Project Proposal

Sales Reporting and Prediction System

MEMBERS	STUDENT ID
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
Marcus Rakkhit	[redacted]
[redacted]	[redacted]

Tutorial - Monday 2:30pm EN402 with [redacted]

Sprint Items	Justification
Add a record to the database	Most of the product backlog items in this project require that the database contains a record in order for it to be displayed, edited, processed etc. Therefore, there is a dependency on this task. Also, a subtask for this item will be that we have designed a database and its schema, which the other items are also dependent on.
Usable Graphical Interface	Firstly, the business value is high as it is a feature explicitly asked for by the client. Task 4, 5, 8 and 9 from the product backlog are dependent on this task (and have very high business value); As a result, it's critical that this task is one of the first to be completed.
Display a record from the database	The business value is relatively high as a main requirement for the client is that they can view data from their inventory. Task 5 (Showing a sales report) depends on this functionality.
Show the weekly/monthly sales	The business value is very high for this task. The client explicitly states they need this functionality
Predict the weekly/monthly sales of an item	This is another core function that the client requested.This is related to the actual logic of predicting the sales, not the displaying of the report. Therefore this task is required before the application can "show the weekly/monthly sales".
Generate a weekly/monthly sales report in a CSV file	The business value for this is high as it's an explicitly requested function. Although high priority, this item requires that the application can predict and display the weekly/monthly sales of an item first.

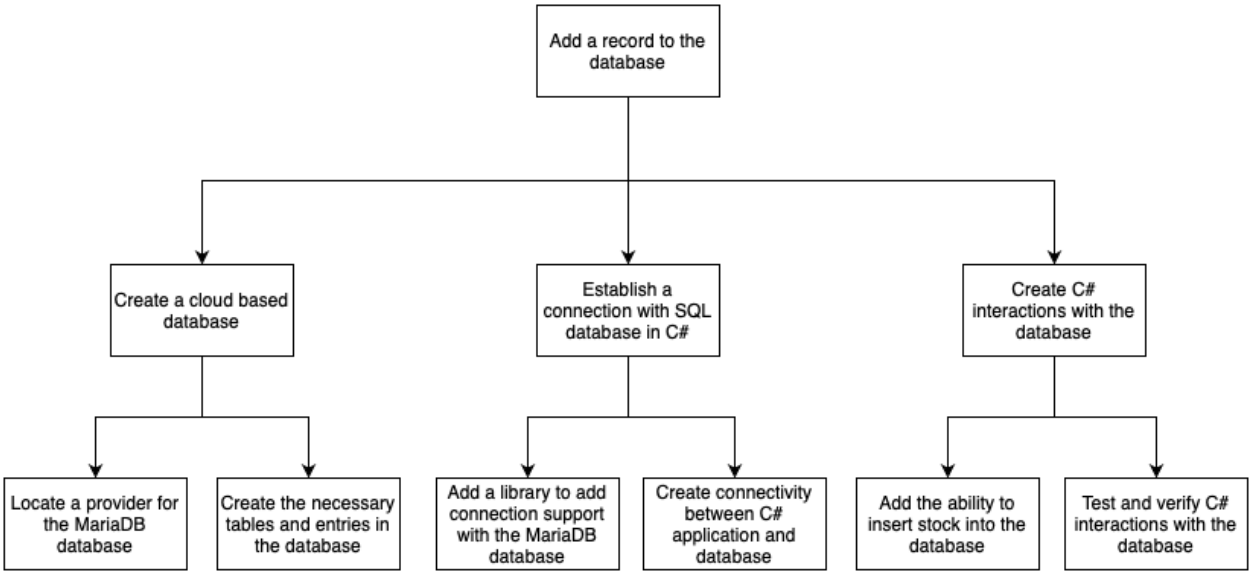
Edit a record to the database	It's important for an inventory application to have the ability to edit items in the database, therefore this is a core function we'd like to have done early on.

Highest Priority Item

Out of all the items listed in the sprint backlog above, we believe adding a record to the database is one of the most essential items to be delivered first. As stated above, it also requires that a database is designed and created before this functionality other backlog items can even be implemented.

Work Breakdown Structure

Adding a Record to the Database



WBS	Time Required	Justification
Create a cloud based database	1 hour	The database needs to be created on a remote server and needs to have the necessary tables created for interaction with the C# application.
Locate a provider for the database	30 minutes	An appropriate provider for the database is a relatively low complexity task, however finding a sustainable and affordable solution may take time.
Create the necessary tables and entries in the database	30 minutes	The complexity of adding the appropriate tables and fields to the database has an intermediate level of complexity and some background knowledge.
Establish a connection with SQL database in C#	1 hour	The C# application needs to add a library to interact with the database, and then connectivity with the server must be verified.
Add a library to add connection support with the MariaDB database	30 minutes	Finding and integrating the appropriate C# library to communicate with the MariaDB database has an intermediate level of complexity and will require research.
Create connectivity between the C# application and the database	30 minutes	Ensuring the communication works with the database will involve a relatively low level of complexity once the necessary research has been completed.
Create C# interactions with the database	2 hours	The C# application must have necessary methods and logic to modify the database when desired, and the interactions must be verified that they're being asserted correctly, and not malformed.

Add the ability to insert stock into the database	1.5 hours	The ability to insert the necessary data from the C# application and into the database is critical to the functionality of the application. The complexity of this task is high and likely time consuming to understand the correct syntax.
Test and verify interactions with the C# database	30 minutes	Verification that the database is working will require an intermediate level of complexity, due to debugging, however verifying the correct results should be relatively simple.