SWE20001: Managing Software Projects

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Naveed Ali | 12.30pm Tuesday | EN310

Chosen Scenario:

Project Name: Sales Management system for Ragib Televisions (SaleManSys)
Company: Ragib Televisions
Software: Sales Management System (SaleManSys)

Ragib Television is a new television store that has been selling televisions of various sizes locally in Riversdale.

At this moment in time, we are noting down all of our sales details on paper and then creating a trend chart by hand to find the televisions currently in demand. Unfortunately, doing so is becoming too slow for our needs, making it difficult for us to stock up the required televisions in time. Furthermore, as our customer numbers increased, we have also noticed an increase in problems where pages would either get lost or have the same information recorded twice, producing an unreliable chart.

So, we are now looking for software that will streamline the recording of the sales records for us and also help us to find the trend of different products over weekly, monthly and yearly time periods. It must also have a simple Graphical User Interface to ensure that our employees can easily use the system.

Background:

This project is to develop a Desktop application to Ragib Televisions (from further onwards mentioned as client), in order to help them store, manage and analyze sales records. The company bases its business activities at Riversdale where they sell Televisions to their customers. After further analysis we have identified the following problem in the current implementation.

- Inefficiency of activities due the use of paper-based system
- Less accuracy and consistency in storing and analyzing sales details
- Failing to maintain proper system to control television stock

The proposed software will allow the client to store, manage and analyze sales records whilst reducing paper-based system, saving costly overhead. It will also allow the client to keep a higher consistency and accuracy of the records, compared to the current system. This software will be user friendly in manner that current staff will be able to adopt the new system with little or no training provided which have been desired by the client

Scope:

The SaleManSys software will hold the data about the sales (inputted by the employees) in a manner similar to the current paper-based system, but with relevant relationship between all tables (television details table, customer details table, sales details table) so that each relevant records from multiple tables can be accessed and verified instantly (no duplication in data stored).

Since, it will use a Graphical User Interface to do this; the user friendly interface would end up reducing the training time needed for the employees when implementing the system in client's corporate environment.

With the given software, the company can note the trends at different time periods (weekly, monthly and yearly) with the push of a button and thus quickly can adjust their stock of televisions to suit with the customer's demands.

Deliverables and product backlog schedule:

Deliverables:

These will include the followings:

- Source Code
- User Manual
- Prototype
- Test Documents
- Binaries

With the new SaleManSys software, user manuals for operations and troubleshooting will be provided containing all the necessary steps for each scenario and technical details for troubleshooting purposes. Also, when the system will be deployed in the business environment, a brief training session will be provided for employees, which would cover all the functionalities of the system (noted in details in the product backlog below)

Product Backlog: (and explained business value):

1. **Add Customers records:** To add a customer's details (which include their names, address, email and phone numbers) to the company's database' table. This is quite beneficial for the company as they can use these information later on to provide targeted advertisement to their customers in the future and further improve their business worth. Also, it is needed to properly set up the database relationship to ensure duplicate sales records do not exist.

2. **View Customer Records:** To view a customer's details mentioned beforehand. The benefit to the company is same as that noted before in point 1

3. **Edit Customer Records:** To modify a customer's details mentioned beforehand to suit for scenarios like change in customer's address or phone number. This helps business keep track of their customers.

4. **Delete Customer Records:** To delete a customer's details to suit for scenarios where employee had made a mistake in customer creation (like making the same customer twice) and thus give them the chance to remove it.

5. **Add Television records:** To add a television's details (which include their name, productId, price and stock) to the company's database' table. This is beneficial for the company as it can use it to keep track of their stock and know when to replenish it.

6. **View Television records:** To view a television's details mentioned beforehand. The benefit to the company is same as that noted in point 5.

7. **Edit Television records:** To modify a television's details mentioned beforehand to suit for scenarios like change in television's price or stock. This helps business keep track of their products.

8. **Delete Television records:** To delete a television's details to suit for scenarios like company no longer plans to sell a certain television or even keep it in stock due to lack of demand.

9. **Add Sales records** To add the sales' details to company's database' table to help the company make the analysis This is beneficial for the company as they can use this to keep track of customer demands and thus can use it to determine which televisions to keep and how much stock to keep for them.

