Project Proposal: GotoGro-MRM

Team Details

Team Name:	MSP 14
Tutorial:	Tue 2:30 ATC325
Tutor:	Dr Kaberi Naznin

Members:	
Dylan Jarvis	102093138
Rabya Tayal	103144215
Simon Tran	103602807
Thomas Babicka	103059885
Cody Cronin-Sporys	103610020
Nicholas Dyt	101624265

Solution Direction

In a realistic setting, the project calls for a web hosted database management system (DBMS) with desktop clients on all relevant point of sale (POS) machines. Together these constitute and enterprise framework. However, given the timeframe, we can simplify the implementation to either be entirely web hosted – accessible through a browser – or entirely desktop hosted, where the database exists directly on the host machine. Table 1 shows a comparison between the two alternatives:

Table 1. Comparison of Solution Directions

Desktop Application	**Web Server**
Database is stored locally on machine	Database is stored on the web on a server
Program is self-contained	Program must be stored a separate web server
Program is not easily accessible from mobile devices	Program can run on any device with internet access
No additional hardware required	Server required, either through cloud-hosting or physical storage
Generally cheaper	Generally more overheads
Requires coding skills only	Requires coding skills, also skill with markup languages HTML, CSS etc.

To make a distinction between the two, we must consider the problem domain, the solution domain as well as personal familiarity with relevant business knowledge, skills, and technology:

Table 2. Knowledge of the Problem Domain

No.	**Item**	**Description**
1	Customer needs	Knowledge of what a customer needs from the business, specifically, why are they using a member-based retail chain
2	Business needs	Knowledge of business needs, why they chose to use a member system, what benefits this provides
3	Current Technology	What kind of physical systems do they currently have in place, how do these function, how can they be used/repurposed
4	Network Architecture	How does the business currently manage its network, security concerns, how can this be leveraged to solve problems

Table 3. Knowledge of the Solution Domain

No.	Item	Description
5	Solution Technology	Knowledge of what technologies exist already to deliver digital reporting, how are these used in industry
6	Server Management	How to set up and manage a physical/cloud-hosted server to store company information
7	Cybersecurity	What are the threats to the system and how can these be managed
8	Desktop Applications	Competency in a coding like C#, C++, Python, Ruby, enough to deliver the desired functionality
9	UI	Competency in linking raw code to a user interface which can be used by the intended users without issue
10	Database Management	Competency in working with and extracting data from a database
11	File Management	Competency in extraction data from software or database and saving it into a file, especially csv format
12	Web Technologies	Competency using HTML, CSS, PHP and Javascript to create a web interface

Table 4. Team Competencies

No.	Item	Dylan	Rabya	Cody	Thomas	Nicholas	Simon
1	Customer Needs	Medium	Low	Medium	Medium	Medium	Low
2	Business Needs	High	Low	Medium	Medium	Medium	Low
3	Current Business Technology	High	Low	Low	Medium	High	Low
4	Network Architecture	Medium	Medium	Medium	Low	Medium	High
5	Solution Technology	Medium	Low	Medium	Medium	High	Medium
6	Server Management	Low	Low	Low	Low	High	High
7	Cybersecurity	High	Low	Medium	Low	High	Medium
8	Desktop Applications	High	Low	High	Medium	Medium	High
9	UI	High	Medium	High	High	Medium	High
10	Database Management	High	High	High	High	High	High
11	File Management	Medium	Medium	Medium	Medium	Medium	Medium
12	Web Technologies	High	High	Medium	High	Medium	Medium

Gap Analysis

Table 4 demonstrates the overall competencies of our team in the skill areas required for the project. Overall, our strongest competency is in database management, which is ideal because both solution directions require a database.

Our weakest skill appears to be in server setup and management, closely followed by customer needs. Extra time must be assigned to these aspects to ensure they are up to standard.

In terms of experience with creating desktop applications versus web applications, we have a strong split, with a slight lean towards web proficiencies. Three of the team members opted for a web-based architecture, while the other three provided justification for a modified enterprise structure. In this model, the database would be hosted on a web server and business logic would be handled on another, but instead of providing access through webpages, a desktop client would be used to interface with the logic layer.

Ultimately, we decided to go for this model. The members of the team more comfortable with desktop applications will be responsible for creating the client software, while the team members with stronger web proficiency will be responsible for the database and business server setup.

