

# Individual Contribution and Reflective Report

**Name:** Dikshyanta Uprety

**Group:** Group 1

## Contribution

In this section, I will write about my contribution to the project. Each contribution will be assigned a number [x] and the evidence regarding that contribution is listed in a table below with the assigned number (x). The evidence is followed by reflection on the contribution.

In the designing phase (part 1) of the project, I was involved in designing the use case diagram [1], the class diagram [2] and one of the sequence diagrams [3]. The use case diagram and class diagram were designed by the entire team in the practical classes during the first semester. One sequence diagram was individually designed by every member in the team.

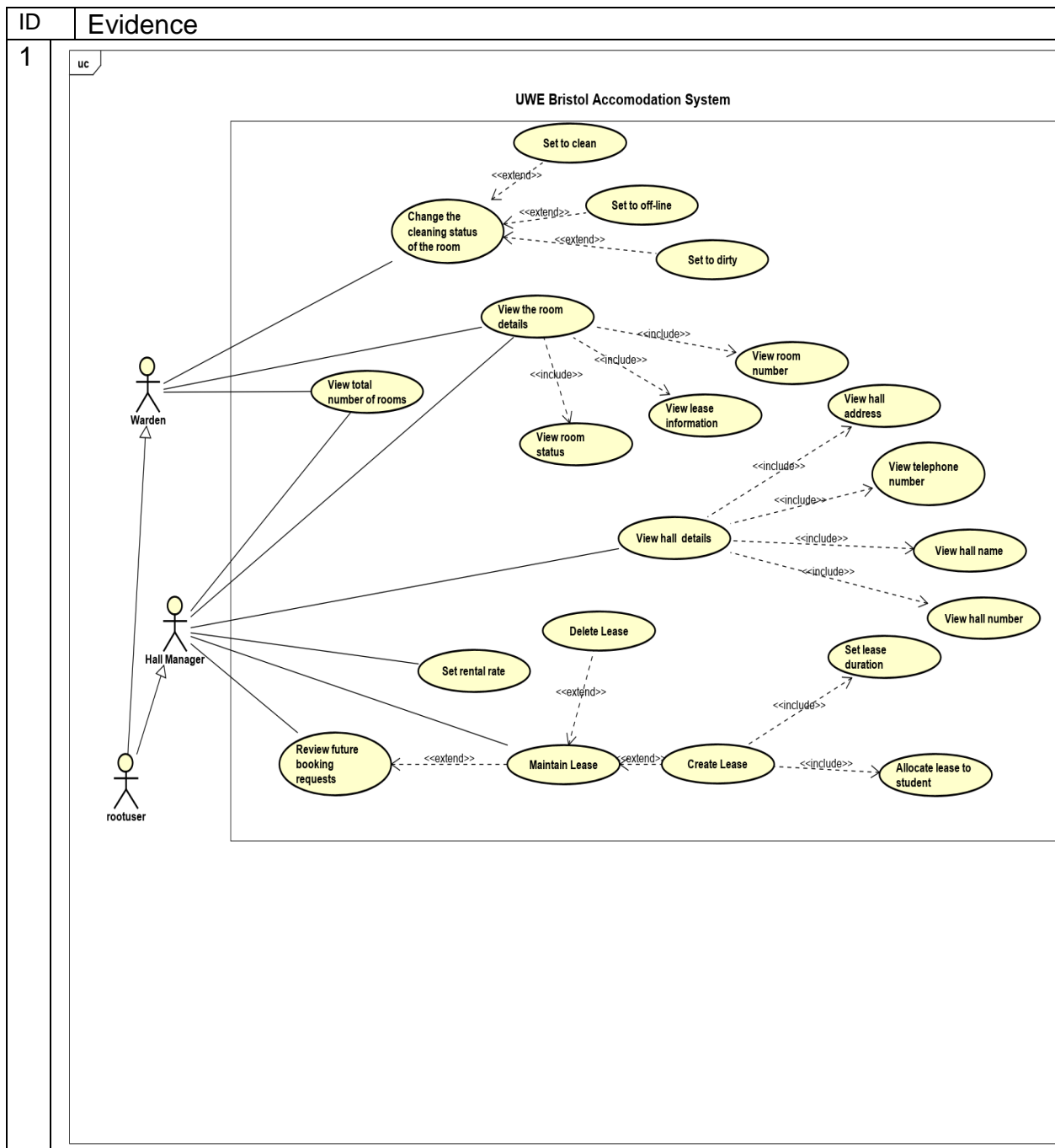
In the system implementation phase (part 2) of the project, I was involved in the input validation for GUI [4], testing [5] and group report writing [6]. The input validation was done for the different inputs taken by the GUI, such as student name, lease number, etc. Each of these attributes should be inputted with proper values for the system to be robust and usable. All the different rules for various inputs are added as methods in a file called InputValidator.java, which can be found in the same directory as the GUI. These methods are invoked to check different input values throughout the system.

