Application development project

In a team of two develop a software system for the education sector.

Elicit 10 software requirements for your application in the form of User Stories.

Provide a complete set of UML diagrams including Use Case Diagrams, Sequence Diagrams, Class Diagrams and Architecture Diagrams.

Produce a low fidelity mock-up of the user interface.

Outline set of a System tests that can be used to validate the User requirements.

Objective

To develop a mobile application that delivers online educational resources for students and lecturers in a college. It should allow lecturers to post notes, make announcements, upload assignment questions, post quizzes and surveys and update grades for the modules that they teach. It should allow students to view notes, receive notifications of announcements, upload their assignments, attempt quizzes and check grades for a certain module their enrolled in. It should allow students to access a timetable of all their module class times and check grades for all their past and present modules. It should allow students and lecturers to create group chat or message any other user in that college. This application would compile all separately available resources into the one location.

Amber's User stories

User Story
As a lecturer, *I want to* post lecture notes for the modules I teach *so that* my student can catch up on lecture material.
Acceptance criteria
Given: A lecturer is on the notes page for their module.
When: The lecture selects notes and clicks upload.
Then: The notes are uploaded so that their students can see those notes.

User Story
As a student, *I want to* be able to see all notes uploaded in a certain module section *so that* I can keep up to date with all lecture material.
Acceptance criteria
Given: A student is logged in and on a module section.
When: The lecture selects notes and clicks upload.
Then: All notes that have been uploaded will be shown to the student.

User Story
As a student, *I want to* be notified when a lecturer uploads announcement's *so that* I don't miss important deadlines.
Acceptance criteria
Given: A student has turned on notifications.
When: The lecturer chooses to upload an update announce.
Then: The student receives an email notification and will see a flag on the new announcement the next time they log in.

User Story
As a lecturer, *I want to* message a lecturer directly through the college site *so that* I can receive advice to do with that module.
Acceptance criteria
Given: A student is on a certain module's notes section and can see which lecturer uploaded though notes.
When: The student clicks on the lecturer's name and type a message to send.
Then: A message will be sent and received the lecturer.

User Story
As a student, *I want to* see a timeline of all my assignments due *so that* I can upload and receive marks for that assignment.

Acceptance Criteria
Given: A student on the assignment timeline and there are assignments due to be uploaded.
When: The student chooses the file and clicks upload.
Then: Then the assignment is recorded as being submitted and the lecturer can grade it.

Melat's User Story

User Story
As a student, I wants to access apps like Timetable, so that I can view my schedule.
Acceptance Criteria
Given: A student is on the Dashboard page.
When: The student selects Timetable.
Then: Then the student is able to view and download their personal schedule.

User Story
As a student I want to have access to previous modules, so that I can revise.
Acceptance Criteria
Given: A students logs in and selects past module.
When: The student has access to all past notes.
Then: Then the student is able to read and download past notes.

User Story
As a lecturer I want to be able to upload quizzes and/or sample exams, so that I can keep track of each students' improvement.
Acceptance Criteria
Given: A lecturer is on their module page.
When: A lecturer uploads a quiz/sample exam with a time limit.
Then: Then the students are able to complete the quiz/sample exam in a limited amount of time.

User Story
As a lecturer, I want to have a survey at the end of each year, so that I can improve my teaching methods.
Acceptance Criteria
Given: A Lecturer in on their module page.
When: Then, the lecturer uploads survey.
Then: The students complete the survey.

User Story
As a student, I want to be able to communicate with my lecturers, so that they can answer any queries I have on that module.
Acceptance Criteria
Given: The student logs in and select a lecturer to whom they'd like to talk to.
When: The students types the query.
Then: Then the lecturer replies to the student.

Melat’s Use Case Diagram

The use case diagram shows the relationship between lecturers and students.

Once the lecturers login they are able to upload notes to their modules they teach. Including being able to upload surveys/ sample exams and upload end of year survey, for student to comment on the lecturer teaching methods and for the lecturer to improve teaching skills.

Once the student logs in, they are able to access timetable, module notes, quizzes/sample exams, surveys and be able to communicate with lecturers. If a lecturer was to upload an announcement the student would receive a notification to inform of the announcement.

Once the student logs in they are able to see the timeline of each assignment which informs them of when an assignment is due, and view their grades.

Depending on the year of a student they can access past modules. When the student accesses a past module they can view past notes, assignments and quizzes.

