Detailed Use Cases

Use Case No: 1	Use Case Name: Take Quiz	Rating: Must
Purpose: School pupil taking the quiz.		
Main Actor: Student	Secondary Actors:	
Pre-conditions: Quiz must have been setup		
Trigger: Pressing the Start Button to start the quiz		
Basic Flow: 1. **A question is displayed to the student.** This use case starts when the student has pressed the button that starts the quiz. 2. **Select answer.** System displays four choices for the question shown. Student selects a single answer. 3. **Continue to next question.** Immediate feedback is given to the student informing them of their answer was correct or incorrect. Student presses the "continue" button and moves on to the next question. 4. **Feedback summary.** When the quiz is finished, system displays a full summary of results to the student. Student presses the "finish" button and is returned to the main menu indicating that they have completed the quiz. Alternative Flow: **Presses "Quit" button.** In the basic flow, excluding the feedback summary, if the student presses "Quit" button whilst on a question, system returns them to the main menu. **Presses "Restart" button.** In the basic flow, excluding the feedback summary, if the student presses "Restart" button whilst on a question, system begins a brand-new quiz with new questions to be answered.		
Extension:		
Related Use Cases:		
Post-conditions: System stores the results for the engagement team.		
Author:	Date:	Approved:	Date:

Use Case No: 2	Use Case Name: Managing the quiz	Rating: Must
Purpose: Managing the quiz.		
Main Actor: Engagement team	Secondary Actors:	
Pre-conditions: Correct password is entered		
Trigger: Setup button is pressed		

Basic Flow:

1. **Enter the school year**
   When the setup screen is shown, a school year field should be filled with the corresponding school year attending the event.
2. **Enter the schools attending.**
   Once a school year has been inputted, a new field is shown that contains the names of the schools attending the event, select the school which the students that are taking part in the quiz are from.
3. **Selecting topic**
   A list of stored topics in the system are displayed and the topic corresponding to the current event should be chosen.
4. **Saving**
   After all information is entered, the save button can be pressed to store the information and the user is returned to the main menu. Once done, the quiz has been successfully set up for the use of the following participants.

Alternative Flow:

Wrong Password Entered

User enters a wrong password in the password field which in turn displays him a message informing them that access is denied and brings them back to the Main Menu.

Add Question

User presses “Add Question” button below the selected topic. System displays a question field, user enters a question. System displays a field to enter the correct answer and three more fields to enter incorrect answers. User presses the “Save” button to store the question and add it to the pool of questions.

Amend question

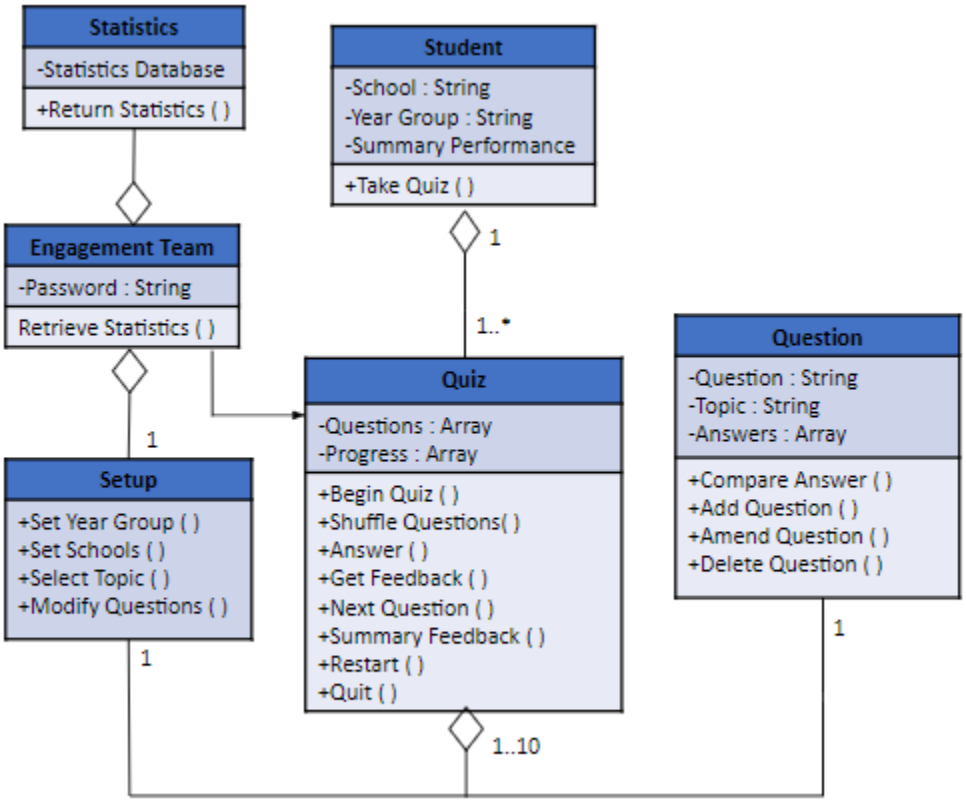
User presses the “Amend Question” button, system displays a list of all questions for the selected topic. User chooses a question, system displays the Question field, the Answer field and the three incorrect answer fields which are open to edit. After changes have been made a “Save” button can be pressed to save the new question information.