A total of 20 test cases were tested against the designed system. This includes both test cases for all use cases in the use case diagram and test cases to check the input validation function. Each test case is defined by ID, Purpose, Expected Result, Actual Result, Status and Evidence. The test cases are arranged in a table with respect to their IDs. Another table in the same document displays the photographic evidence of each test and is referenced in the evidence column of the main table.

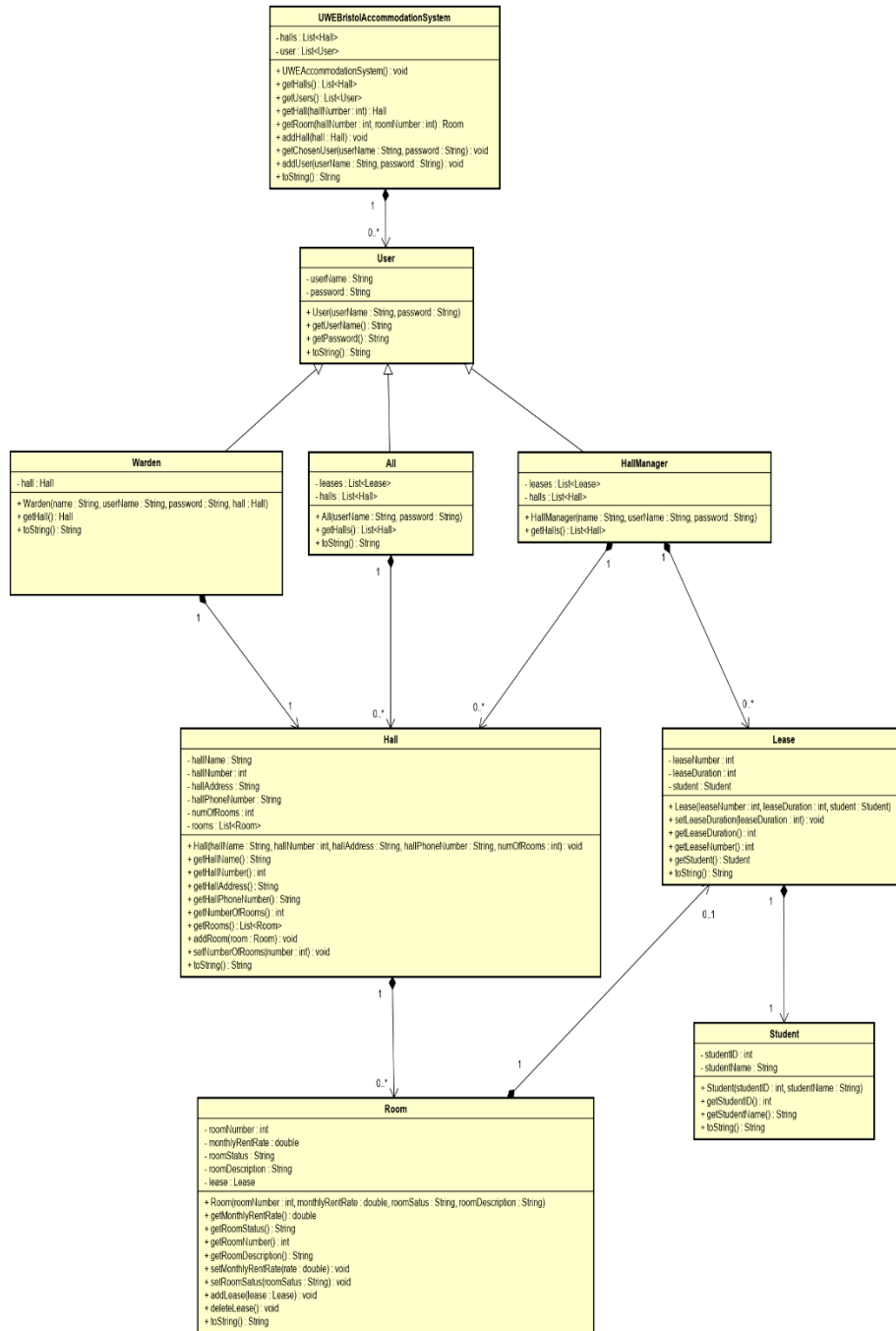
The group report is divided into five parts. They are strategy planning, continuous team iteration, team communication, simplicity and work plan. From these five parts, I wrote the contents for strategy planning, continuous team iteration and team communication sections.