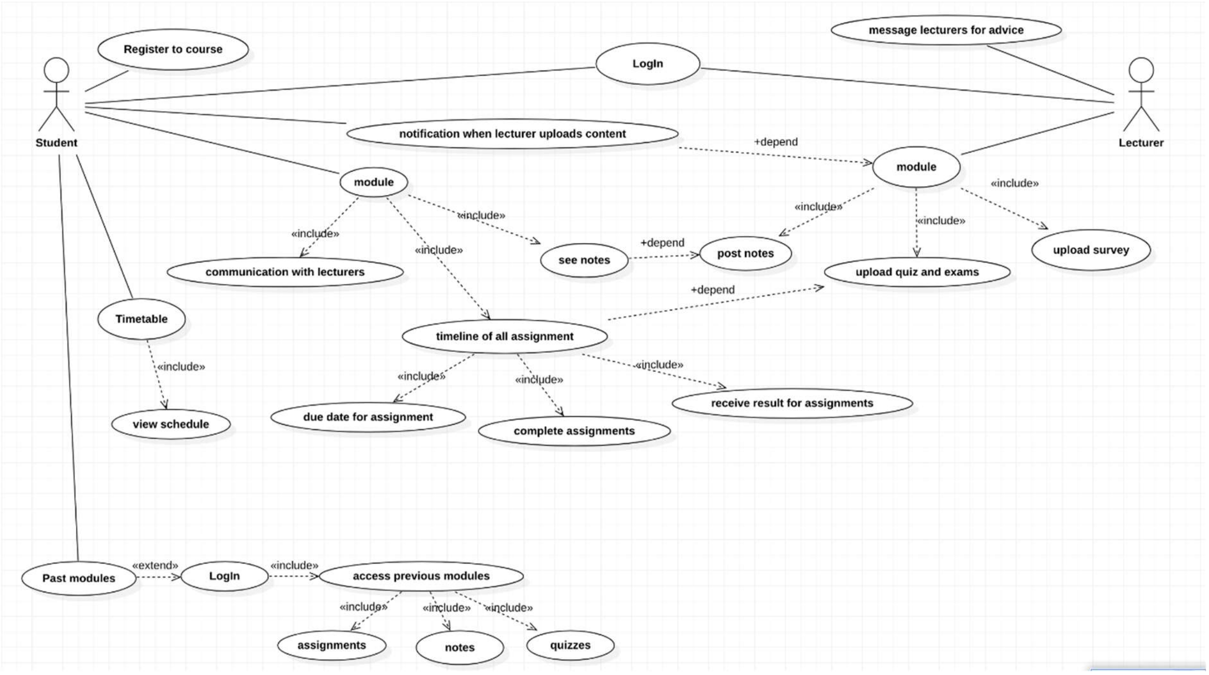
In the use case diagram, I used 4 different types of relationships.

Include – I used include to show the relationship between two or more use cases. For example for module I included communication with lecturers, timeline assignments and see notes.

Extend – I used extends to show that a use case adds steps to another first-class use case. For example, login is extended from past modules.

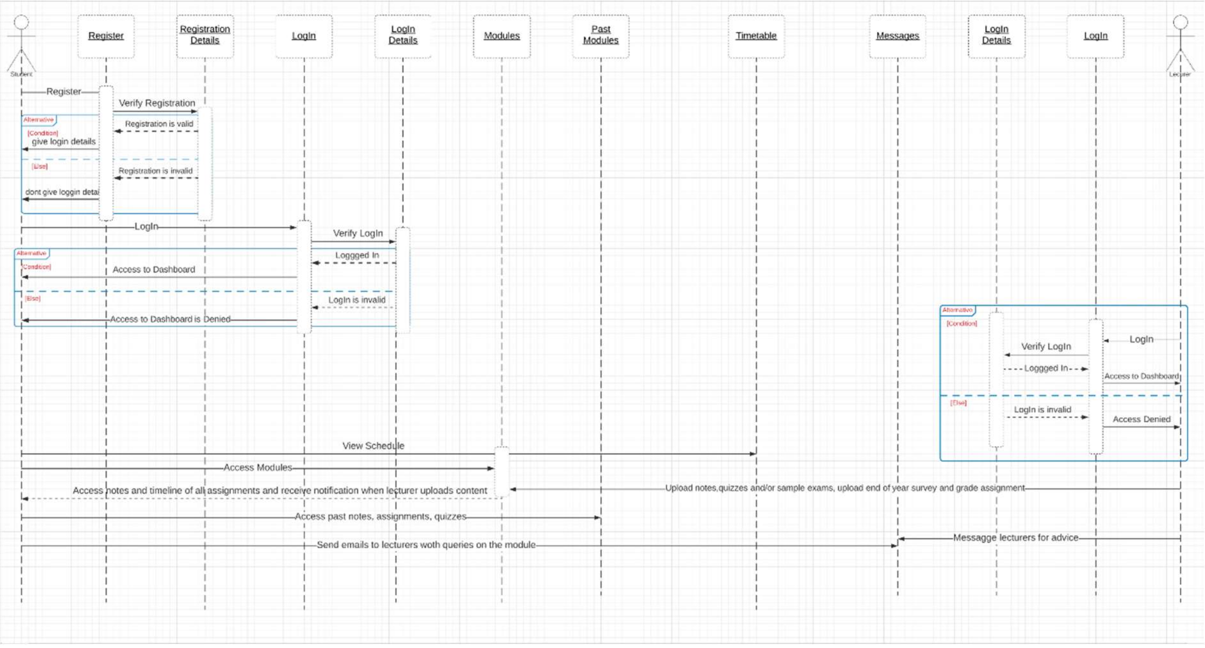
Dependency – I used this relationship to show that a use case depends on another use case for it to work.

