SWE20001: Managing Software Projects

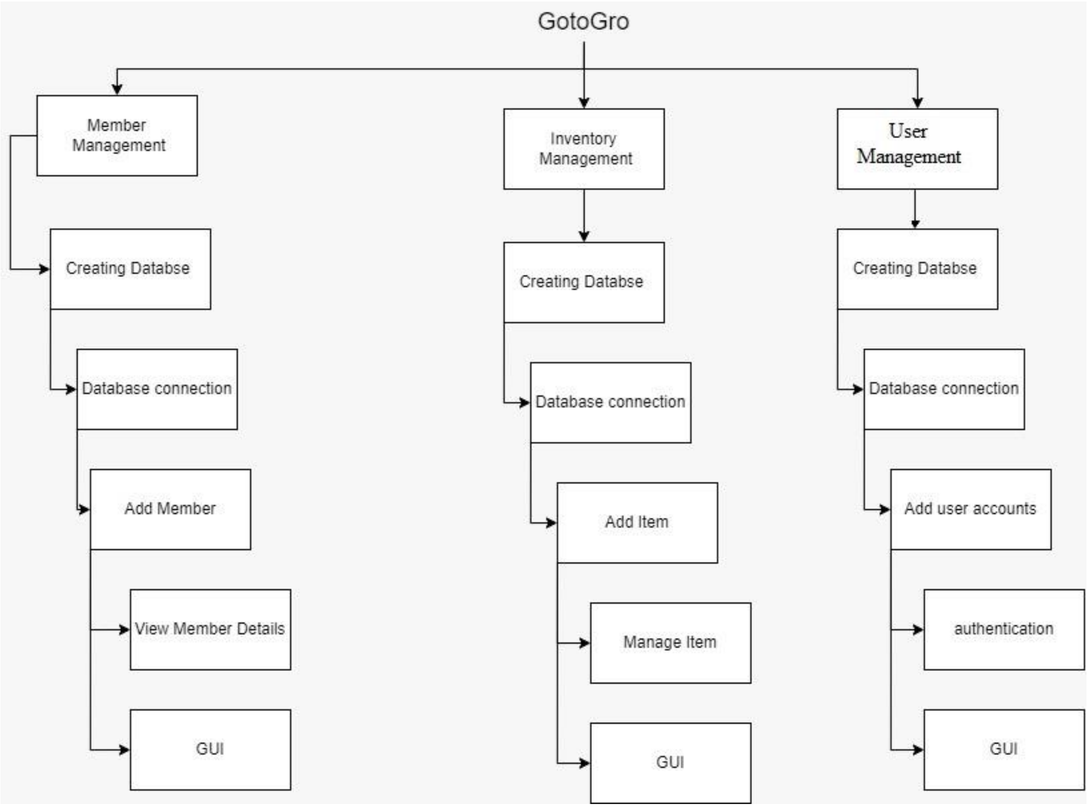
Name: S M Ragib Rezwan

ID: 103172423

Tutor: Naveed Ali | 12.30pm Tuesday | EN31

Project Proposal1 : GotoGro :

In the group work task (08P), we had decided upon the following WBS where we have decided to use product based approach to ensure that no product planned to be developed in sprint 1 is missing from the task breakdown structure/chart.



Here, you can see that we have the following leaf nodes that we must create by the end of Sprint 1:

1. Creating Database for Member Management
2. Creating the connection of the Database for Member Management
3. Creating the functionality and logic to add Member to that database
4. Creating the functionality and logic to view the member details of the records in the member database
5. Creating the GUI feature for adding and viewing the Members on the Client side
6. Creating Database for Inventory Management

7. Creating the connection of the Database for Inventory Management
8. Creating the functionality and logic to add Item to that database
9. Creating the functionality and logic to manage the item details of the records in the member database
10. Creating the GUI feature for adding and managing Items on the Client side
11. Creating Database for User Management (ie the admin accounts for the clients)
12. Creating the connection of the Database for User Management
13. Creating the functionality and logic to add User accounts to that database
14. Creating the functionality and logic to manage the authentication of the user accounts in the database
15. Creating the GUI feature for entering the software with User account privilege on the Client side

Now here, considering Ideal Time, Ideal Effort and Effective effort, we can notice the following table:

[Note: here I am using the fact the each person will only be able to productively work for 70% of their time and using that in my calculation of real effort for each of the leaf nodes.]