## Evidence

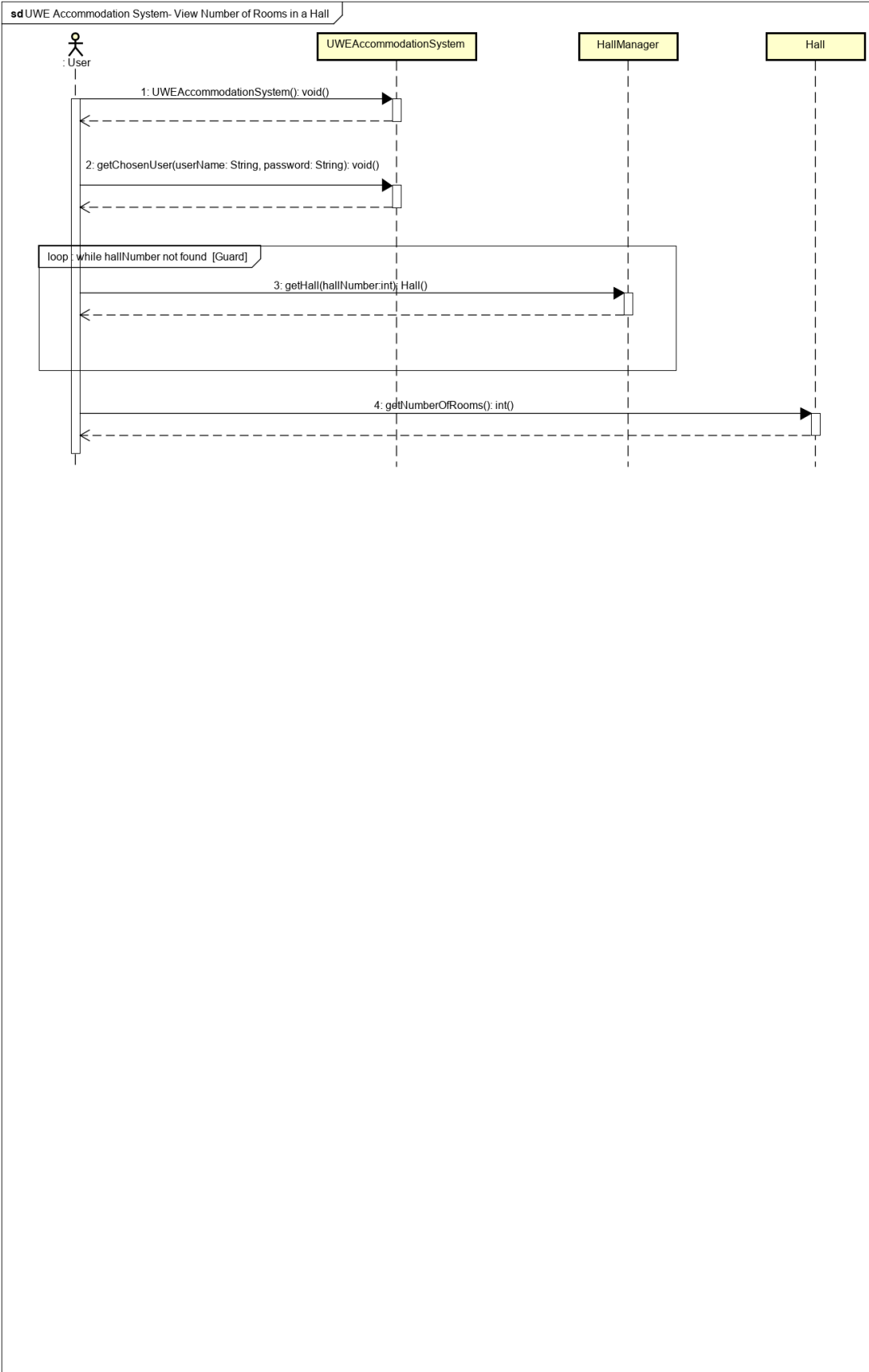
The table below lists the evidence/documents alongside their assigned IDs in my contribution log:



