Project Proposal
CUSP Crasher

01/09/2018

Group 24

Wayne Yao 470324934

Rufeng Tu 470132199

Kun Zhang 420033455

Ziqi Wang 460100227

1. Introduction

CUSP Crachser is an online information platform that helps the student users to select their Unit of Study ("unit" or "UoS" in this proposal) and to prepare for the quizzes and exams.

1.1 Overview

- This project consists of displaying pages and submitting pages
- This platform provides basic information on all the units and lecturers in School of IT in USYD to any user
- This platform allows any users to search the unit and lecturers by name, code, tags and so on
- This platform allows unregistered user to register
- This platform allows registered user to login, edit profile and password
- This platform allows registered user to create reviews on an unit or a lecturer
- This platform allows registered user to create and share sample questions
- This platform allows registered user to upvote a review or a sample question
- This platform allows registered user to view their own history of reviews on an unit, a lecturer or the sample question bank
- This project will be written in Java with Spring MVC + Hibernate Frameworks
- It will be delivered as a web application running on Tomcat

1.2 Aims

- To help users to know about an unit or a lecturer via reviews from others
- To allow users to provide reviews on an unit or a lecturer to help others
- To allow users to create their own questions and prepare for the quiz or exam

1.3 Primary User

- USYD students who are overwhelmed by different units at the beginning of new semester, who ask like crazy on social media “how’s this unit? Is it hard? Is it programming involved? Is the final torturing?”
- USYD students who want to know more about a lecturer of SIT
- USYD students who want to get prepared for the quizzes and exams by self-testing online instead of going through slides for a hundred times and finding out that none of them appears in the exam.

2. Significances

This project is inspired by the common scenarios in the online communication chatting group or social media group, where students interested in a certain unit constantly ask others about the detail information and some comments of the unit or the lecturer. Most of the time, the answers to those questions are not easy to find on traditional CUSP. E.g., To which level of programing skill do I need to be qualified as “understanding Java programing”. Plus it’ll be a really tough job to dig the chat history if a student wants to find someone’s comment couple of days later as that would definitely be buried as hell. The only thing he can do is to ask the same question again, which is exactly the issue we’re going to address in this project.

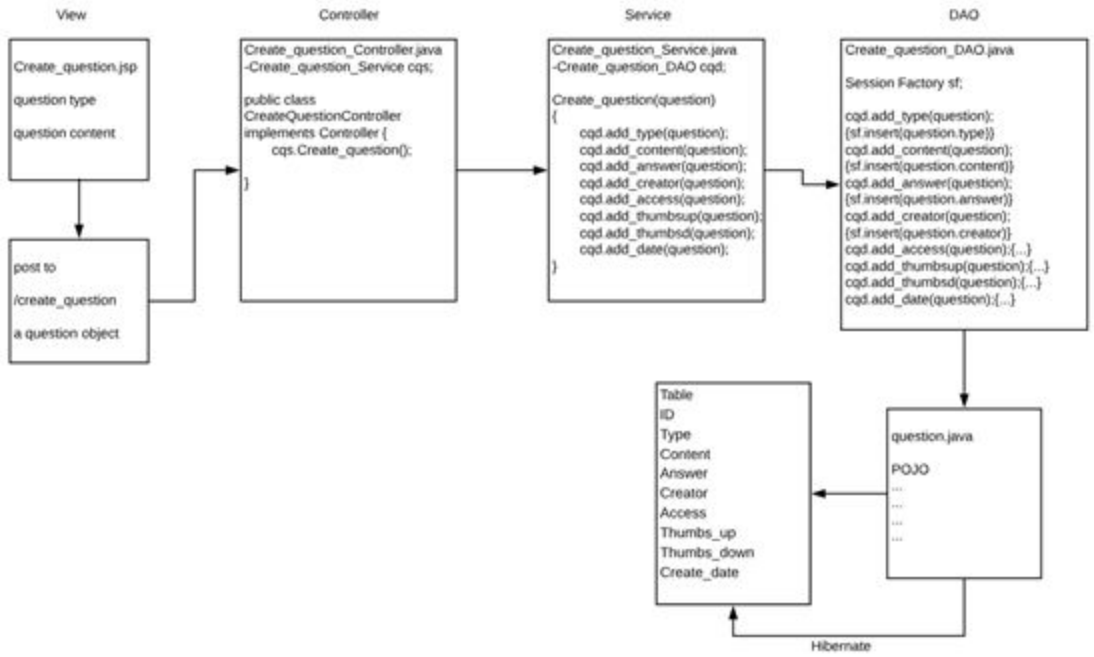
Although there are some websites providing the comments on a certain unit, the rating systems are too general. For example, the website studentvip.com only provide one overall rating on the unit, and there is no rating or comments on the lecturer, nor do they have teaching history of lecturers [1].

