SWE20001: Managing Software Projects

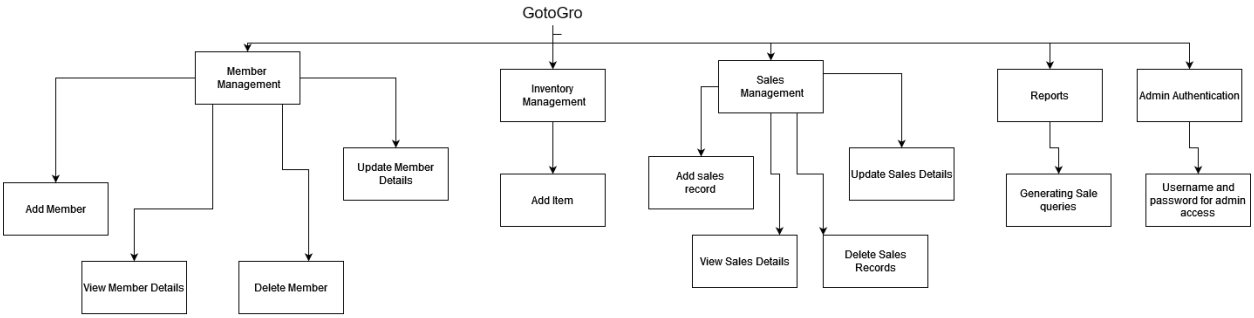
Name: S M Ragib Rezwan

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, although each node will basically take 30 mins of ideal time to be completed. That's not the actual case in reality.

That is because in order to create the first sub-nodes in all of the 4 parent nodes, we need to build the entire backend infrastructure linked to the functionality (from respective database tables, to respective Data Access Layer's entity classes, to respective Business logic layer's classes, to finally the respective method call present in client application class (ie the "main" class)) and also consider time loss due to either error in logic or issue in proper data transition between the different backend infrastructure layers.

Thus in the following times mentioned for the individual items, you shall notice that all the initial subnodes of each parent node are taking up far too long time, compared to the other subnodes present in their respect parent nodes.

Furthermore, I am considering the fact that each person will only be able to productively work for 70% of their time and using that in my calculation of real effort for most of the subnodes. But for all initial subnodes of their own parent nodes, I am keeping extra leeway time in order to consider for the errors that can pop up during the building of the infrastructure

[See below chart for the formula used for calculating the numbers and also the Definition of Done appendix]

Task Name (leaf nodes)	Ideal Time	Ideal Effort	Real Effort	Justification
Add Member	60 mins	60 mins	120mins	This is the first subnode of its respective parent node. So must keep in mind that we will need build its own unique backend infrastructure (which would include their respective tables in database, linked to respective entity class in their data access layers, linked to their respective method calls in the Business Logic Layer's Classes, and finally linked to the main methods being called in the client class (ie the Main class)) and also test to ensure data can not only transition between the layers efficiently and accurately, but also fulfill the requirements set in the Definition of Done that was submitted beforehand. Since this is the buildup stage, it would take a long time in order to set all of these up and ensure things are working as intended. But after that, once all errors have been troubleshot and is working as intended, we would have the necessary foundation to quickly build the rest of the subnodes in that parent class.
View Member Details	30 mins	30 mins	60 Mins	This is one of the remaining subnodes of its own parent node. Since the backend infrastructure has already been build up and is working as intended, we don't need to spend too much time wondering about how things are linked. Instead we can go straight into the logic needed for this functionality and use the linking created in the first subnode before to quickly create this node. Then we would only need to spend the remaining time to modify and test those logics to ensure they are fulfilling the definition of done that has been agreed on beforehand.
Delete Member	30 mins	30 mins	60 Mins	This is one of the remaining subnodes of its own parent node. Since the backend infrastructure has already been build up and is working as intended, we don't need to spend too much time wondering about how things are linked. Instead we can go straight into the logic needed for this functionality and use the linking created in the first subnode before to quickly create this node. Then we would only need to spend the remaining time to modify and test those logics to ensure they are fulfilling the definition of done that has been agreed on beforehand.
Update member Details	30 mins	30 mins	60 Mins	This is one of the remaining subnodes of its own parent node. Since the backend infrastructure has already been build up and is working as intended, we don't need to spend too much time wondering about how things are linked. Instead we can go straight into the logic needed for this