[Note the leaf node numbers refer to their task names here and I have not written their names as it would be too long and their brief abbreviation would be confusing]

Task no (following the list created above)	**Ideal Time**	**Ideal Effort**	**Real Effort**	**Justification**
1	60 mins	60 mins	120mins	Here we are creating the database for the Member Management in C# and then verify that it works by using a test case table (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
2	30 mins	30 mins	60 Mins	Here we are creating the connection for the database for the Member Management in C# and then verify that it works by using a test case table and querying for viewing data (this will be removed later on after troubleshooting is over). Although this should ideally take 30 mins; considering the fact that we are working in C# and the different types of

				errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
3	30 mins	30 mins	60 Mins	Here we are creating the functionality and logic to add Member to that database in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 30 mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
4	30 mins	30 mins	60 Mins	Here we are creating the functionality and logic to view the member details of the records in the member database in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 30 mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
5	60 mins	60 mins	120 Mins	Here we are creating the GUI feature for adding and viewing the Members on the Client side in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
6	60 mins	60 mins	120mins	Here we are creating the database for the Inventory Management in C# and then verify that it works by using a test case table (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the

				fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 1, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
7	30 mins	30 mins	60 Mins	Here we are creating the connection for the database for the Inventory Management in C# and then verify that it works by using a test case table and querying for viewing data (this will be removed later on after troubleshooting is over). Although this should ideally take 30 mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 2, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
8	30 mins	30 mins	60 Mins	Here we are creating the functionality and logic to add Item to that database in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 30 mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 3, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
9	30 mins	30 mins	60 Mins	Here we are creating the functionality and logic to manage the item details of the records in Inventory database in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over).

				Although this should ideally take 30 mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 4, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
10	60 mins	60 mins	120 Mins	Here we are creating the GUI feature for adding and managing Items on the Client side in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 5, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
11	60 mins	60 mins	120mins	Here we are creating the database for User Management (ie the admin accounts for the clients) in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 1, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
12	30 mins	30 mins	60 Mins	Here we are creating the connection for the database for the User Management in C# and then verify that it works by using a test case table and querying for viewing data (this

				will be removed later on after troubleshooting is over). Although this should ideally take 30 mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 2, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
13	30 mins	30 mins	60 Mins	Here we are creating the functionality and logic to add User to that database in C# and then verify that it works by using test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 30 mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 3, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
14	60 mins	60 mins	120 Mins	Here we are creating the functionality and logic to manage the authentication of the user accounts to the User database in C# and then verify that it works by using test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 60 mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
15	60 mins	60 mins	120 Mins	Here we are creating the GUI feature for entering the software with User account privilege on the Client side in C# and then verify that it works by using test case data (this will be removed later on after troubleshooting is over).

	Although this should ideally take 60mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.

Total time taken (with respect of the WBS diagram): **22 hrs** (=1320 mins)

Now, in our project we are planning on using pair programming method where two people will be tasked to code for a single functionality. This is done for three reasons:

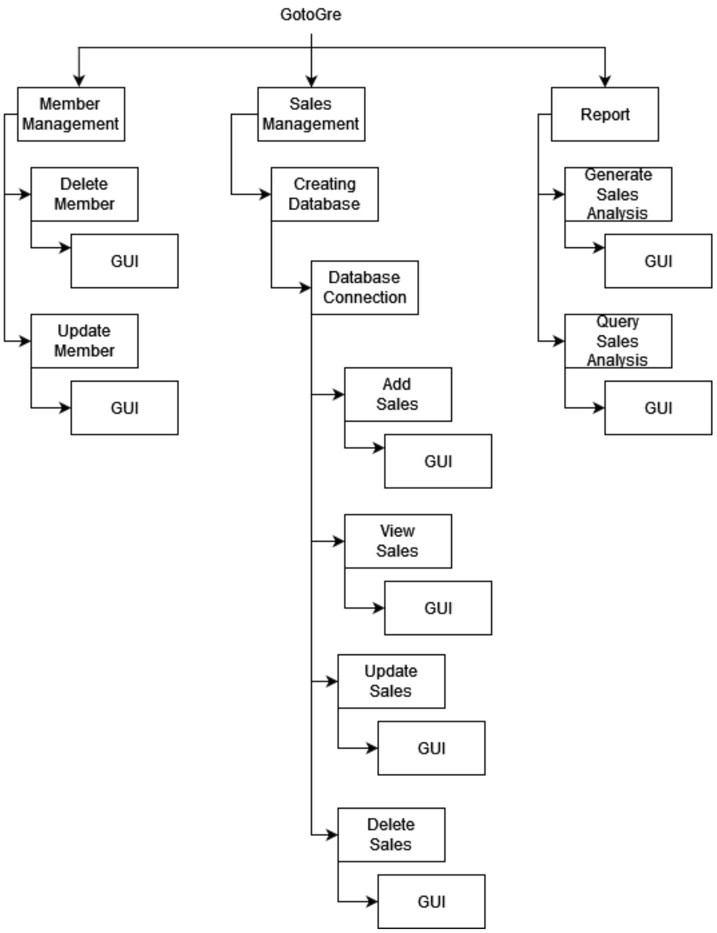
a) Following the information in the lecture about “trunk factor” in risk and by also having firsthand experience of it (as a member of our team had been in a car crash in the early weeks and thus had missed for several later weeks due to recuperation), we have decided to proceed with the pair programming method in order to ensure that even if one person is unable to deliver, the other one can and thus the functionalities present in sprint 1 can be delivered in time (whilst fulfilling the definition of done related to them)

b) Different people have different way of approaching a single problem and thus can come up with different unique backend logic to provide the same outcome. Thus by using this system we can ensure that we can have a variety of method logics and thus can choose the best one in a short period of time

c) Last but not the least, our team has issue in the skills required for the task as almost half of our members have never done such a project before from end to end, let alone in C#. So by following this technique, we can provide them with the time and guidance needed to hone their skills, whilst not delaying the progress of the project itself.

But even so the time we come up is **44hrs** (= 2 * Total time taken) which is short by our intended teamtime of **96 hrs** by 52 hrs. Thus I believe we will end up having to add the following functionalities as well from the sprint 2:

(WBS of the other remaining functionalities)



[Since task has been added to the previous list, the numbers will start from 16 here]

16. Creating the functionality and logic to delete Member to that database
17. Creating the GUI feature for deleting the Members on the Client side
18. Creating the functionality and logic to update Member to that database
19. Creating the GUI feature for update the Members on the Client side
20. Creating Database for Sales Management
21. Creating the connection of the Database for Sales Management
22. Creating the functionality and logic to add sales record to that database
23. Creating the GUI feature for adding the sales record on the Client side
24. Creating the functionality and logic to view sales record to that database
25. Creating the GUI feature for viewing the sales record on the Client side
26. Creating the functionality and logic to update sales record to that database
27. Creating the GUI feature for updating the sales record on the Client side
28. Creating the functionality and logic to delete sales record to that database
29. Creating the GUI feature for deleting the sales record on the Client side
30. Creating the functionality and logic to generate sales analysis
31. Creating the GUI feature for generating sales analysis on the Client side

[Since 25 hrs (50 hours considering pair programming) get fulfilled by the 31st point, I am not including the point no 32 and 33 in the ideal time, ideal effort and effective effort table]