High Level Design

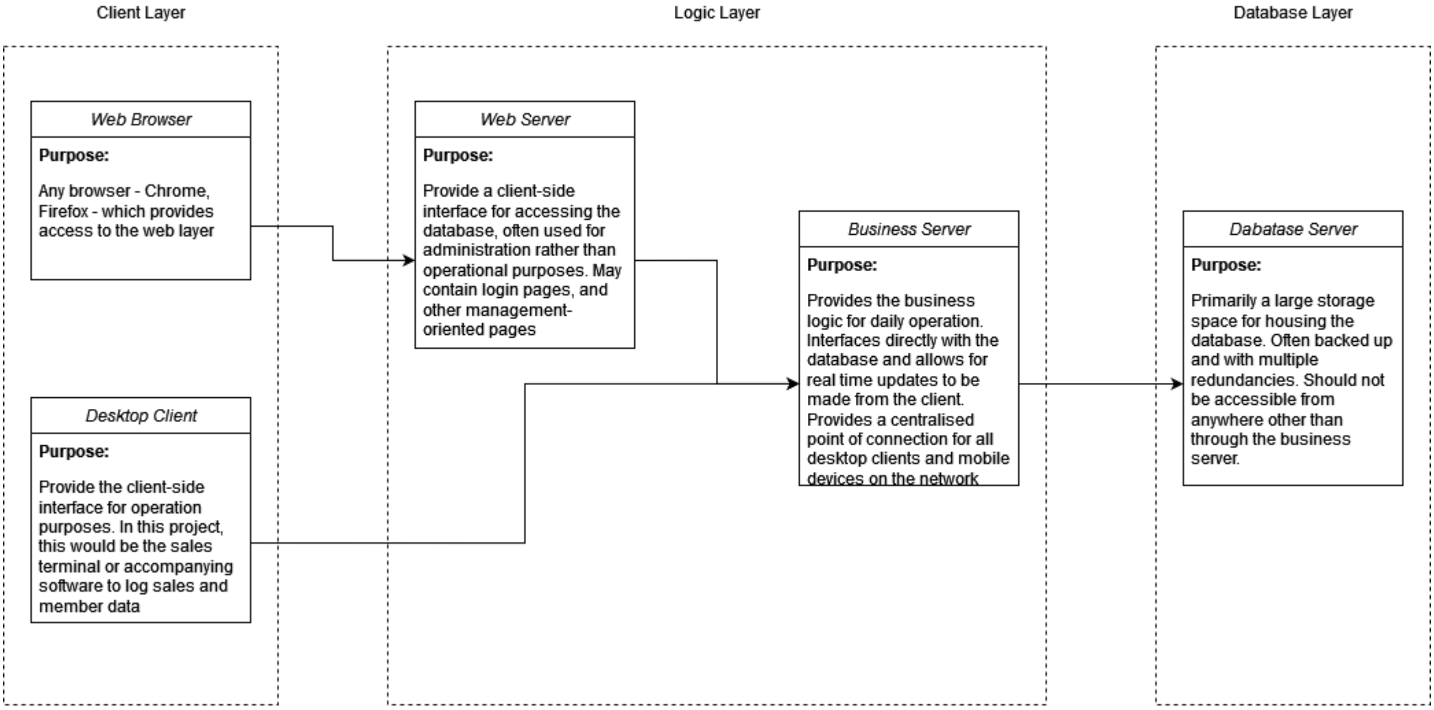
The high-level design is broken down into three layers:

The client layer (application layer), which provides a UI and a visual method of interacting with the system.

The logic layer, where business logic takes place and drives things like calculation and reporting.

The database layer, which houses the raw business data.

Together they create an architecture as follows:



Web Browser
Purpose:
Any browser - Chrome, Firefox - which provides access to the web layer

Desktop Client
Purpose:
Provide the client-side interface for operation purposes. In this project, this would be the sales terminal or accompanying software to log sales and member data

Web Server
Purpose:
Provide a client-side interface for accessing the database, often used for administration rather than operational purposes. May contain login pages, and other management-oriented pages

Business Server
Purpose:
Provides the business logic for daily operation. Interfaces directly with the database and allows for real time updates to be made from the client. Provides a centralised point of connection for all desktop clients and mobile devices on the network

Dabatase Server
Purpose:
Primarily a large storage space for housing the database. Often backed up and with multiple redundancies. Should not be accessible from anywhere other than through the business server.

Member Comments

Table 5. Member Comments

Name	Description
Dylan	I agree with the proposed models, we have the skills to cover all foreseeable complications so we should be able to complete the project in the timeframe.
Simon	I believe that the design of the project is well thought out and accomplishes its goal of outlining the requirements and scope as well as neatly delineating the strengths and weaknesses of our team.
Rabya	I believe the project design is neatly split into client, logic and business layers and the solution design is very good security wise as well. Based on the team member skills, the project seems doable within the given timeframe.
Cody	Our design has been selected and broken up effectively to allow for an effective end product and for many tasks that each team member can be assigned to based on their strengths/ weaknesses in different skills.
Thomas	I think the design of the project is well done and based upon our team member skills we'll be able to complete each aspect of the project to a high standard.
Nic	I think the solution direction proposed make perfect sense and breaks down all key elements into simple components. Using gap analysis and comparing these components with our team skills we have a good split between all necessary skills required to deliver this project successfully. In the high-level design I think the logic layer objects, web server and business server should be one object and not separate. I think one central business server would be able to perform both objects' roles and make the web server redundant. I understand the separation of the two, but it may not be needed in the final design.