Association – To shows a relationship between the actor and use case.



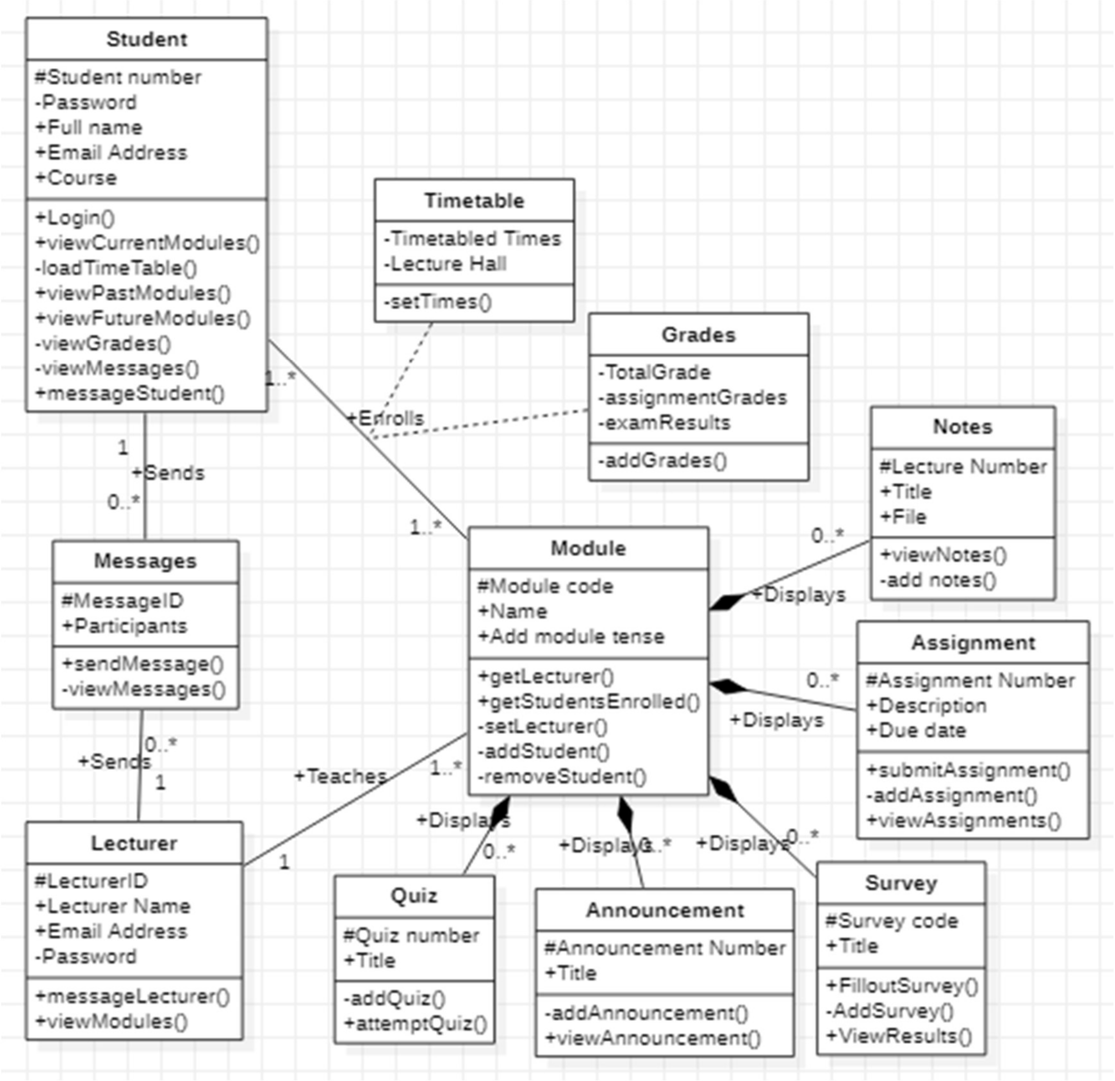
Melat’s Sequence Diagram

The sequence diagram consist of 10 objects. I have 2 different login objects, one for students and the other for lecturers. At the start of the year each student has to register and depending if their registration is successful then they can login and have access to their modules, timetable and be able to communicate to lecturers. The dotted lines coming down the objects and actors is the lifeline which represents an individual participant. The white block on the lifeline is to show the use of each object. The lines pointing right is to show a connection. A line pointing to the left is to show a return message.