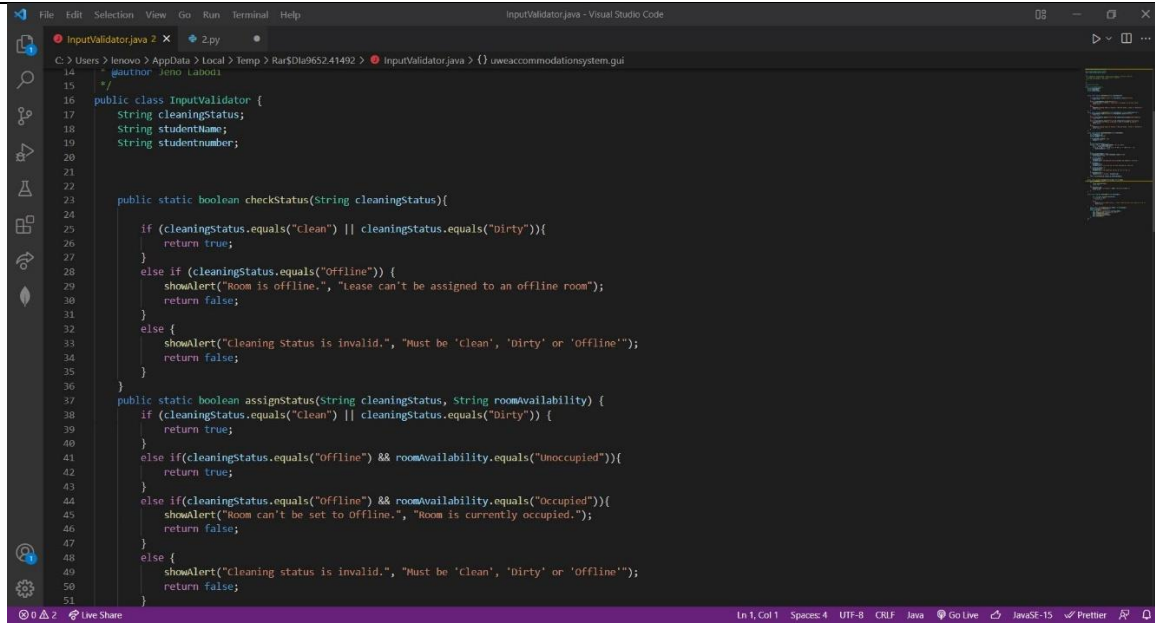
UWE Bristol Accommodation System - Class diagram



