Sprint 1 planning: Management System (GotoGro-MRM) for Goto Grocery Inc.

Task 1:

Included factors:

Feature Dependency

This describes any dependencies that a sprint backlog item has. An example of this would be the backlog item of deleting users. This has a feature dependency of Adding new users as it is impossible to delete users if you cannot yet add them.

Why this factor is important

This is a critical factor for this project as without it we have no visibility as to if a backlog item is ready to be worked on. By prioritising feature dependency, we can easily compare and prioritise backlog items.

Development Effort

Development effort encompasses the time, skills, and resources necessary to complete a project. It relies on factors like task complexity and team expertise and plays an essential role in shaping project plans and sprint schedules.

Why this factor is important

Accurate development effort estimation is important for effective resource allocation, realistic timeline setting, and the fair distribution of tasks based on team members' experience and capacity. It ensures that projects are approached with clarity and foresight, reducing risks, enhancing project outcomes, and fostering a more efficient and sustainable development process.

Business Value

This represents a backlog item's overall value to the business, its assets, brand recognition, goodwill and operations. Examples of a backlog item that would be considered to have high business value include:

- A feature that would give the product a competitive edge
- A fix for a bug that customers have been complaining about for some time

Why this factor is important

This is an important factor to consider when prioritising backlog items. By considering the value to the business it ensures that our work stays tightly in scope and focused on the core deliverables of the project. Only once with high value to the business are completed does it make sense to prioritise more quality of life features.

Not-Included Factors:

Date Needed or Timeline

This describes prioritising backlog items depending on the relevant times they are required. If a client required a certain feature by next month then this would be prioritised first.

Why is this not included?

We decided that this was not distinct enough for our project brief to be identified as a seperate factor. This is because of its similarity with the feature dependency factor. It is very likely that the dates needed for a certain backlog item directly correlate with that item's dependencies.

Risk involved

This refers to the potential issues or threats associated with implementing a backlog item. Risk can arise from various factors such as technical challenges, security vulnerabilities, operational disruptions, or factors external to the project like market changes. Understanding the risks associated with each item allows the team to plan and allocate resources more efficiently, ensuring smooth delivery.

Why is this not included?

- ➔ **Internal Nature of the Project**: Given the project is internal, the risk exposure is inherently low. The system isn't outward-facing, which means it isn't a high-priority target for potential external threats. This allows us to work with a certain level of ease without being overly concerned about external vulnerabilities.
- ➔ **Security/Privacy**: Thanks to our reliance on trusted entities with recognized certifications such as ISO/IEC 27001, ISO/IEC 27017, ISO/IEC 27018, ISO/IEC 27701, SOC 2, and SOC 3, we are confident in the robust security and privacy measures in place. These measures align with industry best practices, as outlined in the Google Cloud security overview whitepaper (Google Cloud, n.d.). This foundation negates the need to further emphasise or allocate resources towards this factor.
- ➔ **Task Dependency at Runtime**: When our system goes live and enters its runtime phase, most tasks are built to function independently. Their modular nature ensures smooth operations, with minimal risk of one task's failure or delay affecting the rest. This lack of interdependence at runtime is a significant advantage, ensuring that our system can remain robust and functional even if isolated issues arise.

- ➔ **Stable Development Environment**: Our team operates in a stable development environment with established tools, protocols, and a skilled team. This environment minimises the risk of unforeseen technical challenges or disruptions, further ensuring a smooth developmental progression.

Other factors

Why are other factors not considered?

- ➔ **Simplicity and Clarity**: By focusing on just the three main factors, we can ensure that our ranking system remains simple and clear. Introducing more factors would not only complicate our decision-making process but could also create confusion and disputes among team members about the relative importance of each factor.
- ➔ **Defined Scope**: Our project has a very defined and limited scope. As such, introducing more factors might not be entirely relevant or beneficial. It's essential to maintain a clear vision and direction, and not get sidetracked by factors that, while they might be important in other contexts, do not necessarily add value in the context of our specific project.
- ➔ **Efficiency in Decision Making**: Every additional factor we consider requires time to assess, debate, and weigh against others. By limiting our focus to three primary factors, we streamline our decision-making process, allowing the team to quickly and efficiently prioritise backlog items.
- ➔ **Avoiding Paralysis by Analysis**: Too many factors could lead to over-analysis, where the team spends more time debating the importance of backlog items than actually working on them. In essence, we're avoiding the pitfall of being "stuck in planning" and ensuring that we move into the action phase more seamlessly.
- ➔ **Uniformity in Understanding**: With fewer factors to consider, there's a higher chance that all team members have a uniform understanding of what's important. This uniformity ensures everyone is on the same page, reducing the likelihood of misunderstandings or disagreements later on.