Given that being said, a platform where students could easily find the detail reviews on aspects such as "Stress", "Difficulty", "Usefulness" on an unit. Moreover, we also provide platform for users to view and contribute on ratings like "Teaching Ability" and "Responsiveness" on a lecturer.

In addition, our website provide a feature that user could create sample questions, share with others and have self-tests with private and public sample question bank. This function is also unseen in all the existing unit rating websites.

3. Project Outline

3.1 System architecture



3.2 Component division

As mentioned in previous description, CUSP Crachser is an online information platform that helps the student users to select their UoS and to prepare for the quizzes and exams. Based on this guideline, the main component can be divided as follows.

3.2.1 The user registration system.(managed by Wayne)

The main features of this part include registration, login and management of user profile. In this part, users are allowed to manage their personalized information. Also their previous given comments and lecture rating will be displayed as well.

3.2.2 The units reviewing system. (managed by Kun)

The main features included in this part are showing key static information of a unit, showing unit reviews and rating the units in terms of difficulty, stress and practicality. With the help of this system, users will be able to decide whether a UoS fits their needs for the study of the current semester.

3.2.3 The lecturers reviewing system. (managed by Rufeng Tu)

The main features of this system are showing the teaching history of a lecturer, rating the lecturer in terms of year of the review, teaching ability as well as responsiveness. The users can use information provided by this system as reference while selecting a UoS given by a certain lecturer.

3.2.4 The questions creation and reviewing system. (managed by Ziqi Wang)

The main features in this part include Reviewing question creation, taking quiz, viewing your questions and question discussion board. Users can create question and take quiz related to a UoS for the purpose of either improving their understanding or preparing for exams.

4. Project Plan

The project plan and timeline will be described in plain English as the following:

1. Project planning stage (14/08/2018 - 20/08/2018)

At this stage, brainstorming about what we're going to do, defining what kind of problem we are going to resolve and how we'll solve it. Define the development methodology (probably waterfall model). The output of this stage is the raw idea that is embed in everyone's mind about our website.

2. Development plan stage (25/08/2018 - 03/09/2018)

2.1 logic design and individual feature implementation (25/08/2018 - 28/08/2018)

Assign team member with different features and logic. Team member can volunteer to take the responsibility of a specific feature and discuss the difficulty he might be faced with. Also the workload is another consideration. We should guarantee that the workload for each member is roughly the same.

2.2 Data model design (29/08/2018 - 03/09/2810) Milestone

Determine the overall database structure design. Upon everyone agreeing on a main framework of the data model, they can add necessary attributes and entities to the framework based on their assigned feature. The result of this activity is a complete ERD and ER relation model that will be used in the following development. This stage is a milestone.

3. Development (09/09/2018 - 16/10/2018)

3.1 Individual development (12/09/2018 - 06/10/2018)

All team members are dedicated to the implementation their features in this stage. We'll need to complete a frontend page, which is responsible for the presentation of data and our business logic. We'll also need to program the actual backend logic. E.g., clicking this button will trigger an operation in database and redirect user to another specific page with specific content. Due to the nature of those separated features, we may or may not have overlapped needs such as some database operation or some common utility functions. Hence it's important not to implement the same function twice. To resolve this problem, we decided to interactive with each other in API fashion. Once a member has some API ready, he announces it to the whole team. Note that dangling function will be gathered in a single place (e.g. tools/ or utils/) for maintenance purpose.

