactions, and the project's focus on modernizing and enhancing store operations. By dividing the member management system into distinct layers, the system functions more efficiently, is easier to maintain, and becomes more straightforward to scale if the need arises.

KoST Analysis:

Knowledge:

Problem Domain:

- Understanding the challenges faced by Goto Grocery in member management and the grocery industry. Such as difficulties managing their current paper-based records system which makes analysing records and extracting relevant information harder.
- Knowledge of inventory management, member needs, sales records, and reporting is required to design an effective solution.

Solution Domain:

- Implementing an electronic records management system for easy access and analysis, reducing manual efforts in managing orders and member data.
- Introducing electronic inventory control that synchronizes with records, enhancing stock level tracking and accuracy.
- Utilizing data analysis, including trend and predictive analysis, to ensure preparedness for consumer demands and streamlined stock management.

Skills:

- **UI Design:** Experience in designing easy-to-use user interfaces taking into consideration usability and accessibility standards.
- **Frontend Development:** Experience with frontend development languages (such as HTML, JavaScript, and CSS) and frameworks such as VueJs and React.
- **Backend Development:** Experience with backend development languages (such as C# and C++), design patterns and technologies.
- **Database Management:** Experience with database design, SQL and working with relational databases as the system requires storing member and sales data and extracting that data for analysis and reporting.

Technology:

- No new technologies are needed for the implementation of this design as many existing solutions use the same architecture. All the technologies mentioned above are accessible, for example, frontend frameworks and backend technologies have been used in so many projects and are highly reliable and accessible for the purpose of this project.

Alternatives Considered:

Table 1 Alternative Considered and Justification for Discarding

Alternative	Description	Rationale for Discarding
Desktop Application	Developing a desktop application instead of a web application. This application would be installed on individual computers within the store and accessed locally.	Discarded due to limitations in accessibility and platform dependence. A web application, on the other hand, offers easier access and can be used across different devices and operating systems.
Commercial Software	Purchasing and customizing existing commercial software for member management	Discarded due to the need for a tailored solution to match Goto Grocery's specific requirements as well as added customisation costs.

Rationale for Chosen Solution:

After careful consideration, the above two alternatives (see Table 1) were discarded in favour of developing a multi-layer web-based application for the following reasons:

- Customisation: Goto Grocery needed a solution that could be tailored precisely to its unique requirements, offering functionalities that met the store's specific needs.
- Accessibility and Flexibility: A web-based application provides accessibility from various devices and locations. Staff members could manage the system remotely, ensuring flexibility and convenience.
- Scalability: A web application can be easily scaled to accommodate future growth in terms of users, data volume, and functionalities.
- Cost-Effectiveness: While the initial development of a custom web application involves upfront costs, it offers a long-term cost advantage compared to purchasing licenses and customizing commercial software.
- Alignment with Modern Trends: Web applications align with contemporary technology trends, making them a more appealing choice for users accustomed to intuitive and user-friendly interfaces.

In conclusion, the chosen solution of a web-based application with a multi-layer architecture offered the best balance of customization, accessibility, scalability, and cost-effectiveness, making it the most suitable choice for Goto Grocery's member management system.

High-Level Design:

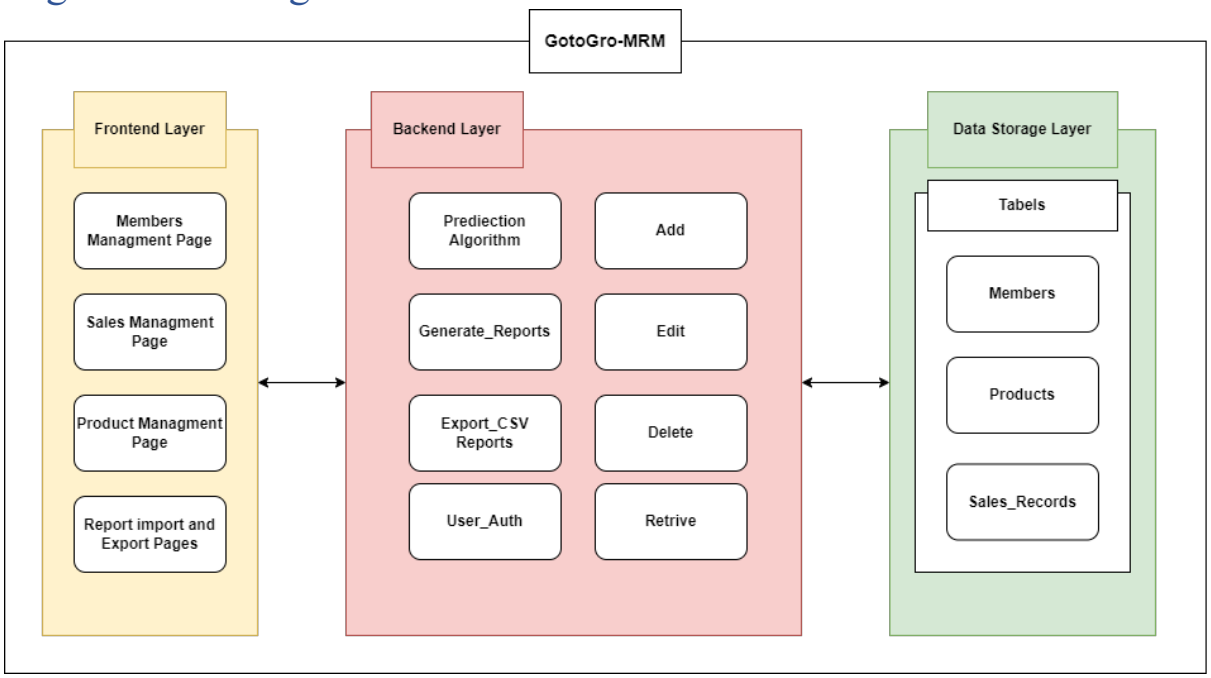

Figure 1 High-Level Design of the Chosen GotoGro MRM

Frontend Layer (User Interface):

This layer is responsible for the user interface components that members and store staff interact with. It includes web pages for adding/editing/removing members, sales records, and products as well as viewing reports and analytics.

Backend Layer (Application Logic and Functionality):

This layer handles the business logic of the system. It processes user requests from the frontend layer, interacts with the data storage layer (for data retrieval, insertion, and deletion), and performs actions such as predicting member grocery needs, generating reports, and managing sales data. It ensures that data flows correctly between the user interface and the data storage. User authentication and account management also reside in this layer.

Data Storage Layer:

This layer manages the storage of data. It includes database tables for members, sales records, and products. The layer is responsible for maintaining data integrity and ensuring accurate record-keeping.

Components, Roles, and Responsibilities:

Frontend Layer (User Interface):

- User Interface Components: Web pages for adding/editing/removing members, sales records, and products.
- User Authentication: Handles user login and authentication, ensuring secure access to the system.
- Account Management: Allows users to manage their accounts, change passwords, and update profile information.

Backend Layer (Application Logic and Functionality):

- Business Logic: Processes user requests, implements prediction algorithms, generates reports, and handles system functionalities.
- Integration: Interacts with the data storage layer to retrieve and store data as needed.
- Data Validation: Validates user inputs to ensure data accuracy and consistency.

Data Storage Layer:

- Database Tables: Stores members' details, sales records, and product information.
- Data Integrity: Enforces data integrity through proper validation and constraints.
- Query Processing: Handles database queries and transactions for efficient data retrieval and manipulation.

Conclusion:

Overall, the chosen multi-layer architecture design (see Figure 1) divides the member management system into three distinct layers that work together seamlessly. The Frontend layer receives input and user requests from the users and sends it to the Backend/Application layer where the data is analysed and processed. The Backend/Application layer then interacts with the Data Storage layer to insert, delete and/or retrieve data as required. The flow is then reversed from the Data Storage layer to the Application layer and lastly to the Frontend layer to present the outcome to the users. Ensuring that each layer has its own specific roles and responsibilities within a multi-layer architecture promotes a structured, organized, and efficient approach to software development. It leads to more a robust, maintainable, and scalable application that aligns well with modern software engineering practices while meeting the project objectives of enhancing store operations, accuracy in record-keeping, and providing an improved shopping experience for customers.