Estimation Effort: GotoGro-MRM

Sprint Backlog Item WBS

The sprint backlog item selected for this prediction is the “Add new Member Interface”. This item can be broken down into numerous product backlog items as shown in **Figure 1**:

5	Critical	Member Table	Create Member Table	Member table with data validation - NOT NULL, etc - which records member details	-	Dylan	0.3	
6	Critical	Member Table	Create Queries to Add Member	Simple query to add member, will be triggered by visual fields on the UI	5	Dylan	0.3	
7	Minor	Member Table	Create Queries to Delete Member	Simple query to delete member, will be triggered by visual fields on the UI	5	Dylan	0.3	
8	Minor	Member Table	Create Queries to Modify Member	Simple query to modify member, will be triggered by visual fields on the UI	5-6	Dylan	0.3	
15	Major	Add/Mod New Member UI	Input Fields for All Member Details	Text checking on input fields to minimise the chance of invlid data being entered	5-6	Rabya	4	
16	Major	Add/Mod New Member UI	Confirm Button	Button to confirm the member details, checks the inputs then sends it to the member table	15	Nic	2	
17	Critical	Add/Mod New Member UI	Autoincrementing Member ID	When the confirm button is pressed the member ID is automatically generated and added to the database	16	Rabya	2	
18	Major	Add/Mod New Member UI	Viewport to View Members	Snapshot of the member table, needs to be able to be filtered by search interface	5	Rabya	2	
19	Major	Add/Mod New Member UI	Search Input Field	By typing member ID in and confirming, the viewport will display the member searched for (or nothing if no results found)	18	Simon	2	
20	Major	Add/Mod New Member UI	Modify a Member Record	Selecting the searched member result will populate the text fields with saved data. Writing over these with new information and confirming will save over the old record with the new information	8, 15-19	Simon	4	

Figure 1: *Product backlog of the Add New Member Interface.*

As can be seen, there are multiple components involved, broken down into two main groups being the database and the UI. This can be more easily seen in diagrammatic form:

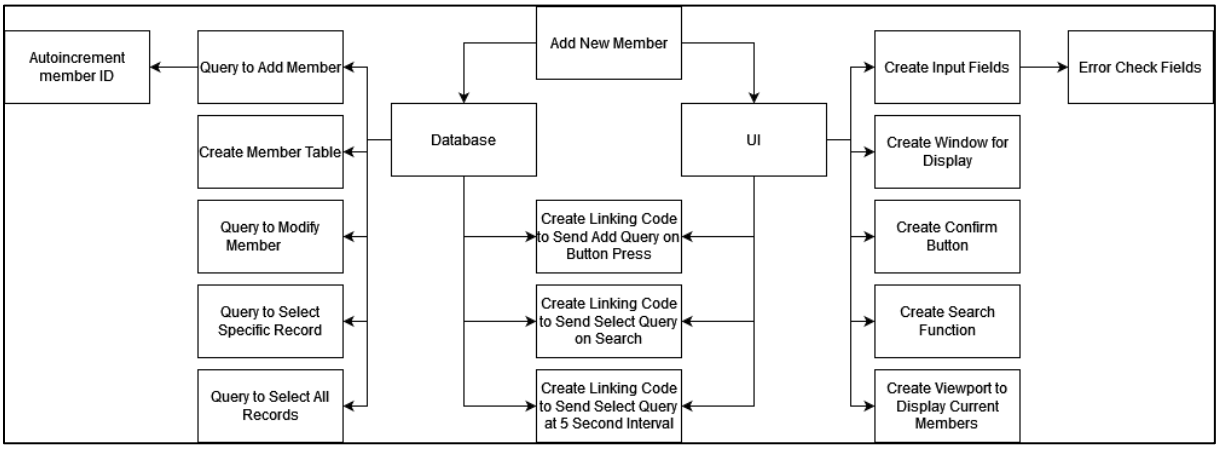

Figure 2: *WBS diagram.*

Again, referring to **Figure 1**, these objects are mostly associated with an estimated timeframe derived from doing similar things. We can confidently say that it will take approximately 20 minutes to create and test all the relevant SQL statements as the team has experience in SQL.