Amber’s Class Diagram

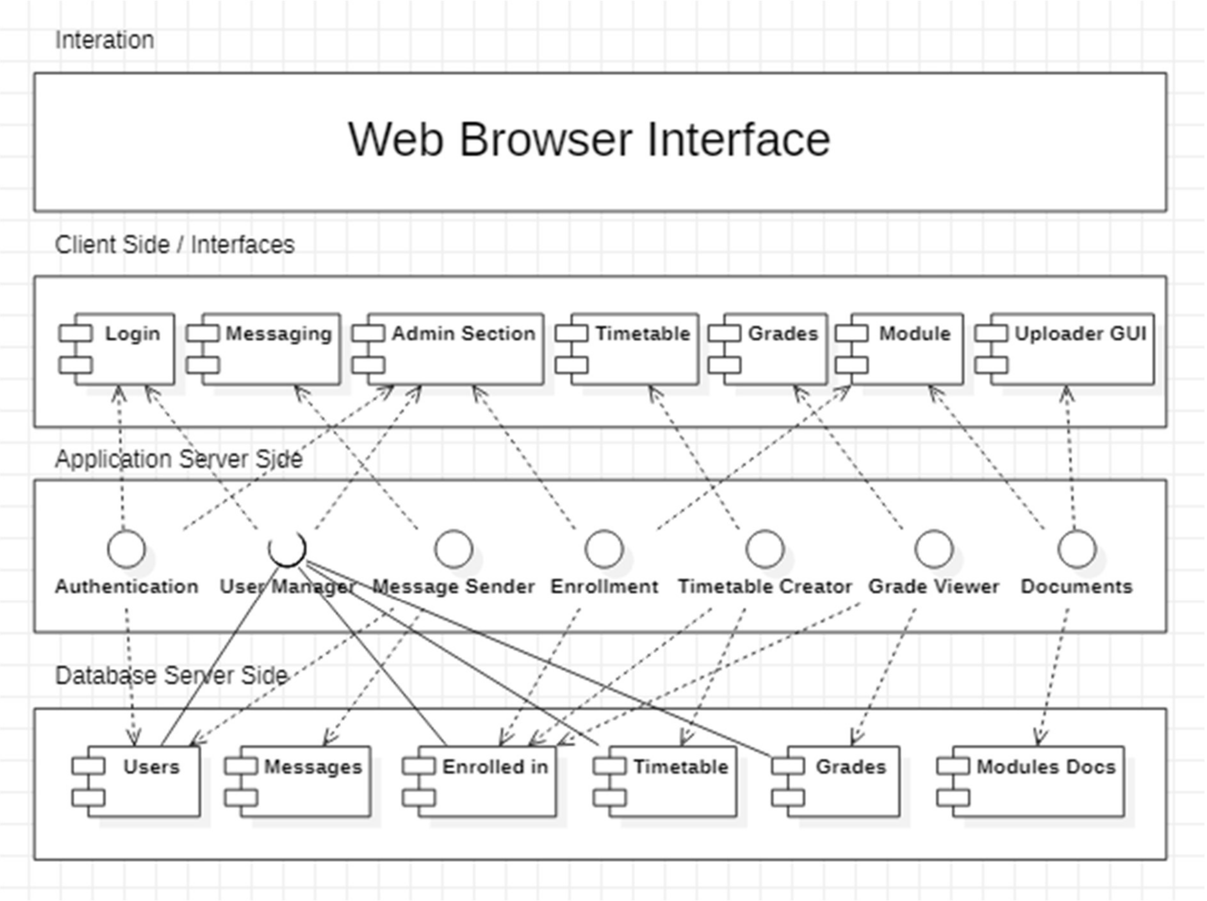
The module class has the child classes notes, assignment, survey, announcements and quiz. If a lecturer is a teacher of the parent module, they can access any of the methods from the child classes. Students have limited access here and can not add any documents to the module apart from submitted assignments. Users can only access their own timetable and grades. Only admin users can enrol students in a module or change the rights of a lecturer. All users can message any other user regardless of rights. All method requiring limited access have been made private for security.



Amber’s Architecture Diagram

When a user logs in their details are authenticated. If authentication is successful. Then a certain interface will be displayed them depending on their usage rights and the modules they’re enrolled in. From the home screen they can choose to see modules, timetable, grades or messages which will be return required information from the database by the database. Lecturers will have more right within the modules they teach than other users. Admin users sign into a special admin section which allows them to manage users, enrolled users in modules and manipulate all data within the database.

There will be multiple tables in the database. A table called “users” containing encrypted authentication details, user ID, personal details and usage rights. A table called “messages” containing user ID, display names, messaging rights and previous messages. A table called “timetable” containing module times and module locations. A table called “grades” containing user ID and grade history for each student. A table called “enrolled in” containing a list of every student and modules they have been enrolled in. A table called “modules” contains a record of all documents uploaded in a module and the User ID of the uploader.