10. **View Sales records with timestamps:** To view the sales' details mentioned beforehand to help the company make the analysis and thus make the weekly, monthly, yearly trend chart for their products and thus keep their customer demand in mind while stocking up their televisions

11. **Edit Sales records:** To edit the records in the Sales table. This is useful for the company as if the employee made any mistakes in inputting the sales record, they can quickly fix it using this functionality before using the data in their analysis.

12. **Delete Sales records:** To delete the records in the sales table. This is useful for the company for scenarios where the employee had mistakenly inputted the same sales data twice and thus can use this functionality to delete the duplicate record before using the data in their analysis.

13. **Add Admin records:** To add accounts with administrative privileges to the system. This is useful for the company as they can decide which employees should have access to the system and add their details here.

14. **View Admin records:** To view administrative accounts in the system. This is useful for the company as they can check which employees currently have access to the system and see their details here.

15. **Edit Admin records:** To edit administrative accounts in the system. This is useful for the company as they can decide which employees should have access to the system and can update their details here.

16. **Delete Admin records:** To edit administrative accounts in the system. This is useful for the company as they can decide which employees should have access to the system and can remove employees who have left their company

17. **Authenticate Admin records:** To enable the administrative accounts to access the system itself and input in all sales, item and customer details in the system. This is useful for the company as it provides security as only authorized admins can access the system input in data to customer, television and sales.

Product Backlog Schedule:

No.	Item	Dependencies	Business value (1 least – 10 most)	Release Schedule (Sprint 1 or 2)
1	Add Customer Records		7	1
2	View Customer Records	2	7	1
3	Edit Customer Records	2	7	1
4	Delete Customer Records	2	7	1
5	Add Television Records		7	1
6	View Television Records	5	7	1
7	Edit Television Records	5	7	1
8	Delete Television Records	5	7	1
9	Add Sales Records		8	2
10	View Sales Records (all)	9	8	2
11	Edit Sales Records	9	8	2
12	Delete Sales Records	9	8	2
13	Add Admin Records		7	2
14	View Admin Records	13	7	2
15	Edit Admin Records	13	7	2
16	Delete Admin Records	13	7	2
17	Authenticate Admin Records	13	10	2

Software Quality:

In order to maintain the quality of this software, I have decided to use the following from ISO 25010 model: *(these will be later used in testing relevant functionalities to ensure that desired quality is maintained throughout the software)*

- Functional suitability: (Functional Completeness). To achieve completeness, the set of functions specified in backlog features (No-1 to 17) 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 Television employee when checking the backlog items should take less than 10 seconds to load, once the software has been fully developed and deployed

Usability: (User Interface Aesthetics) The software's menu interface for accessing all the product backlog items should be simple or self-explanatory enough for the users to easily understand and use (without any further training for it). Furthermore, the user, product owner and system testers must be allowed to test the system for a prolonged period of time (like an hour or two) until they have accessed all the features mentioned in the product backlog (No-1 to 17). 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 from the product backlog items (no-1 to 17), 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 Protection.

- Security: (Confidentiality) The System is protected by username and password pairs present in the administrative or admin account and only authenticated administrative user can access it after waiting less than 10 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 10 seconds of login time.