				functionality and use the linking created in the first subnode before to quickly create this node. Then we would only need to spend the remaining time to modify and test those logics to ensure they are fulfilling the definition of done that has been agreed on beforehand.
Add Item	60 mins	60 mins	120mins	This is the first subnode of its respective parent node. So must keep in mind that we will need build its own unique backend infrastructure (which would include their respective tables in database, linked to respective entity class in their data access layers, linked to their respective method calls in the Business Logic Layer's Classes, and finally linked to the main methods being called in the client class (ie the Main class)) and also test to ensure data can not only transition between the layers efficiently and accurately, but also fulfill the requirements set in the Definition of Done that was submitted beforehand. Since this is the buildup stage, it would take a long time in order to set all of these up and ensure things are working as intended.
Add sales record	60 mins	60 mins	120mins	This is the first subnode of its respective parent node. So must keep in mind that we will need build its own unique backend infrastructure (which would include their respective tables in database, linked to respective entity class in their data access layers, linked to their respective method calls in the Business Logic Layer's Classes, and finally linked to the main methods being called in the client class (ie the Main class)) and also test to ensure data can not only transition between the layers efficiently and accurately, but also fulfill the requirements set in the Definition of Done that was submitted beforehand. Since this is the buildup stage, it would take a long time in order to set all of these up and ensure things are working as intended. But after that, once all errors have been troubleshot and is working as intended, we would have the necessary foundation to quickly build the rest of the subnodes in that parent class.
View Sales Details	30 mins	30 mins	60 Mins	This is one of the remaining subnodes of its own parent node. Since the backend infrastructure has already been build up and is working as intended, we don't need to spend too much time wondering about how things are linked. Instead we can go straight into the logic needed for this functionality and use the linking created in the first subnode before to quickly create this node.

				Then we would only need to spend the remaining time to modify and test those logics to ensure they are fulfilling the definition of done that has been agreed on beforehand.
Delete sales Records	30 mins	30 mins	60 Mins	This is one of the remaining subnodes of its own parent node. Since the backend infrastructure has already been build up and is working as intended, we don't need to spend too much time wondering about how things are linked. Instead we can go straight into the logic needed for this functionality and use the linking created in the first subnode before to quickly create this node. Then we would only need to spend the remaining time to modify and test those logics to ensure they are fulfilling the definition of done that has been agreed on beforehand.
Update Sales Details	30 mins	30 mins	60 Mins	This is one of the remaining subnodes of its own parent node. Since the backend infrastructure has already been build up and is working as intended, we don't need to spend too much time wondering about how things are linked. Instead we can go straight into the logic needed for this functionality and use the linking created in the first subnode before to quickly create this node. Then we would only need to spend the remaining time to modify and test those logics to ensure they are fulfilling the definition of done that has been agreed on beforehand.
Generate Sales Queries	60 mins	60 mins	120mins	This is the first subnode of its respective parent node. So must keep in mind that we will need build its own unique backend infrastructure (which would include their respective tables in database, linked to respective entity class in their data access layers, linked to their respective method calls in the Business Logic Layer's Classes, and finally linked to the main methods being called in the client class (ie the Main class)) and also test to ensure data can not only transition between the layers efficiently and accurately, but also fulfill the requirements set in the Definition of Done that was submitted beforehand. Since this is the buildup stage, it would take a long time in order to set all of these up and ensure things are working as intended.
Username and Password	60 mins	60 mins	120mins	This is the first subnode of its respective parent node. So must keep in mind that we will need build its own unique backend infrastructure (which would include their respective tables in database, linked to respective entity class in their data access layers, linked to their respective method calls in the Business Logic Layer's Classes, and finally linked to the main methods being called in the client class (ie the Main

				class)) and also test to ensure data can not only transition between the layers efficiently and accurately, but also fulfill the requirements set in the Definition of Done that was submitted beforehand. Since this is the buildup stage, it would take a long time in order to set all of these up and ensure things are working as intended.

Note:

I am using the following formulae for real effort Calculations: (rounding up values to 30, 60,90 and 120 where needed)

If it is a starting subnode of its parent node:

Ideal Time * 1.5/0.7 = Real effort

If not a starting subnode of its parent node:

Ideal Time/0.7 = Real Effort

Definition of done (Appendix):

Functional suitability: (Functional Completeness). To achieve completeness, the set of functions specified in backlog features (new member, add item, edit/manage item, add sales record, edit/manage sales record, edit/manage member and sales queries based on time stamps and members) should be more than 95% completed and total number of errors per KLOC <=5.

Performance efficiency: (Time Behaviour) The response time for error message/ acceptance message for the employee when checking the "Add New User" functionality should take less than 5 seconds to load, once the software has been fully developed and deployed

Performance efficiency: (Resource Utilization) The amount of memory taken by the backlog features (the GUI aspect, adding and updating members, updating sales record and all other functionality in the sprint backlog for the entire project) should not exceed 10mb for at least 90% of the features.

Usability: (User Interface Aesthetics) The software's menu interface for accessing the "add user", "modify user", "sales record for user", etc (present in sprint 1 backlog) should be simple or self-explanatory enough for the users to easily understand and use (without any further training for it). Let the user, product owner and system testers test the system. If the overall satisfaction of the users demonstrated is more than 90 % then the user interface has been achieved.

Usability: (User Error Protection) If the failed user operation for the system (ie either a failed input for any of the functionality, like “add members”, “update members”, etc, or a cancelled operation) is noted to be less than 5% by the time it has been deployed, the software can be thought to have fulfilled the User Error Protection1

Security: (Confidentiality) The System is protected by username and password pair and only authenticated user can access it after waiting less than 5 seconds (for login time). This can be tested by using a set of username and password pairs and seeing whether more than 95% of the username and password pair can gain access to the system, within 5 seconds of login time.