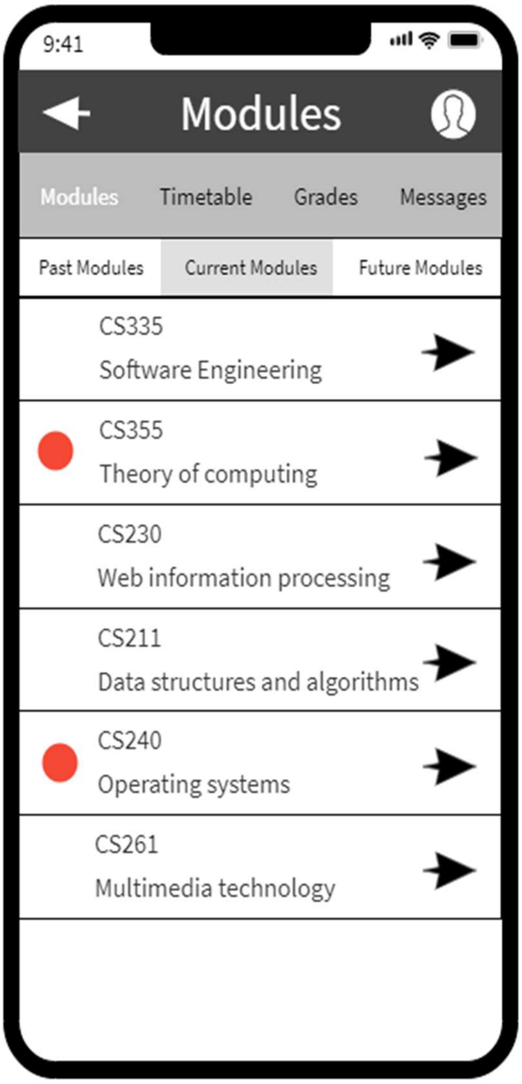
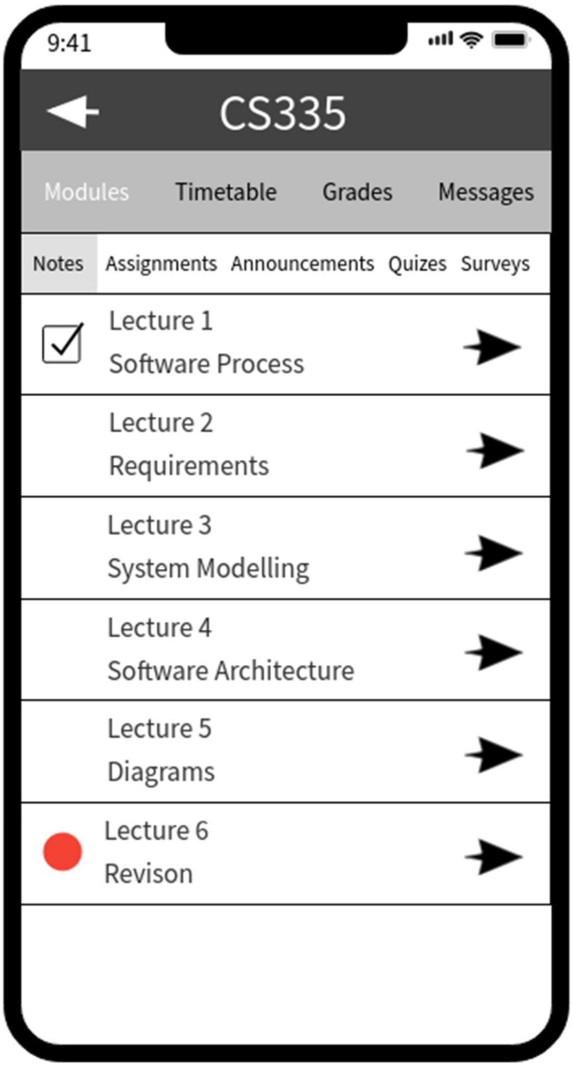
Amber's Low Fidelity Mockups from student perspective