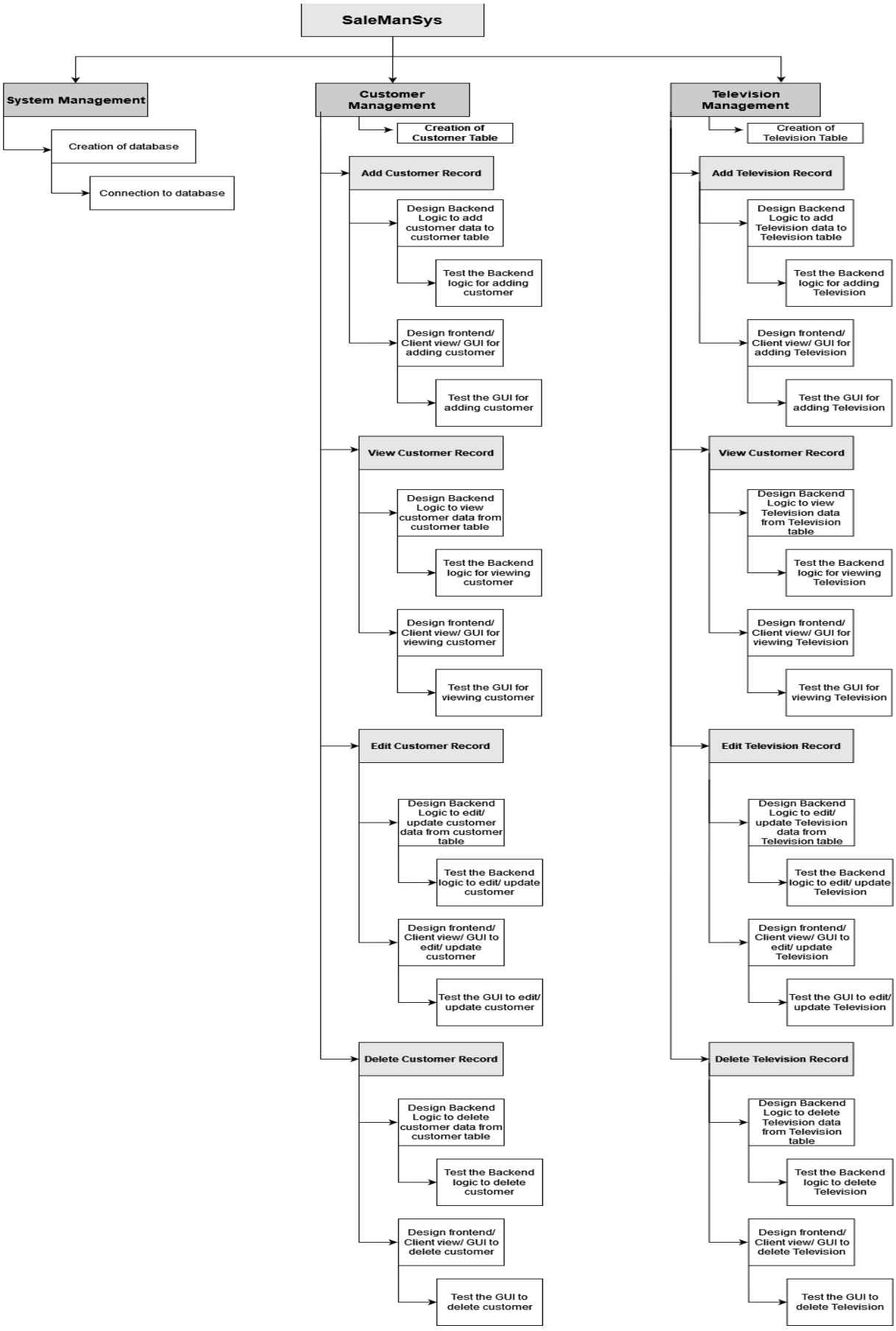
The WBS (Work Breakdown Structure) and the estimates for Sprint-1:

(ie the detailed planning for sprint 1)

The WBS:

Here I have decided to use the product based approach to ensure that no product planned to be developed in sprint 1 is missing from the task breakdown structure/chart.

The WBS tasks in list form:


SaleManSys
System Management
Creation of database
Connection to database
Customer Management
Creation of Customer Table
Add Customer Record
Design Backend Logic to add customer data to customer table
Test the Backend logic for adding customer
Design frontend/ Client view/ GUI for adding customer
Test the GUI for adding customer
View Customer Record
Design Backend Logic to view customer data from customer table
Test the Backend logic for viewing customer
Design frontend/ Client view/ GUI for viewing customer
Test the GUI for viewing customer
Edit Customer Record
Design Backend Logic to edit/ update customer data from customer table
Test the Backend logic to edit/ update customer
Design frontend/ Client view/ GUI to edit/ update customer
Test the GUI to edit/ update customer
Delete Customer Record
Design Backend Logic to delete customer data from customer table
Test the Backend logic to delete customer
Design frontend/ Client view/ GUI to delete customer
Test the GUI to delete customer
Television Management
Creation of Television Table
Add Television Record
Design Backend Logic to add Television data to Television table
Test the Backend logic for adding Television
Design frontend/ Client view/ GUI for adding Television
Test the GUI for adding Television
View Customer Record
Design Backend Logic to view Television data from Television table
Test the Backend logic for viewing Television
Design frontend/ Client view/ GUI for viewing Television
Test the GUI for viewing Television
Edit Television Record
Design Backend Logic to edit/ update Television data from Television table
Test the Backend logic to edit/ update Television
Design frontend/ Client view/ GUI to edit/ update Television
Test the GUI to edit/ update Television
Delete Television Record
Design Backend Logic to delete Television data from Television table
Test the Backend logic to delete Television
Design frontend/ Client view/ GUI to delete Television
Test the GUI to delete Television