Delete question

User chooses questions from a list which they want to delete by checking the box next it. After the user makes a selection of questions to delete, they then click on the delete button. The change is then confirmed through a pop-up message, where the user has the option to confirm or cancel. If they wish they can amend their selection of

questions to delete or they cancel the operation all together.			
Extension:			
Related Use Cases:			
Post-conditions: The quiz has been updated, ready for the use of the next students			
Author:	Date:	Approved:	Date:

UML Class Diagram



Description

Classes shown in the diagram have individual attributes and methods. Class attributes specify characteristics of the class. Methods are actions that the class can perform. For a system to be complete all of its functional requirements must be implemented within the classes. There are main and alternative flows in the system, and they provide different pathways for the user.

Statistics Class

- The “Statistics Database” attribute will have stored all of the statistics for the questions answered while the students were taking the quiz. It can store the time taken to answer each question and the percentage of correctly answered.
- The “Return Statistics ( )” method will make it plausible for the engagement team to access the statistics and then modify questions according to their result making the quiz more balanced.

Engagement Team Class

- The attributes the Engagement Team has is the “password”, which will be used once a member of the Team presses the Setup button and enters the correct password bringing them to the setup menu.
- A method this class contains is the “Retrieve Statistics ( )” which will in turn get statistics after each completion of a Quiz.

Setup Class

For the setting up of the Quiz which will be done by the Engagement Team some functions need to be performed in order for the Quiz to be ready to used.

- The “Set Year Group ( )” method will set the current Year group that will attend the event that that Time giving them questions in their appropriate difficulty.
- “Set Schools ( )” method will store the schools that will be present at the event.
- The “Select Topic ( )” method will be used to select the topic for the event presenting to the Student taking the Quiz questions according to the occasion.
- “Modify Questions ( )” method leads this function to its connected class “Questions” which the Engagement Team will be able to Add questions, amend questions or delete questions of the Event Topic.

Questions Class

The Quiz will contain 1 up to 10 questions each time it is run, these questions will have specific attributes and methods connected to them.

- In this class the “Questions Database” attribute will have all of the questions stored where the quiz can access them to present them to the student.
- The “Topic” attribute will be set to filter the questions appropriate for the event.
- Corresponding to each question an “Answer” attribute is used.
- Each question must have the other 3 wrong choices that the student is able to press, and this is entered with the “Other Choices” attribute.

- A method used in this class will be “Compare Answer ( )” which will compare the answer given by the student from the “Answer ( )” method in the “Quiz” class, with the one stored for the Question shown giving an either correct or wrong result.
- While the “Modify Questions ( )” method is used in the “Setup” class, these different 3 methods (“Add Question ( )”, “Amend Question ( )” “Delete Question ( )”) can be in turn be chosen to do specific actions to the questions stored.

Quiz Class

- A “Questions” attribute will have the questions used for the quiz stored so they can be accessed.
- In addition a “Progress” attribute will have the progress of the Student taking the quiz stored so a final result can be produced after they complete the quiz.
- The “Begin Quiz ( )” method is used to start the quiz by presenting the first question to the student.
- The “Shuffle Questions ( )” method will shuffle the questions stored in the attribute and present different ones each time a quiz is run.
- When the student submits an answer the “Answer( )” method is used which will go to the “Compare Answer ( )” method in the “Answers” class that will find if the correct answer was entered.
- After an answer has been compared the “Get Feedback ( )” method will be run to present the student his feedback for the answer given to the question shown.
- When the student reviews the feedback and presses the continue button the “Next Question ( )” method is used to generate a following question that will be shown to the student to answer.
- After the completion of the quiz the method ““Summary Feedback ( )” will be used to show the Student their overall result giving them a mark for their effort and a review for each question.
- During the quiz 2 other methods can be used, the “Restart ( )” method and the “Quit ( )” method that will in turn either restart the whole quiz starting from the 1st question or quit back to the main menu.