3



4



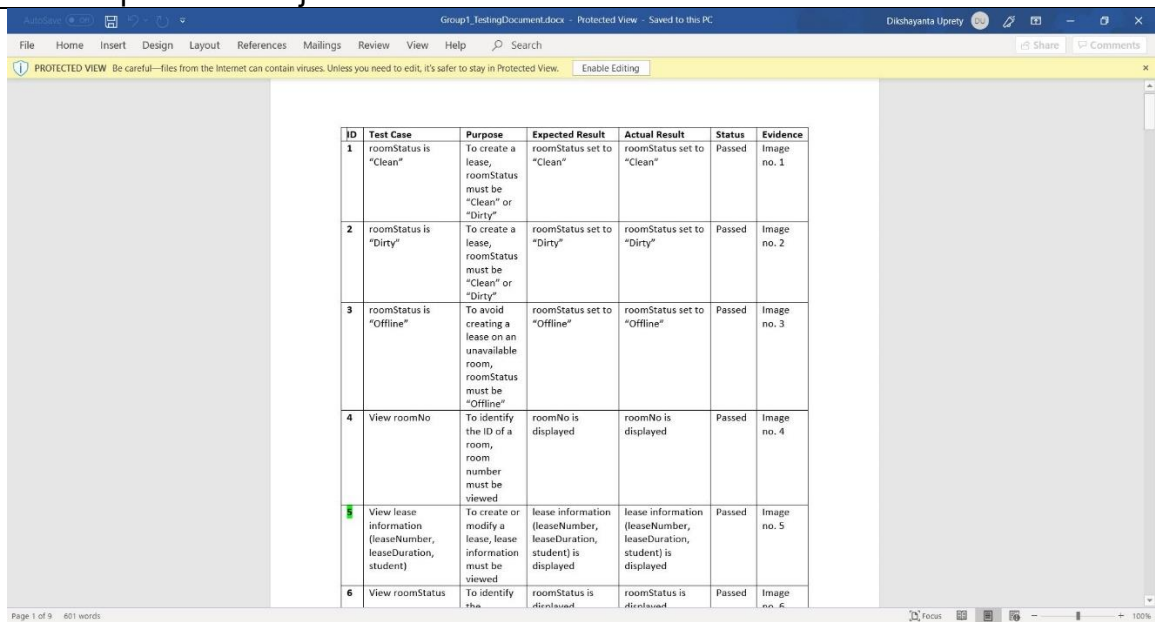
```

14  *
15  */
16  public class InputValidator {
17      String cleaningStatus;
18      String studentName;
19      String studentNumber;
20
21
22
23
24      public static boolean checkStatus(String cleaningStatus){
25
26          if (cleaningStatus.equals("Clean") || cleaningStatus.equals("Dirty")){
27              return true;
28          }
29          else if (cleaningStatus.equals("Offline")) {
30              showAlert("Room is offline.", "Lease can't be assigned to an offline room");
31              return false;
32          }
33          else {
34              showAlert("Cleaning Status is invalid.", "Must be 'Clean', 'Dirty' or 'Offline'");
35              return false;
36          }
37      }
38      public static boolean assignStatus(String cleaningStatus, String roomAvailability) {
39          if (cleaningStatus.equals("Clean") || cleaningStatus.equals("Dirty")) {
40              return true;
41          }
42          else if(cleaningStatus.equals("Offline") && roomAvailability.equals("Unoccupied")){
43              return true;
44          }
45          else if(cleaningStatus.equals("Offline") && roomAvailability.equals("Occupied")){
46              showAlert("Room can't be set to Offline.", "Room is currently occupied.");
47              return false;
48          }
49          else {
50              showAlert("Cleaning status is invalid.", "Must be 'Clean', 'Dirty' or 'Offline'");
51              return false;
52          }
53      }
54  }

```

File : 'InputValidator.java'

5



ID	Test Case	Purpose	Expected Result	Actual Result	Status	Evidence
1	roomStatus is "Clean"	To create a lease, roomStatus must be "Clean" or "Dirty"	roomStatus set to "Clean"	roomStatus set to "Clean"	Passed	Image no. 1
2	roomStatus is "Dirty"	To create a lease, roomStatus must be "Clean" or "Dirty"	roomStatus set to "Dirty"	roomStatus set to "Dirty"	Passed	Image no. 2
3	roomStatus is "Offline"	To avoid creating a lease on an unavailable room, roomStatus must be "Offline"	roomStatus set to "Offline"	roomStatus set to "Offline"	Passed	Image no. 3
4	View roomNo	To identify the ID of a room, room number must be viewed	roomNo is displayed	roomNo is displayed	Passed	Image no. 4
5	View lease information (leaseNumber, leaseDuration, student)	To create or modify a lease, lease information must be viewed	lease information (leaseNumber, leaseDuration, student) is displayed	lease information (leaseNumber, leaseDuration, student) is displayed	Passed	Image no. 5
6	View roomStatus	To identify the	roomStatus is displayed	roomStatus is displayed	Passed	Image no. 6