Task 2:

We have decided that the criteria for prioritising the backlog items will occur in the following order:

1. Feature Dependency
    - Defined by future backlog items dependencies on the completion of this item
2. Development Effort
    - Defined by the estimated number of hours of developer effort required to complete a backlog item to the standard defined within the agreed DoD
3. Business Value
    - Defined by a backlog items relevance to the defined scope of the project and stakeholders demand for its completion

It is important to note that the ordering of these factors is specific to the project's scope, goals and stakeholders. In another project, such factors may be omitted or be prioritised differently. Our team

decided that the dependencies that a particular backlog item has outweighs the other factors listed. This is due to the greenfields nature of the project resulting in many backlog items having large dependencies on each other. For example, due to the lack of existing infrastructure to build on, the team will have to carefully prioritise the first sprint to ensure there are minimal blockers throughout while completing enough to unlock backlog items for future sprints.

Development effort has been prioritised second. Due to the small size of the team it is crucial that we effectively manage the little development time we have available. As such, the team will have to thoroughly evaluate the time commitments of each of the backlog items to see if the time should be better spent elsewhere. This was prioritised underneath feature dependencies. This decision is a natural choice considering a backlog item with many dependencies would need to be prioritised regardless of the development effort it requires. In other words, it would be required to do the 'hard work' first in order to unblock the higher value items later on.

Lastly we have prioritised business value as a third factor. This acts as a litmus test for any items that are deemed to be equivalent for all the preceding factors. Where two items are equivalent, the team will discuss the value a given backlog item aims to deliver to the overall vision of the project. This prevents the team from working on anything that may have feature dependencies but ultimately of little value to the overall project. This has been prioritised last due to the delivery of this project being a single large showcase rather than iterative delivery of features. If features were delivered iteratively business value would make sense to prioritise higher as the goal of the sprints would be to get a minimum viable product working as quickly as possible. However, given this is not the case, the team has the ability to spend time instead laying solid foundations in the unblocking of dependencies, enabling fast growth in later sprints

Task 3:

Ranking	Layer	Epic		No.	Item	Feature Dependency	Development Effort	Business Value	Estimated points*
1	Back-end	DBMS	Create database tables	F1	Create database table for members	★★★★★ (Core foundation feature)	★★★★☆ (High data sensitivity)	★★★★★ (Digitizes member records.)	3
2				F2	Create database table for sales records	★★★★★ (Core foundation feature)	★★★★☆ (High data sensitivity)	★★★★★ (Tracks member purchases.)	3
3				F3	Create database table for products	★★★★★ (Core foundation feature)	★★★★☆ (High data sensitivity)	★★★★★ (Centralizes product inventory.)	3
4			Member API endpoints	F4	Create endpoint for adding members	★★★★☆ (Vital for user registration)	★★★☆☆ (Standard CRUD operation)	★★★★☆ (Simplifies member registration.)	3
5				F5	Create endpoint for editing members	★★★★☆ (Crucial for data accuracy)	★★★☆☆ (Standard CRUD operation)	★★★★☆ (Enhances member data management.)	3
6				F6	Create endpoint for removing members	★★★★☆ (Necessary for data management)	★★★☆☆ (Standard CRUD operation)	★★★★☆ (Supports member removal.)	3
7				F15	Create endpoint for retrieving members	★★★★☆ (Key for accessing member info)	★★★☆☆ (Standard CRUD operation)	★★★★☆ (Fetches member information.)	3
8			Sales API endpoints	F7	Create endpoint for adding sales records	★★★★☆ (Vital for sales tracking)	★★★☆☆ (Standard CRUD operation)	★★★★☆ (Streamlines sales data entry.)	3
9				F8	Create endpoint for editing sales records	★★★★☆ (Crucial for data accuracy)	★★★☆☆ (Standard CRUD operation)	★★★★☆ (Facilitates sales record edits.)	3