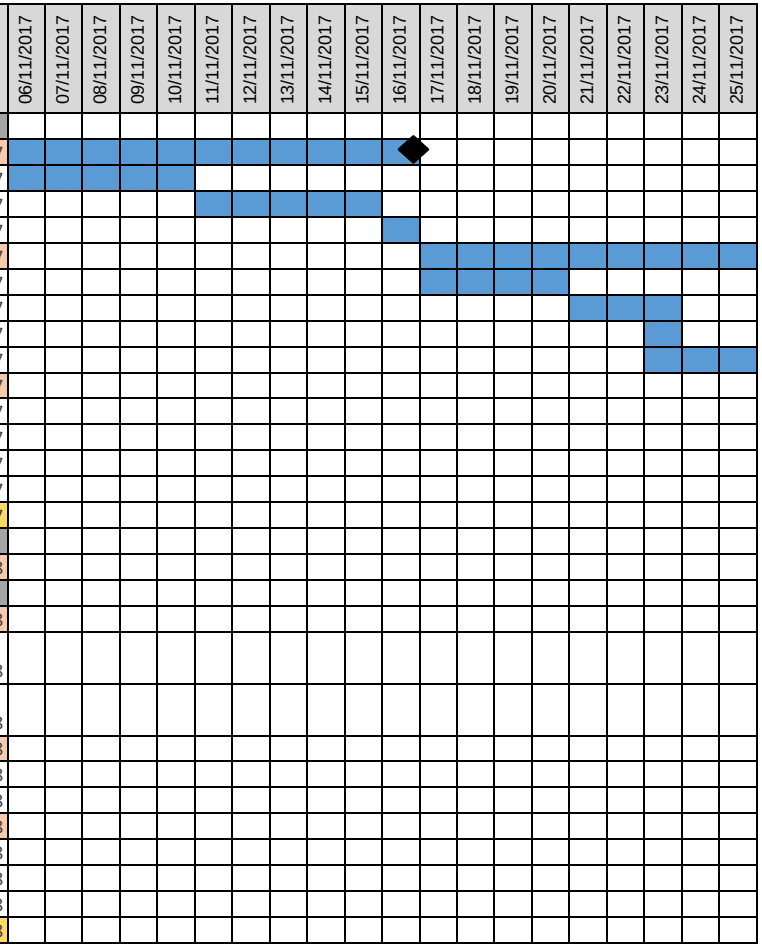
Student Class

- The “School” attribute will have the school of the student stored.
- In addition, the “Year Group” attribute will store the year group of the student.
- The “Summary Performance” attribute will store the performance of the student in the class.
- The main method that is used is “Take Quiz ( )” which the student will use to begin the whole procedure of taking the quiz.

Gantt Chart

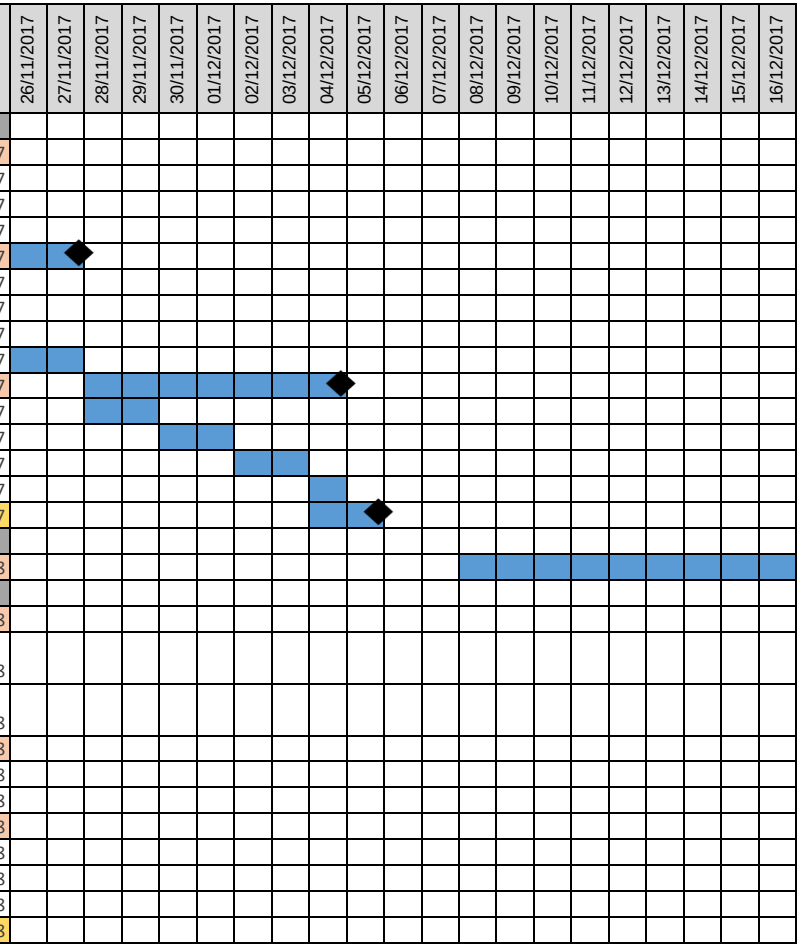
Project Name	Project Duration	Project start date	Project end date
Quiz - Team 4	106	6/11/2017	20/2/2018