3.2 Individual testing (06/10/2018 - 10/10/2018)

Team members are to unit testing their code to make sure the functionality and usability of their unit with proper testing code. This stage mainly focuses on if APIs are stable and robust and if APIs can handle error-prone input correctly. Bugs that are to influence main business logic are the highest priority issue and must be fixed as soon as possible.Upon the closure of this substage, we can proceed to integrity testing.

3.3 Integrity testing (10/10/2018 - 16/10/2018) Milestone

The integrity testing mainly focuses on whether different units of code properly works together to produce expected result, whether the coherence and cohesion of the product of different members make the whole website an integrity object. Most importantly, is the functionality of the website is fulfilled. I.e., whether the correct input yields reasonable result or more precisely, whether the coordination of different APIs is smooth. This stage is considered as milestone as it marks the birth of stable version of our website.

4. Post maintenance (16/10/2018 - 20/10/2018) Milestone

The post maintenance stage involves the completion of code document. Team members are expected to produce a copy of detailed document of their code on the functionality, exception handling and possible breakage of API in the future. The closure of this stage is a milestone as it formally indicates the termination of this project.

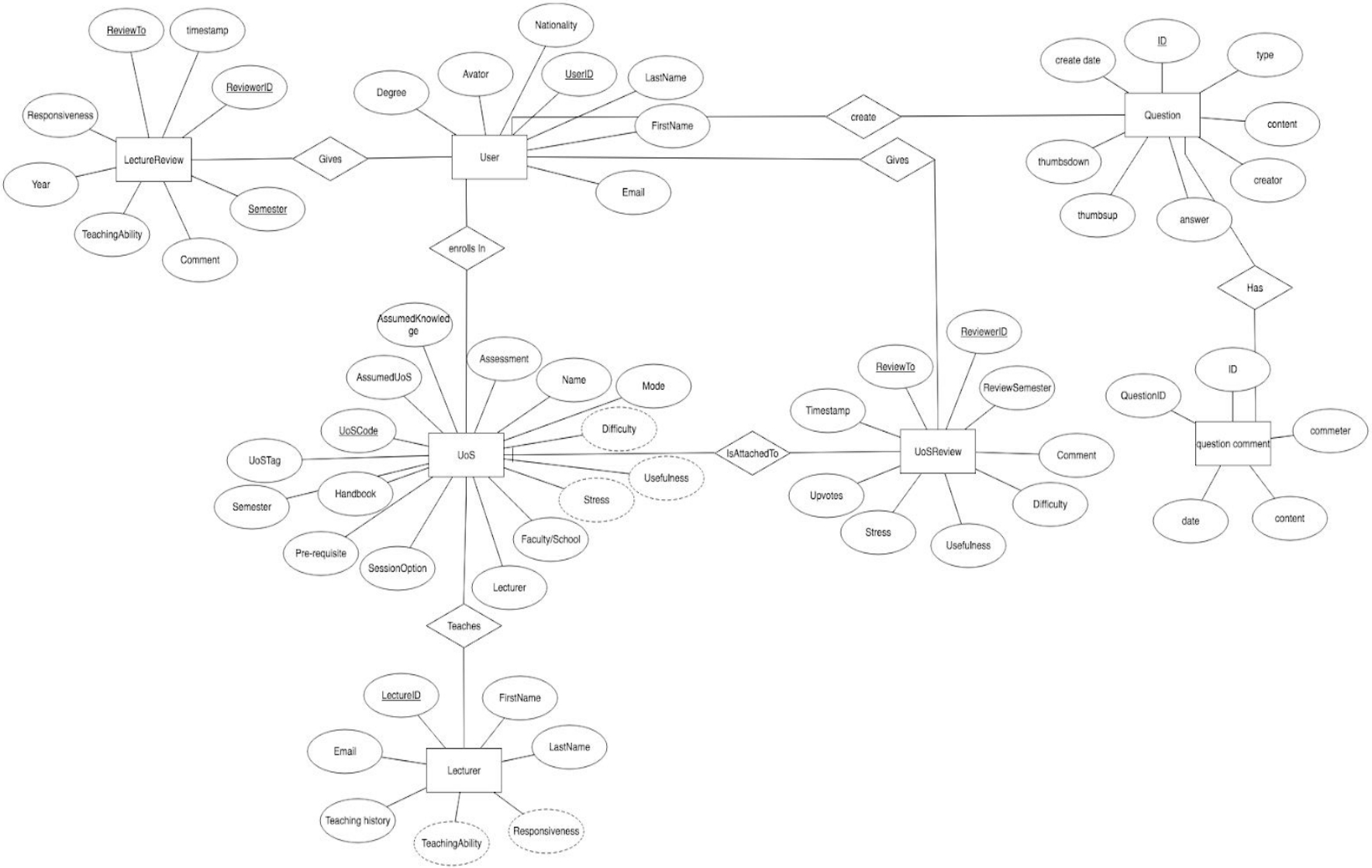
5. System Data Model

The ER Diagram of the whole database is the following: (next page)

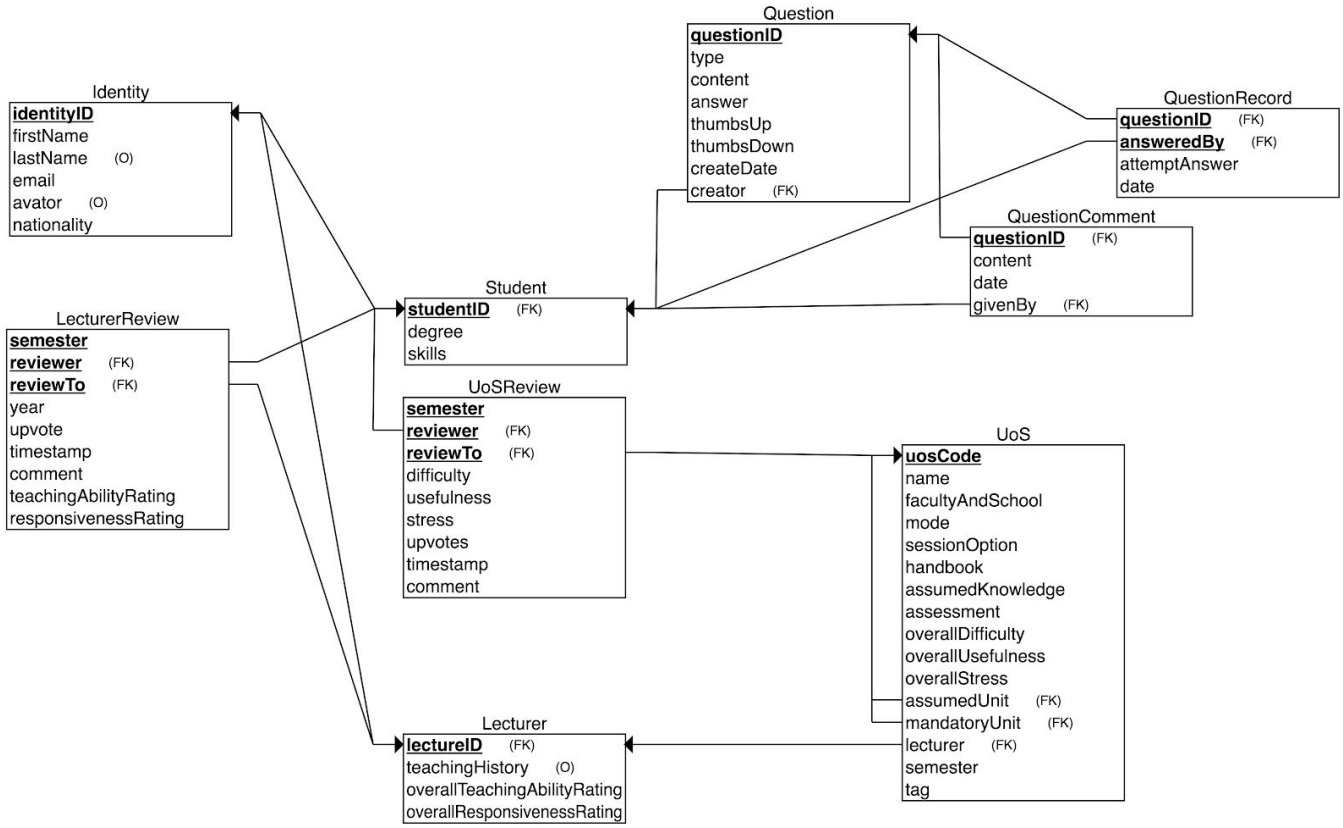
Some notes about this ERD:
1. We create 7 entity type to map our business logic to the data type we're going to store. Their names should be self-explained.
User: our registered student.
UoS: our units that is to be reviewed and commented.
Lecturer: our lecturer that is also to be reviewed.
LecturerReview and UoSReview: entity type that can represent the meta data of action "review". E.g., A UoS review may contain such information: who gave this review (reviewID), which unit this review is related to (reviewTo), what aspects of this unit is actually reviewed (usefulness, stress, difficulty), did this review upvote this unit (upvote). The same logic applies to LecturerReview except that the subject is lecturer now.
Question: a user created question used by self-testing and is public to others (so that they can use this question to test themselves as well).
Question comment: comment on a specific question.