[Note: in the diagram, the grey colour boxes represent system name and backlog item while the white boxes represent subtasks that must be fulfilled to accomplish it.]

1. Create Database
2. Create Connection to the Database
3. Create Customer Table
4. Design Backend functionality and logic to add customer data to customer table
5. Test Backend functionality and logic to add customer data to customer table
6. Design GUI for adding customer
7. Test GUI for adding customers
8. Design Backend functionality and logic to view customer data from customer table
9. Test Backend functionality and logic to view customer data to customer table
10. Design GUI for viewing customer
11. Test GUI for viewing customers
12. Design Backend functionality and logic to edit/update customer data from customer table
13. Test Backend functionality and logic to edit/update customer data to customer table
14. Design GUI for edit/update customer
15. Test GUI for edit/update customers
16. Design Backend functionality and logic to delete customer data from customer table
17. Test Backend functionality and logic to delete customer data to customer table
18. Design GUI for delete customer
19. Test GUI for delete customers
20. Create Television Table
21. Design Backend functionality and logic to add Television data to Television table
22. Test Backend functionality and logic to add Television data to Television table
23. Design GUI for adding Television
24. Test GUI for adding Television
25. Design Backend functionality and logic to view Television data from Television table
26. Test Backend functionality and logic to view Television data to Television table
27. Design GUI for viewing Television
28. Test GUI for viewing Television
29. Design Backend functionality and logic to edit/update Television data from Television table
30. Test Backend functionality and logic to edit/update Television data to Television table
31. Design GUI for edit/update Television
32. Test GUI for edit/update Television
33. Design Backend functionality and logic to delete Television data from Television table
34. Test Backend functionality and logic to delete Television data to Television table
35. Design GUI for delete Television
36. Test GUI for delete Television

Now considering Ideal Time Ideal Effort and Effective effort, we can obtain 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/ task numbers refer to their task numbers and names in the list above 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	60mins	120mins	Here I am creating the database for the entire system in C# and then verifying 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 the project is being done in C# and the different types of errors that usually pop up, some leeway time has been kept for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
2	30 mins	30mins	60mins	Here I am creating the connection to the database for the entire system in C# and then verifying 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 30mins; considering the fact that the project is being done in C# and the different types of errors that usually pop up, some leeway time has been kept for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
3	60 mins	60mins	120mins	Here I am creating the Customer table in C# and then verifying that it works by using a test data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that the project is being done in C# and the different types of errors that usually pop up, some leeway time has been kept for troubleshooting and research. Furthermore, there

				is also the fact that each person will only be effectively working 70% productively.
4	60 mins	60mins	120mins	Here I am creating the functionality and logic to add customer data to customer table in C# and then verifying that it works by using a test data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that the project is being done in C# and the different types of errors that usually pop up, some leeway time has been kept for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
5	60 mins	60mins	120mins	Here I am vigorously testing the functionality and logic to add customer data to customer table in C# until the quality mentioned in ISO25010 in Software Quality section has been achieved. Although this should ideally take 60mins; considering the fact that the project is being done in C# and the different types of errors that usually pop up, some leeway time has been kept for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
6	60 mins	60mins	120mins	Here I am creating the GUI to add customers in C# and then verifying that it works by using a test data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that the project is being done in C# and the different types of errors that usually pop up, some leeway time has been kept for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
7	60 mins	60mins	120mins	Here I am vigorously testing the GUI feature for adding the Customers on the Client side in C# until the quality mentioned in ISO25010 in