1.0. Modules

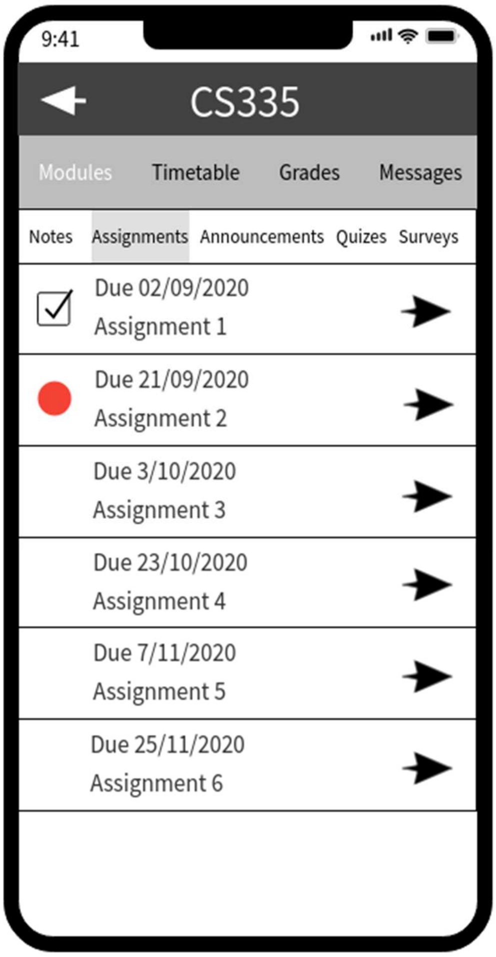
2.0. Notes

3.0. Assignments

4.0. Assignments Upload

1.0. Modules	**2.0. Notes**
 9:41 Modules Modules Timetable Grades Messages Past Modules Current Modules Future Modules CS335 Software Engineering CS355 Theory of computing CS230 Web information processing CS211 Data structures and algorithms CS240 Operating systems CS261 Multimedia technology	 9:41 CS335 Modules Timetable Grades Messages Notes Assignments Announcements Quizes Surveys Lecture 1 Software Process Lecture 2 Requirements Lecture 3 System Modelling Lecture 4 Software Architecture Lecture 5 Diagrams Lecture 6 Revison

3.0. Assignments



☑ Due 02/09/2020
Assignment 1

● Due 21/09/2020
Assignment 2

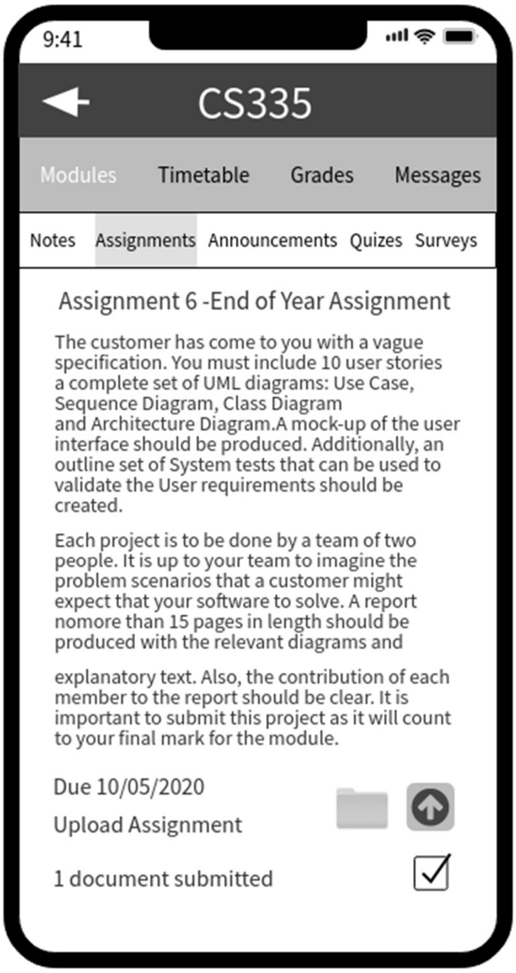
Due 3/10/2020
Assignment 3

Due 23/10/2020
Assignment 4

Due 7/11/2020
Assignment 5

Due 25/11/2020
Assignment 6

4.0. Assignment Notes



Assignment 6 -End of Year Assignment

The customer has come to you with a vague specification. You must include 10 user stories a complete set of UML diagrams: Use Case, Sequence Diagram, Class Diagram and Architecture Diagram.A mock-up of the user interface should be produced. Additionally, an outline set of System tests that can be used to validate the User requirements should be created.

Each project is to be done by a team of two people. It is up to your team to imagine the problem scenarios that a customer might expect that your software to solve. A report nomore than 15 pages in length should be produced with the relevant diagrams and

explanatory text. Also, the contribution of each member to the report should be clear. It is important to submit this project as it will count to your final mark for the module.