Task ID	Task Description	Assigned to	Task Duration	Start Date	End Date
1	**Design Phase 1**				
1.1	Functional Requirements		11	6/11/2017	16/11/2017
1.1.1	Draft functional requirements	Ryan Harris	5	6/11/2017	10/11/2017
1.1.2	Acceptance Criteria for each requirement	Ryan Harris	5	11/11/2017	15/11/2017
1.1.3	Catergorize each requirement into MoSCoW notation	Whole Team	1	16/11/2017	16/11/2017
1.2	Non-Functional Requirements		13	17/11/2017	27/11/2017
1.2.1	Draft Non-functional requirements	Rayner Da Cruz	4	17/11/2017	20/11/2017
1.2.2	Acceptance Criteria for each requirement	Rayner Da Cruz	3	21/11/2017	23/11/2017
1.2.3	Catergorize each requirement into MoSCoW notation	Whole Team	1	23/11/2017	23/11/2017
1.3	Use Case Diagram	Spyros Lontos	5	23/11/2017	27/11/2017
1.4	Use Case Descriptions:		7	28/11/2017	4/12/2017
1.4.1	Setup Quiz	Reem Alotaibi	2	28/11/2017	29/11/2017
1.4.2	Take Quiz	Alfie Potter	2	30/11/2017	1/12/2017
1.4.3	Obtain Statistics	Adriano Sole	2	2/12/2017	3/12/2017
1.4.4	Modify Questions	Spyros Lontos	1	4/12/2017	4/12/2017
1.5	Milestone: Review Before submission	Whole Team	2	4/12/2017	5/12/2017
	Christmas Break and Exam Period				
	Christmas Break and Exam Period		50	8/12/2017	26/1/2018
2	**Design Phase 2**				
2.1	Detailed Use Case Descriptions		6	29/1/2018	3/2/2018
2.1.1	Detailed Use Case 1 - Basic & Alternate flow, Pre & Post conditions	Whole Team	3	29/1/2018	31/1/2018
2.1.2	Detailed Use Case 2 - Basic & Alternate flow, Pre & Post conditions	Whole Team	3	1/2/2018	3/2/2018
2.2	UML Class Diagram		5	4/2/2018	13/2/2018
2.2.1	Design UML Class Diagram using the Use Cases as the basis	Spyros Lontos	5	4/2/2018	8/2/2018
2.2.1	Class Diagram descriptions	Spyros Lontos	5	9/2/2018	13/2/2018
2.3	Project Plan		1	14/2/2018	16/2/2018
2.3.1	Design gantt chart with formulas	Rayner Da Cruz	1	14/2/2018	14/2/2018
2.3.2	Fill in dates and timings for the project	Adriano Sole	2	15/2/2018	16/2/2018
2.4	Risk Assessment	Alfie Potter	4	15/2/2018	18/2/2018
2.5	Milestone: Review Before submission	Whole Team	2	19/2/2018	20/2/2018



Project Name	Project Duration	Project start date	Project end date
Quiz - Team 4	106	6/11/2017	20/2/2018