File: 'Test.docx'

6	<div data-bbox="760 201 899 222">Agile report – Group 1</div> <div data-bbox="592 231 675 245"> <b>Strategy planning</b> </div> <div data-bbox="592 252 1065 342"> <p>A major part of strategic planning has already been achieved in the first part of the project. The diagrams produced during this phase will guide the team throughout the process of system implementation and GUI design. Firstly, the use case diagram contains different users of the system and their use cases (user needs). It gives the gist of the system and what it should be capable of doing, as the use cases of these users must be fulfilled for the system to be functional. This allows the team to identify the tasks they need to do and divide them equally among the members.</p> </div> <div data-bbox="592 348 1052 422"> <p>Secondly, the class diagram, also designed during the first part of the project, provides a static view of all the classes necessary in the system. The diagram identifies all the attributes and methods in each class for the system to be able to provide functionalities to satisfy the user needs described in the use case diagram. This diagram will act as the blueprint of our system and will be used to design all the classes and their methods and attributes.</p> </div> <div data-bbox="592 428 1062 504"> <p>While the class diagram provides a static overview of the classes, the three sequence diagrams designed by the team show how these classes interact with each other using their methods and attributes, to produce a particular functionality and satisfy a use case described in the use case diagram. This diagram will assist the team in designing the GUI as it displays how the methods in the classes will be used to produce dynamic results in the working system.</p> </div> <div data-bbox="592 510 1065 613"> <p>Besides from these diagrams, the tasks during this phase will be divided in the following manner. One of the members of the team will be in charge of the GUI (Connor), another will take charge of testing (Ileno) and the other will take charge of the report (Dikshyanta). Each of them will be responsible for their assigned parts and would work with their teammates iteratively to progress that aspect. Conversely, they will also work with their teammates in the other two aspects that they will not be in charge of. The module practical every week will be used to convey updates and queries and work together. A WhatsApp group would supplement by allowing online meetings and text exchange.</p> </div> <div data-bbox="592 619 716 634"> <b>Continuous team iteration</b> </div> <div data-bbox="592 640 1065 724"> <p>The team will use an agile approach to the project and will keep it flexible and iterative between planning and implementation. This is because the team understands that while designing the system, the team might come across an obstacle which might require the team to tweak the design without losing functionality. A sprint cycle will be used by the team. The entire project is divided into twenty sprint periods and each sprint period contains a small portion of work to be done. The whole sprint cycle is listed in the Work Plan section of this document. During a sprint period, the team will work together...</p> </div> <div data-bbox="256 762 587 795"> File: 'group report.docx' </div>
---	--

## Reflection

This part contains the skills and knowledge I acquired through each contribution made to this project.

In the first part of the project, while producing the three UML diagrams (use case, class, sequence), I learned the appropriate syntax for each diagram. For example, an arrow from one user to another in a use case diagram is used to show a parent child relation in between the users.

Secondly, I learned the advantages and differences between the diagrams and how each can aid us during software development process. For example, use case diagram can help the team to better understand the users and their needs whereas class diagram provides the blueprint of classes, attributes and methods. Both of them are useful, but for very different purposes.

More importantly, by using these diagrams for my assignment, I learned the importance of adhering to a standard notation. Because UML is very widely accepted modelling language, if we encounter a problem while reading or producing a UML diagram, we can make use of the numerous resources available to us (books, internet, etc.) to find a solution. However, if a programmer uses his own symbols and does not follow a standard model then it will be difficult for other programmers to understand his diagram. That is because the diagram is not standard, and a symbol could mean a lot of things.

As a result, people might not be able to interpret or perhaps misinterpret the diagram. Therefore, I learned that we should always use a standard like UML for making models.

In the second part, (input validation and testing), I learned the importance of designing a system to be as robust as possible. Before we validated input and tested it, the user could enter a number as the student name. A computer may accept it but in practice, we do not expect numbers as valid names. Therefore, that piece of data is useless and might cause a problem later on in the system. Therefore, in order to get only usable authentic values from the user, the input values must be tested and validated.