Due 10/05/2020
Upload Assignment

1 document submitted ☑

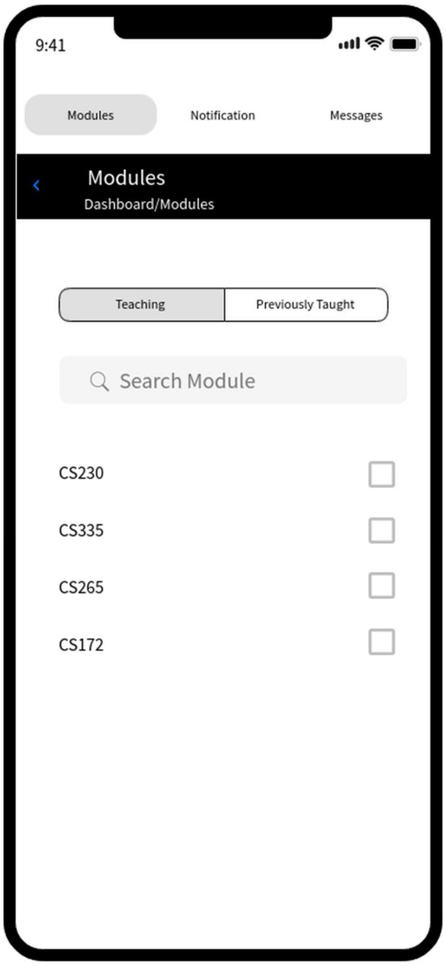
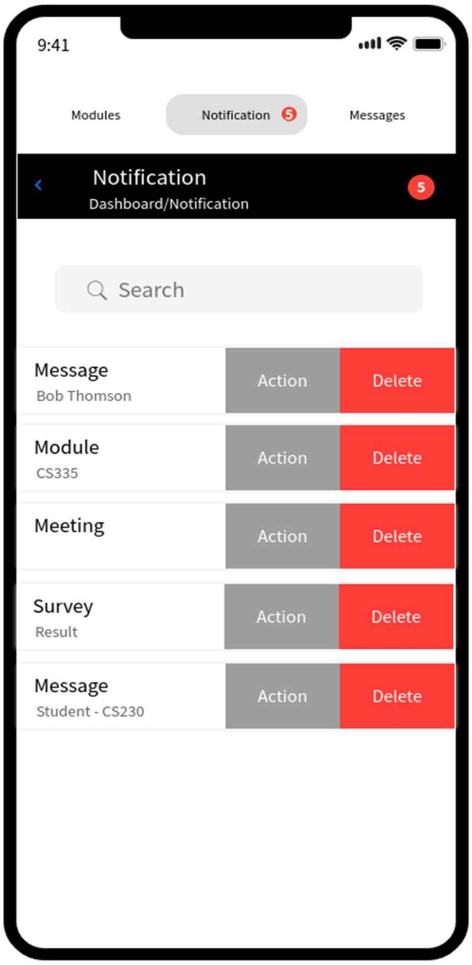
Melat’s Low Fidelity Mockups from lecturer perspective

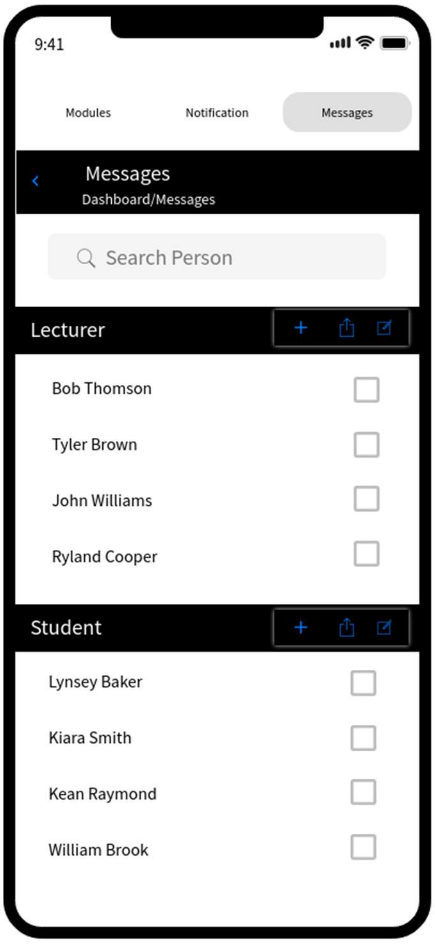
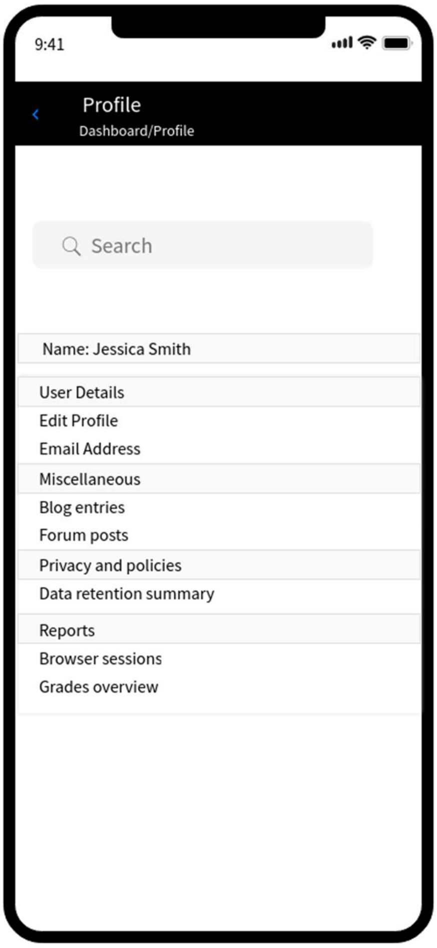
1.0. Modules – Lecturer Perspective