Task ID	Task Description	Assigned to	Task Duration	Start Date	End Date
1	**Design Phase 1**				
1.1	Functional Requirements		11	6/11/2017	16/11/2017
1.1.1	Draft functional requirements	Ryan Harris	5	6/11/2017	10/11/2017
1.1.2	Acceptance Criteria for each requirement	Ryan Harris	5	11/11/2017	15/11/2017
1.1.3	Catergorize each requirement into MoSCoW notation	Whole Team	1	16/11/2017	16/11/2017
1.2	Non-Functional Requirements		13	17/11/2017	27/11/2017
1.2.1	Draft Non-functional requirements	Rayner Da Cruz	4	17/11/2017	20/11/2017
1.2.2	Acceptance Criteria for each requirement	Rayner Da Cruz	3	21/11/2017	23/11/2017
1.2.3	Catergorize each requirement into MoSCoW notation	Whole Team	1	23/11/2017	23/11/2017
1.3	Use Case Diagram	Spyros Lontos	5	23/11/2017	27/11/2017
1.4	Use Case Descriptions:		7	28/11/2017	4/12/2017
1.4.1	Setup Quiz	Reem Alotaibi	2	28/11/2017	29/11/2017
1.4.2	Take Quiz	Alfie Potter	2	30/11/2017	1/12/2017
1.4.3	Obtain Statistics	Adriano Sole	2	2/12/2017	3/12/2017
1.4.4	Modify Questions	Spyros Lontos	1	4/12/2017	4/12/2017
1.5	Milestone: Review Before submission	Whole Team	2	4/12/2017	5/12/2017
	Christmas Break and Exam Period				
	Christmas Break and Exam Period		50	8/12/2017	26/1/2018
2	**Design Phase 2**				
2.1	Detailed Use Case Descriptions		6	29/1/2018	3/2/2018
2.1.1	Detailed Use Case 1 - Basic & Alternate flow, Pre & Post conditions	Whole Team	3	29/1/2018	31/1/2018
2.1.2	Detailed Use Case 2 - Basic & Alternate flow, Pre & Post conditions	Whole Team	3	1/2/2018	3/2/2018
2.2	UML Class Diagram		5	4/2/2018	13/2/2018
2.2.1	Design UML Class Diagram using the Use Cases as the basis	Spyros Lontos	5	4/2/2018	8/2/2018
2.2.1	Class Diagram descriptions	Spyros Lontos	5	9/2/2018	13/2/2018
2.3	Project Plan		1	14/2/2018	16/2/2018
2.3.1	Design gantt chart with formulas	Rayner Da Cruz	1	14/2/2018	14/2/2018
2.3.2	Fill in dates and timings for the project	Adriano Sole	2	15/2/2018	16/2/2018
2.4	Risk Assessment	Alfie Potter	4	15/2/2018	18/2/2018
2.5	Milestone: Review Before submission	Whole Team	2	19/2/2018	20/2/2018



Project Name	Project Duration	Project start date	Project end date
Quiz - Team 4	106	6/11/2017	20/2/2018