				Software Quality section has been achieved. 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.
8	60 mins	60mins	120mins	Here I am creating the functionality and logic to view customer data to customer table in C# and then verifying that it works by using a test data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that the project is being done in C# and the different types of errors that usually pop up, some leeway time has been kept for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
9	60 mins	60mins	120mins	Here I am vigorously testing the functionality and logic to view customer data to customer table in C# until the quality mentioned in ISO25010 in Software Quality section has been achieved. Although this should ideally take 60mins; considering the fact that the project is being done in C# and the different types of errors that usually pop up, some leeway time has been kept for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
10	60 mins	60mins	120mins	Here I am creating the GUI to view customers in C# and then verifying that it works by using a test data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that the project is being done in C# and the different types of errors that usually pop up, some leeway time has been kept for

				troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
11	60 mins	60mins	120mins	Here I am vigorously testing the GUI feature for viewing the Customers on the Client side in C# until the quality mentioned in ISO25010 in Software Quality section has been achieved. 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.
12	60 mins	60mins	120mins	Here I am creating the functionality and logic to edit/update customer data to customer table in C# and then verifying that it works by using a test data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that the project is being done in C# and the different types of errors that usually pop up, some leeway time has been kept for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
13	60 mins	60mins	120mins	Here I am vigorously testing the functionality and logic to edit/update customer data to customer table in C# until the quality mentioned in ISO25010 in Software Quality section has been achieved. Although this should ideally take 60mins; considering the fact that the project is being done in C# and the different types of errors that usually pop up, some leeway time has been kept for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
14	60	60mins	120mins	Here I am creating the GUI to edit/update

	mins			customers in C# and then verifying that it works by using a test data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that the project is being done in C# and the different types of errors that usually pop up, some leeway time has been kept for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
15	60 mins	60mins	120mins	Here I am vigorously testing the GUI feature to edit/update the Customers on the Client side in C# until the quality mentioned in ISO25010 in Software Quality section has been achieved. 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.
16	60 mins	60mins	120mins	Here I am creating the functionality and logic to delete customer data to customer table in C# and then verifying that it works by using a test data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that the project is being done in C# and the different types of errors that usually pop up, some leeway time has been kept for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
17	60 mins	60mins	120mins	Here I am vigorously testing the functionality and logic to delete customer data to customer table in C# until the quality mentioned in ISO25010 in Software Quality section has been achieved. Although this should ideally take 60mins; considering the fact that the project is being done

				in C# and the different types of errors that usually pop up, some leeway time has been kept for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
18	60 mins	60mins	120mins	Here I am creating the GUI to delete customers in C# and then verifying that it works by using a test data (this will be removed later on after troubleshooting is over). Although this should ideally take 60mins; considering the fact that the project is being done in C# and the different types of errors that usually pop up, some leeway time has been kept for troubleshooting and research. Furthermore, there is also the fact that each person will only be effectively working 70% productively.
19	60 mins	60mins	120mins	Here I am vigorously testing the GUI feature to delete the Customers on the Client side in C# until the quality mentioned in ISO25010 in Software Quality section has been achieved. 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.
20	30 mins	30mins	60mins	Here I am creating the Television table in C# and then verifying that it works by using a test data (this will be removed later on after troubleshooting is over). **[Since Similar task has been done in task no 3, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]**
21	30 mins	30mins	60mins	Here I am creating the functionality and logic to add Television data to Television table in C# and then verifying that it works by using a test data