2.0. Notification – Lecturer Perspective

3.0. Messages – Lecturer Perspective

4.0. Profile

1.0. Modules – Lecturer Perspective	**2.0. Notification-Lecturer Perspective**
 9:41 Modules Notification Messages Modules Dashboard/Modules Teaching Previously Taught Search Module CS230 □ CS335 □ CS265 □ CS172 □	 9:41 Modules Notification 5 Messages Notification Dashboard/Notification 5 Search Message Bob Thomson Action Delete Module CS335 Action Delete Meeting Action Delete Survey Result Action Delete Message Student - CS230 Action Delete

3.0.Messages-Lecturer Perspective	4.0. Profile
 9:41 Modules Notification Messages Messages Dashboard/Messages Search Person Lecturer Bob Thomson □ Tyler Brown □ John Williams □ Ryland Cooper □ Student Lynsey Baker □ Kiara Smith □ Kean Raymond □ William Brook □	 9:41 Profile Dashboard/Profile Search Name: Jessica Smith User Details Edit Profile Email Address Miscellaneous Blog entries Forum posts Privacy and policies Data retention summary Reports Browser sessions Grades overview

Amber's System Test

Test Case	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TT.0	Check lecturers can post notes on the module page that they teach.	1. Log in as a lecturer. 2. Go to module notes page. 3. Click upload notes. 4. Choose pdf file to upload. 5. Choose title of notes. 6. Click Submit.	UserID =LecturerAnne Password =panda NotesFile =Lecture7.pdf	Uploaded notes should now show on the notes page with a red dot beside them as they are new.	As Expected	Pass
TT.1	Check students can post notes on the modules their enrolled in.	1. Log in as a student. 2. Go to module notes page.	UserID =StudentJack Password =bear NotesFile =Meme.pdf	There should be no option to upload notes.	As Expected	Pass
TT.2	Check students can view all notes of modules their enrolled in.	1. Log in as a student. 2. Go to module notes page.	UserID =StudentJack Password =bear	All the notes the lecturer has previously uploaded are visible.	As Expected	Pass
TT.3	Check if a student gets notified when a lecturer uploads an announcement.	1. Log in as a student. 2. Go to notification page	UserID =StudentJack Password =bear	Notication page should have an alert for the new lecturer announcement.	As Expected	Pass
TT.4	Check if a lecturer can message a student.	1. Log in as a lecturer. 2. Go to message page 3. Choose student from list 4. Type message 5. Click send	UserID =LecturerAnne Password =panda	Message will be sent to lecturer.	As Expected	Pass

TT.5	Check if a student can find the lecturer to message by clicking on lecture notes.	1. Log in as a student. 2. Go to module notes page 3. Click on lecture notes details 4. Click on the lecturer name who uploaded them 5. Click message	UserID =StudentJack Password =bear	A message conversation will be opened up between student and lecturer.	As Expected	Pass
TT.6	Check if a student can view an ordered timeline of assignments to be uploaded.	1. Log in as a student. 2. Choose module 3. Choose assignments tab	UserID =StudentJack Password =bear	A timeline of all assignments due for that module will be displayed.	As Expected	Pass
TT.7	Check if a student can upload a due assignment.	1. Log in as a student. 2. Choose module 3. Choose assignments tab 4. Choose specific assignment 5. Choose file 6. Click upload	UserID =StudentJack Password =bear	Status of assignment changes to one document uploaded.	As Expected	Pass

Melat's System Test

Test Case	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TT.0	Check Students can access to Timetable, to view their schedule	7. Log in as a student. 8. Go to apps and then Timetable. 9. Click View Timetable. 10. They can choose to download as a pdf.	StudentID =Katie Password =katie123 Download =Timetable.pdf	Download Timetable or view Timetable online.	As Expected	Pass
TT.1	Check students can view all their past modules including notes, assignments and quizzes.	3. Log in as a student. 4. Go to past modules notes page. 5. View all notes.	StudentID =Tom Password = Tom1234 NotesFile =pastnotes.pdf	All the past notes the lecturer has previously uploaded are visible.	As Expected	Pass
TT.2	Check lecturers can upload quizzes/sample exams	3. Log in as a lecturer. 4. Upload quizzes/exam samples.	LecturerID =William Password =William123	Upload quizzes/sample exams	As Expected	Pass
TT.3	Check lecturers can send messages to other lecturers	3. Log in as a lecturer. 4. Search for lecturer. 5. Click on the lecturer, to contact them	LecturerID =Kyle Password =Kyle123	A message conversation will be opened up between a lecturer and another lecturer. Usually a lecturer messages another lecturer for advice.	As Expected	Pass
TT.4	Check if a lecturer can upload surveys at the end of each year.	6. Log in as a lecturer. 7. Go to a module page.	LecturerID =Jennifer Password = Jennifer123	At the end of each year, the lecturer uploads a survey, for student to	As Expected	Pass

		8. Upload survey.		comment on the lecturer teaching methods.		