Where the tasks are more complex and specifically where the team’s knowledge on a topic cannot give us complete confidence in every aspect of the backlog item, the times have been purposely overestimated.

For components like the UI, extra time has been factored in to account for learning of the UI software the team is using (Ignition), a product which only one developer is familiar with. For the purpose of this assignment, we will assume that the most skilled team member is NOT working on this specific backlog item.

Among these items, we can see a vague hierarchy, specifically where the SQL and database must be done first, followed by the visual input fields and finally the linking code. Of note, in Figure 1, there are two items with a listed time of 4 development hours. In a proper sprint

model, each item should be achievable within a single day (or in our case, 2 hours). To this effect, those items are separated in **Figure 2** as the UI component and the requisite linking code, both being allocated 2 hours each.

To clarify the linking code, these will be snippets of code linked to UI components which will trigger when said UI component is activated, often by clicking on it, but also through other methods such as typing into a text box. The linking code will be written in python.

Actual Estimate

With each member performing 6-8 hours per week over the sprint, we can make an accurate estimate of how log this feature may take to implement, and how many developers need to be engaged to complete the entire item before the end of the sprint. **Figure 3** shows a visual indicator of each time estimate:

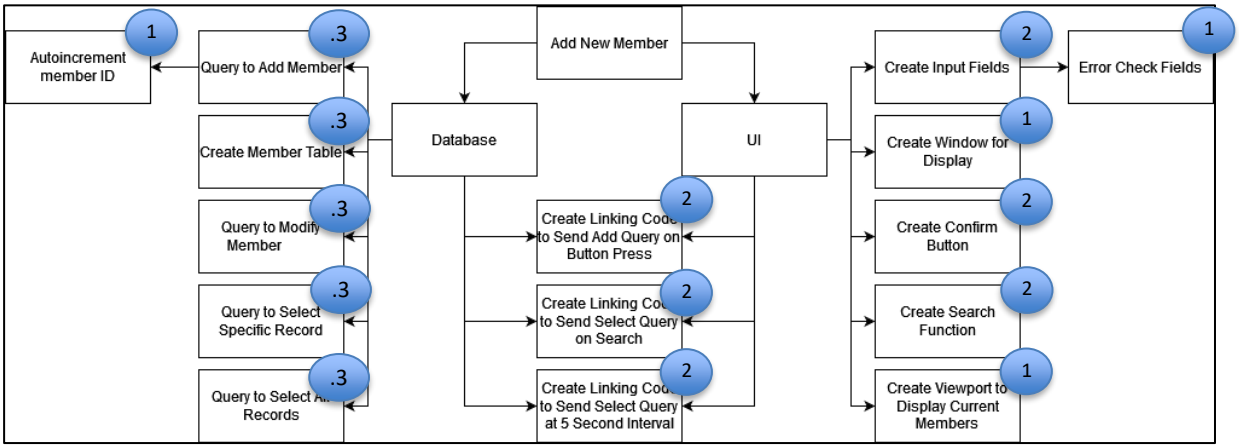

Figure 3: *Product backlog items with realistic time estimations.*

This sums to 18 hours of development effort (rounded up for leeway). With this diagram it is easy to see that this backlog item can't be realistically tackled by a single developer over the course of one sprint.

Ideally, two or three team members need to be engaged. This works well as the database and UI components can be built separately and concurrently. The only tasks requiring both to be done are the linking code sections.

With this plan in mind and given the 2 hours daily workload of a team member, the database components would be finished on Day 2 worked by a single developer. The UI components sum to 9 hours of work, meaning they are not achievable by a single developer. Using two developers however, the components would be completed by Day 3.

After this, the final work can be started on linking code. If all three developers came together these could be finished by the end of Day 4, however it is more likely that only 2 team members will be allocated to work at this stage. That would leave the entire component complete by Day 5.

In summary, the Add/Modify Member Interface is estimated to be completed to a demonstrable standard by Day 5 of sprint 1 using 3 developers over 3 days, and 2 developers for the next 2 days.