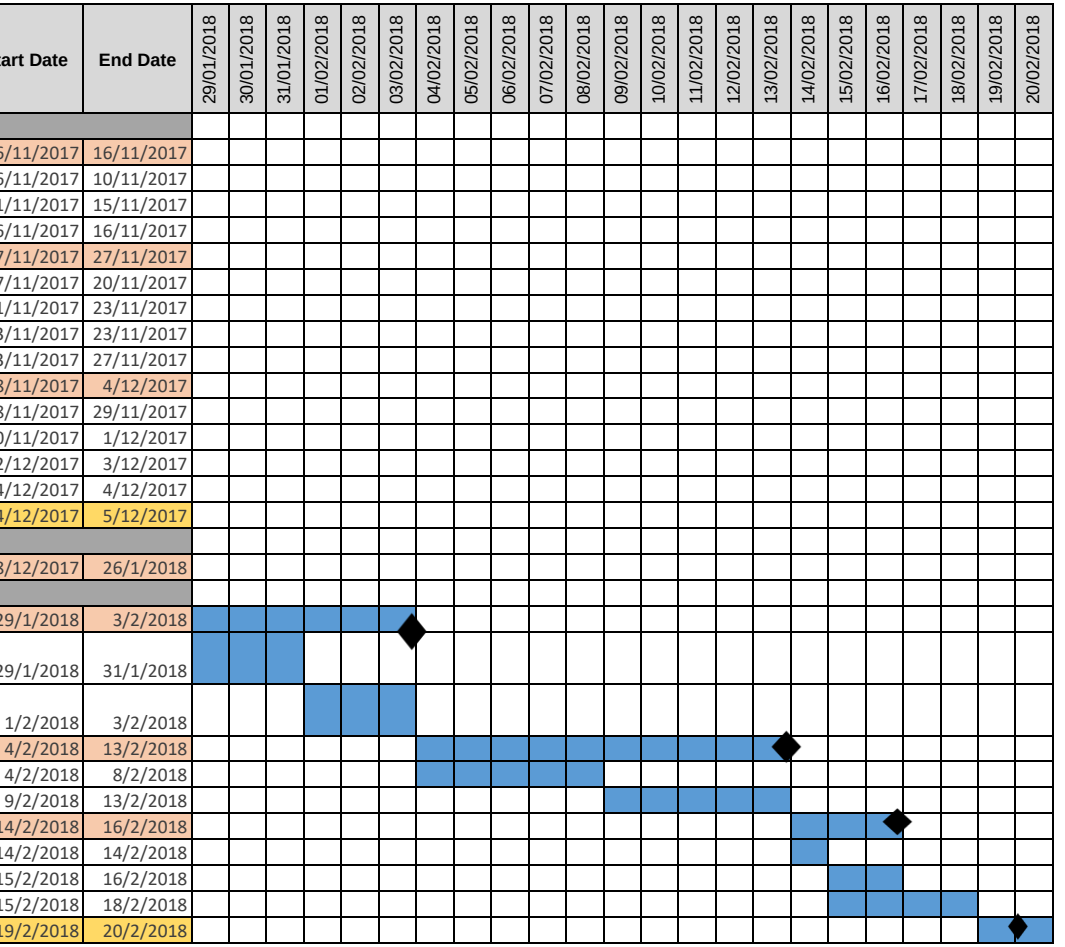
Task ID	Task Description	Assigned to	Task Duration	Start Date	End Date	17/12/2017	18/12/2017	19/12/2017	20/12/2017	21/12/2017	22/12/2017	23/12/2017	24/12/2017	25/12/2017	26/12/2017	27/12/2017	28/12/2017	29/12/2017	30/12/2017	31/12/2017	01/01/2018	02/01/2018	03/01/2018	04/01/2018	05/01/2018	06/01/2018
1	**Design Phase 1**																									
1.1	Functional Requirements		11	6/11/2017	16/11/2017																					
1.1.1	Draft functional requirements	Ryan Harris	5	6/11/2017	10/11/2017																					
1.1.2	Acceptance Criteria for each requirement	Ryan Harris	5	11/11/2017	15/11/2017																					
1.1.3	Catergorize each requirement into MoSCoW notation	Whole Team	1	16/11/2017	16/11/2017																					
1.2	Non-Functional Requirements		13	17/11/2017	27/11/2017																					
1.2.1	Draft Non-functional requirements	Rayner Da Cruz	4	17/11/2017	20/11/2017																					
1.2.2	Acceptance Criteria for each requirement	Rayner Da Cruz	3	21/11/2017	23/11/2017																					
1.2.3	Catergorize each requirement into MoSCoW notation	Whole Team	1	23/11/2017	23/11/2017																					
1.3	Use Case Diagram	Spyros Lontos	5	23/11/2017	27/11/2017																					
1.4	Use Case Descriptions:		7	28/11/2017	4/12/2017																					
1.4.1	Setup Quiz	Reem Alotaibi	2	28/11/2017	29/11/2017																					
1.4.2	Take Quiz	Alfie Potter	2	30/11/2017	1/12/2017																					
1.4.3	Obtain Statistics	Adriano Sole	2	2/12/2017	3/12/2017																					
1.4.4	Modify Questions	Spyros Lontos	1	4/12/2017	4/12/2017																					
1.5	Milestone: Review Before submission	Whole Team	2	4/12/2017	5/12/2017																					
	Christmas Break and Exam Period																									
	Christmas Break and Exam Period		50	8/12/2017	26/1/2018																					
2	**Design Phase 2**																									
2.1	Detailed Use Case Descriptions		6	29/1/2018	3/2/2018																					
2.1.1	Detailed Use Case 1 - Basic & Alternate flow, Pre & Post conditions	Whole Team	3	29/1/2018	31/1/2018																					
2.1.2	Detailed Use Case 2 - Basic & Alternate flow, Pre & Post conditions	Whole Team	3	1/2/2018	3/2/2018																					
2.2	UML Class Diagram		5	4/2/2018	13/2/2018																					
2.2.1	Design UML Class Diagram using the Use Cases as the basis	Spyros Lontos	5	4/2/2018	8/2/2018																					
2.2.1	Class Diagram descriptions	Spyros Lontos	5	9/2/2018	13/2/2018																					
2.3	Project Plan		1	14/2/2018	16/2/2018																					
2.3.1	Design gantt chart with formulas	Rayner Da Cruz	1	14/2/2018	14/2/2018																					
2.3.2	Fill in dates and timings for the project	Adriano Sole	2	15/2/2018	16/2/2018																					
2.4	Risk Assessment	Alfie Potter	4	15/2/2018	18/2/2018																					
2.5	Milestone: Review Before submission	Whole Team	2	19/2/2018	20/2/2018																					