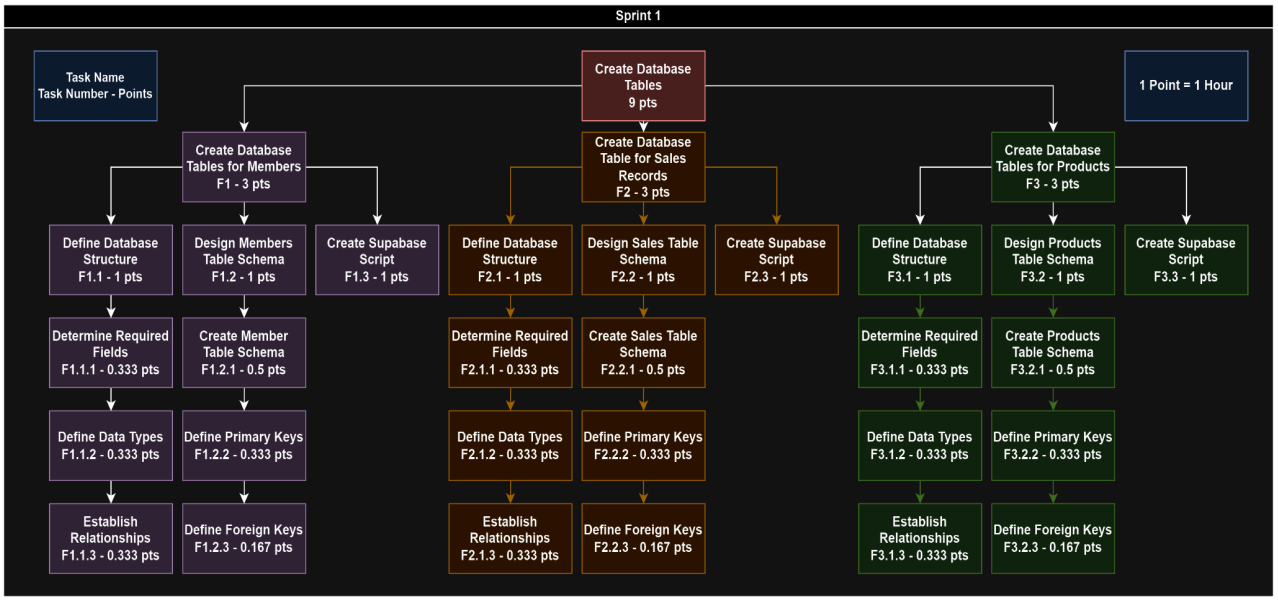
10				F9	Create endpoint for removing sales records	★★★★☆ (Necessary for data management)	★★★☆☆ (Standard CRUD operation)	★★★★☆ (Manages sales record deletion.)	3
11				F14	Create endpoint for retrieving sales	★★★★☆ (Key for sales insights)	★★★☆☆ (Standard CRUD operation)	★★★★☆ (Accesses sales data.)	3
12			Product API endpoints	F10	Create endpoint for adding products	★★★★☆ (Supports inventory growth)	★★★☆☆ (Standard CRUD operation)	★★★★☆ (Enables product additions.)	3
13				F11	Create endpoint for editing products	★★★★☆ (Ensures product data accuracy)	★★★☆☆ (Standard CRUD operation)	★★★★☆ (Assists in product edits.)	3
14				F12	Create endpoint for removing products	★★★★☆ (Maintains product integrity)	★★★☆☆ (Standard CRUD operation)	★★★★☆ (Aids product removal process.)	3
15				F13	Create endpoint for retrieving products	★★★★☆ (Key for viewing product range)	★★★☆☆ (Standard CRUD operation)	★★★★☆ (Retrieves product listings.)	3
16		Report and Analytics		F17	Generate sales reports	★★★★☆ (Provides sales overview)	★★★☆☆ (Complex data aggregation)	★★★★☆ (Provides insights & CSV compatibility.)	8
17		Report and Analytics		F32	Generate inventory reports	★★★★☆ (Vital for inventory management)	★★★☆☆ (Data gathering & formatting)	★★★★☆ (Provides insights & CSV compatibility.)	8
18		User Authentication		F18	User authentication and account management	★★★☆☆ (Key for secure access)	★★★★★ (High complexity due to security concerns)	★★★★☆ (Secures user access.)	8

19	Front-end	Member management web page	F33	Build web page to allow users to add new members	★★★☆☆ (Enhances member addition process)	★★★★☆ (Front-end design & functionality)	★★★★☆ (Enhances member management UX.)	5
20			F34	Build web page to allow users to edit existing members	★★★☆☆ (Ensures data accuracy & usability)	★★★★☆ (Front-end design & functionality)	★★★★☆ (Enhances member management UX.)	5

*1 point = 1 hour

Task 4:

Figure 1 - Work Breakdown Structure



References:

Google Cloud, n.d. *Google security overview*. [online] Available at: https://cloud.google.com/docs/security/overview/whitepaper [Accessed 4 September 2023].