				(this will be removed later on after troubleshooting is over). **[Since Similar task has been done in task no 4, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]**
22	30 mins	30mins	60mins	Here I am vigorously testing the functionality and logic to add Television data to Television table in C# until the quality mentioned in ISO25010 in Software Quality section has been achieved. **[Since Similar task has been done in task no 5, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]**
23	30 mins	30mins	60mins	Here I am creating the GUI to add Television in C# and then verifying that it works by using a test data (this will be removed later on after troubleshooting is over). **[Since Similar task has been done in task no 6, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]**
24	30 mins	30mins	60mins	Here I am vigorously testing the GUI feature for adding the Television on the Client side in C# until the quality mentioned in ISO25010 in Software Quality section has been achieved. **[Since Similar task has been done in task no 7, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]**
25	30 mins	30mins	60mins	Here I am creating the functionality and logic to view Television data to Television table in C# and then verifying that it works by using a test data (this will be removed later on after

				troubleshooting is over). **[Since Similar task has been done in task no 8, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]**
26	30 mins	30mins	60mins	Here I am vigorously testing the functionality and logic to view Television data to Television table in C# until the quality mentioned in ISO25010 in Software Quality section has been achieved. **[Since Similar task has been done in task no 9, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]**
27	30 mins	30mins	60mins	Here I am creating the GUI to view Television in C# and then verifying that it works by using a test data (this will be removed later on after troubleshooting is over). **[Since Similar task has been done in task no 10, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]**
28	30 mins	30mins	60mins	Here I am vigorously testing the GUI feature for viewing the Television on the Client side in C# until the quality mentioned in ISO25010 in Software Quality section has been achieved. **[Since Similar task has been done in task no 11, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]**
29	30 mins	30mins	60mins	Here I am creating the functionality and logic to edit/update Television data to Television table in C# and then verifying that it works by using a test data (this will be removed later on after troubleshooting is over).

				[Since Similar task has been done in task no 12, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]
30	30 mins	30mins	60mins	Here I am vigorously testing the functionality and logic to edit/update Television data to Television table in C# until the quality mentioned in ISO25010 in Software Quality section has been achieved. **[Since Similar task has been done in task no 13, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]**
31	30 mins	30mins	60mins	Here I am creating the GUI to edit/update Television in C# and then verifying that it works by using a test data (this will be removed later on after troubleshooting is over). **[Since Similar task has been done in task no 14, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]**
32	30 mins	30mins	60mins	Here I am vigorously testing the GUI feature to edit/update the Television on the Client side in C# until the quality mentioned in ISO25010 in Software Quality section has been achieved. **[Since Similar task has been done in task no 15, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]**
33	30 mins	30mins	60mins	Here I am creating the functionality and logic to delete Television data to Television table in C# and then verifying that it works by using a test data (this will be removed later on after troubleshooting is over).

				[Since Similar task has been done in task no 16, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]
34	30 mins	30mins	60mins	Here I am vigorously testing the functionality and logic to delete Television data to Television table in C# until the quality mentioned in ISO25010 in Software Quality section has been achieved. **[Since Similar task has been done in task no 17, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]**
35	30 mins	30mins	60mins	Here I am creating the GUI to delete Television in C# and then verifying that it works by using a test data (this will be removed later on after troubleshooting is over). **[Since Similar task has been done in task no 18, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]**
36	30 mins	30mins	60mins	Here I am vigorously testing the GUI feature to delete the Television on the Client side in C# until the quality mentioned in ISO25010 in Software Quality section has been achieved. **[Since Similar task has been done in task no 19, I am assuming it will take the half the time as that task to accomplish this and thus have given it as 30 mins for ideal time instead of the usual 60]**

Total Effective Effort: **54 hrs** (=3240 mins).

Now I have decided to follow the pair programming for the following reasons:

a) Following the information in the lecture about “trunk factor” in risk, it is better 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.

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

Thus considering the effect of the pair programming, we get **108 hrs** (=2* total effective effort).

Now a **Team of 6 people** where each is working a **2-week sprint, 8 hrs per week** would normally need to have a total of **96hrs** (=6*2*8) of work load per sprint.

Although effective effort exceed the team’s total time by **12 hrs**, this time has been calculated by taking extrapolation of times taken on the task (ie extra research and troubleshooting times) and being on the assumption that there will always be a different way to code the same object.. But that is not the case in reality as once a format gets produced and agreed upon for a reusable functionality, people tend to continue reusing that instead of “reinventing the wheel” (so as to drastically save time and also ensure that the functionality is following the accepted standards). Thus it is more likely that the timeline would instead act as a guideline while the tasks would end up being competed far before that in reality. So I believe the workload for sprint 1 is adequate.