Project Name	Project Duration	Project start date	Project end date
Quiz - Team 4	106	6/11/2017	20/2/2018

Task ID	Task Description	Assigned to	Task Duration	Start Date	End Date	07/01/2018	08/01/2018	09/01/2018	10/01/2018	11/01/2018	12/01/2018	13/01/2018	14/01/2018	15/01/2018	16/01/2018	17/01/2018	18/01/2018	19/01/2018	20/01/2018	21/01/2018	22/01/2018	23/01/2018	24/01/2018	25/01/2018	26/01/2018	27/01/2018	28/01/2018
1	**Design Phase 1**																										
1.1	Functional Requirements		11	6/11/2017	16/11/2017																						
1.1.1	Draft functional requirements	Ryan Harris	5	6/11/2017	10/11/2017																						
1.1.2	Acceptance Criteria for each requirement	Ryan Harris	5	11/11/2017	15/11/2017																						
1.1.3	Catergorize each requirement into MoSCoW notation	Whole Team	1	16/11/2017	16/11/2017																						
1.2	Non-Functional Requirements		13	17/11/2017	27/11/2017																						
1.2.1	Draft Non-functional requirements	Rayner Da Cruz	4	17/11/2017	20/11/2017																						
1.2.2	Acceptance Criteria for each requirement	Rayner Da Cruz	3	21/11/2017	23/11/2017																						
1.2.3	Catergorize each requirement into MoSCoW notation	Whole Team	1	23/11/2017	23/11/2017																						
1.3	Use Case Diagram	Spyros Lontos	5	23/11/2017	27/11/2017																						
1.4	Use Case Descriptions:		7	28/11/2017	4/12/2017																						
1.4.1	Setup Quiz	Reem Alotaibi	2	28/11/2017	29/11/2017																						
1.4.2	Take Quiz	Alfie Potter	2	30/11/2017	1/12/2017																						
1.4.3	Obtain Statistics	Adriano Sole	2	2/12/2017	3/12/2017																						
1.4.4	Modify Questions	Spyros Lontos	1	4/12/2017	4/12/2017																						
1.5	Milestone: Review Before submission	Whole Team	2	4/12/2017	5/12/2017																						
	Christmas Break and Exam Period																										
	Christmas Break and Exam Period		50	8/12/2017	26/1/2018																						
2	**Design Phase 2**																										
2.1	Detailed Use Case Descriptions		6	29/1/2018	3/2/2018																						
2.1.1	Detailed Use Case 1 - Basic & Alternate flow, Pre & Post conditions	Whole Team	3	29/1/2018	31/1/2018																						
2.1.2	Detailed Use Case 2 - Basic & Alternate flow, Pre & Post conditions	Whole Team	3	1/2/2018	3/2/2018																						
2.2	UML Class Diagram		5	4/2/2018	13/2/2018																						
2.2.1	Design UML Class Diagram using the Use Cases as the basis	Spyros Lontos	5	4/2/2018	8/2/2018																						
2.2.1	Class Diagram descriptions	Spyros Lontos	5	9/2/2018	13/2/2018																						
2.3	Project Plan		1	14/2/2018	16/2/2018																						
2.3.1	Design gantt chart with formulas	Rayner Da Cruz	1	14/2/2018	14/2/2018																						
2.3.2	Fill in dates and timings for the project	Adriano Sole	2	15/2/2018	16/2/2018																						
2.4	Risk Assessment	Alfie Potter	4	15/2/2018	18/2/2018																						
2.5	Milestone: Review Before submission	Whole Team	2	19/2/2018	20/2/2018																						

Project Name	Project Duration	Project start date	Project end date
Quiz - Team 4	106	6/11/2017	20/2/2018