32. Creating the functionality and logic to query sales analysis
33. Creating the GUI feature for querying sales analysis on the Client side

The Ideal Time, Ideal Effort and Effective effort of them are the followings:

[Note: here I am still using that 70% productivity aspect in my calculations]

[Note the leaf node numbers refer to their task names here and I have not written their names as it would be too long and their brief abbreviation would be confusing]

Added Task no (following the list created above)	Ideal Time	Ideal Effort	Real Effort	Justification
16	30 mins	30 mins	60 Mins	Here we are creating the functionality and logic to delete Member in Member database in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 30 mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 4, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
17	60 mins	60 mins	120 Mins	Here we are creating the GUI feature for deleting member on the Client side in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some

				leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 5, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
18	30 mins	30 mins	60 Mins	Here we are creating the functionality and logic to update Member in Member database in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 30 mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 4, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
19	60 mins	60 mins	120 Mins	Here we are creating the GUI feature for updating member on the Client side in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 5, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
20	60 mins	60 mins	120mins	Here we are creating the database for the Sales Management in C# and then verify that it works by using a test case table (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that we are working in C# and the different types of

				errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 1, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
21	30 mins	30 mins	60 Mins	Here we are creating the connection for the database for the Sales Management in C# and then verify that it works by using a test case table and querying for viewing data (this will be removed later on after troubleshooting is over). Although this should ideally take 30 mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 2, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
22	30 mins	30 mins	60 Mins	Here we are creating the functionality and logic to add Sales records to that database in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 30 mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 3, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
23	60 mins	60 mins	120 Mins	Here we are creating the GUI feature for adding sales record on the Client side in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the

				fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 5, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
24	30 mins	30 mins	60 Mins	Here we are creating the functionality and logic to view Sales records to that database in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 30 mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 3, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
25	60 mins	60 mins	120 Mins	Here we are creating the GUI feature for viewing sales record on the Client side in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 5, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
26	30 mins	30 mins	60 Mins	Here we are creating the functionality and logic to update Sales records to that database in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over).

				Although this should ideally take 30 mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 3, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
27	60 mins	60 mins	120 Mins	Here we are creating the GUI feature for updating sales record on the Client side in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 5, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
28	30 mins	30 mins	60 Mins	Here we are creating the functionality and logic to delete Sales records to that database in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). Although this should ideally take 30 mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 3, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
29	60 mins	60 mins	120 Mins	Here we are creating the GUI feature for deleting sales record on the Client side in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over).

				Although this should ideally take 60mins; considering the fact that we are working in C# and the different types of errors that usually pop up, we have decided to keep some leeway time for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively. **[Note: But since we have done similar task for task no 5, it might end up taking half time in reality. But that is something that must be verified first during this sprint]**
30	60 mins	60 mins	120 Mins	Here we are creating the functionality and logic to generate sales analysis in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). **This is a new thing that haven't been done by any of the members in the group before and thus I am giving it double ideal time, ideal effort and real effort time when compared to other functionality aspects. Also keeping in mind that each person would only be working 70% productively.**
31	60 mins	60 mins	120 Mins	Here we are creating the GUI feature for generating sales analysis in C# and then verify that it works by using a test case data (this will be removed later on after troubleshooting is over). **Although I have noted the ideal time to be 60 mins and hence calculated real effort to be 120mins, this may not be accurate. That's because it is a new thing and so it might take even exceed the assumed time.**

In total the hours taken to perform the tasks 16 to 31 is 25hrs. This means, following pair programming method, it will be 50 hrs. Thus when combined with the pair programming time for the task 1 to 15, we get total effective effort time as **94 hrs**.

Since the team's effective time for the 2 week sprint-1 should be 96 hrs, we can only fulfill this by considering all the 31 tasks. So it would be better to modify the backlog item list to include items related to task 16 to 31 (*updating member, deleting member, adding sales record, viewing sales record, updating sales record, deleting sales record and generating sales analysis*) in the sprint-1 as well.