Task ID	Task Description	Assigned to	Task Duration	Start Date	End Date
1	**Design Phase 1**				
1.1	Functional Requirements		11	6/11/2017	16/11/2017
1.1.1	Draft functional requirements	Ryan Harris	5	6/11/2017	10/11/2017
1.1.2	Acceptance Criteria for each requirement	Ryan Harris	5	11/11/2017	15/11/2017
1.1.3	Catergorize each requirement into MoSCoW notation	Whole Team	1	16/11/2017	16/11/2017
1.2	Non-Functional Requirements		13	17/11/2017	27/11/2017
1.2.1	Draft Non-functional requirements	Rayner Da Cruz	4	17/11/2017	20/11/2017
1.2.2	Acceptance Criteria for each requirement	Rayner Da Cruz	3	21/11/2017	23/11/2017
1.2.3	Catergorize each requirement into MoSCoW notation	Whole Team	1	23/11/2017	23/11/2017
1.3	Use Case Diagram	Spyros Lontos	5	23/11/2017	27/11/2017
1.4	Use Case Descriptions:		7	28/11/2017	4/12/2017
1.4.1	Setup Quiz	Reem Alotaibi	2	28/11/2017	29/11/2017
1.4.2	Take Quiz	Alfie Potter	2	30/11/2017	1/12/2017
1.4.3	Obtain Statistics	Adriano Sole	2	2/12/2017	3/12/2017
1.4.4	Modify Questions	Spyros Lontos	1	4/12/2017	4/12/2017
1.5	Milestone: Review Before submission	Whole Team	2	4/12/2017	5/12/2017
	Christmas Break and Exam Period				
	Christmas Break and Exam Period		50	8/12/2017	26/1/2018
2	**Design Phase 2**				
2.1	Detailed Use Case Descriptions		6	29/1/2018	3/2/2018
2.1.1	Detailed Use Case 1 - Basic & Alternate flow, Pre & Post conditions	Whole Team	3	29/1/2018	31/1/2018
2.1.2	Detailed Use Case 2 - Basic & Alternate flow, Pre & Post conditions	Whole Team	3	1/2/2018	3/2/2018
2.2	UML Class Diagram		5	4/2/2018	13/2/2018
2.2.1	Design UML Class Diagram using the Use Cases as the basis	Spyros Lontos	5	4/2/2018	8/2/2018
2.2.1	Class Diagram descriptions	Spyros Lontos	5	9/2/2018	13/2/2018
2.3	Project Plan		1	14/2/2018	16/2/2018
2.3.1	Design gantt chart with formulas	Rayner Da Cruz	1	14/2/2018	14/2/2018
2.3.2	Fill in dates and timings for the project	Adriano Sole	2	15/2/2018	16/2/2018
2.4	Risk Assessment	Alfie Potter	4	15/2/2018	18/2/2018
2.5	Milestone: Review Before submission	Whole Team	2	19/2/2018	20/2/2018



Risk Management

Risk	Likelihood	Impact	Strategy to Minimize Disruption
• A team member is unable to complete their tasks or contribute to the project due to unforeseen circumstances	Low	Medium	- Having several forms of communication in the team, so if such thing was to occur we could deal with it as soon as possible. - An additional method to minimize this would be to assign jobs to pairs of people, so not only do tasks get completed quicker, but they also don't get left unfinished due to a single team members absence
• Project is not completed on time	Low	Large	- Assigning out the roles equally, to ensure all tasks are worked on and therefore completed by deadline - Make final checks on each part of the coursework to check for any incomplete/ missed sections - Follow our Gantt Chart to keep the team organized and all on the same page to what should have been completed by certain specific dates
• Project files corrupting/ getting lost	Medium	Large	- As a team we are collaborating on each part of the coursework using the google drive. This not only is great for us to all contribute to our project but also reduces the risk of losing any of our work. - We are also making offline copies every 3 days in case there is a problem with google drive, stopping us accessing our work

● Team members poorly communicating	Low	Large	- We have set up several forms of communication from Facebook group chats to a google drive folder, and this gives a formal and an informal way to communicate opinions, issues and anything to do with the project - We also meet up on a regular basis in person, to give members the chance to express things in person and to give the members who don't like to communicate virtually, the chance to communicate.
● Not meeting the clients needs	Low	Large	- During and after completed each section of the coursework, the whole team re-reads the specification and double checks we have met each requirement to the standard wanted - We have also been getting feedback from the available help in our labs to get second opinions on our work, to avoid any bias
● Team members not using the same version of the programming language	Low	Medium	- To minimize the risk of code incompatibility between different version of the same language we will make sure we settle on one version of the language and make sure we periodically install updates.