2. The tool we use to create this ERD doesn't use standard notation to represent the cardinality of the relationship of two participating entity type.So we formally make an explanation here:

A user can give 0 or more review to UosReview/LectureReview. A UoSReview/lectureReview must be given by exactly one user in the same semester.
A user can enrolls in 0 or more UoS and vice versa.
A user can create 0 or more questions. A question can only be created by exactly one user.
A lecturer teaches 0 or more UoS. A UoS must be taught by at least one lecturer.
A UoS is attached to 0 or more UoSReview. A UoSReview must be attached to exactly one UoS.
A question has 0 or more question comment. A question comment must be attached to exactly one question.


ReviewTo
timestamp
ReviewerID
Responsiveness
Year
TeachingAbility
Comment
Semester
LectureReview
Gives
Degree
Avator
Nationality
UserID
LastName
FirstName
Email
User
enrolls In
create
Gives
create date
ID
type
content
creator
answer
thumbsdown
thumbsup
Question
Has
ID
QuestionID
question comment
commeter
date
content
AssumedKnowledge
Assessment
AssumedUoS
Name
Mode
UoSCode
Difficulty
UoSTag
Handbook
Semester
Pre-requisite
SessionOption
Faculty/School
Lecturer
Stress
Usefulness
UoS
IsAttachedTo
ReviewerID
ReviewTo
ReviewSemester
Timestamp
Comment
Upvotes
Stress
Usefulness
Difficulty
UoSReview
Teaches
LectureID
FirstName
Email
LastName
Lecturer
Teaching history
TeachingAbility
Responsiveness

The ER relation model translated from ER diagram is the following:



As indicated in the picture, this ER model contains 9 tables that are directly translated from ERD. Note that due to the limit of the tool we use to create ERD, we cannot represent “ISA” relation in the ER diagram but we implement it in our ER model: the entity type “Student” and “Lecturer” is a “Identity”. This separation makes logic clear. Besides these three tables, the “UoS”, “UoSReview”, “LecturerReivew” constitute the core tables that support our main business logic. Users can comment, review, find units or update, delete their history with the support of these six tables. The three tables at up right corner (“Question”, “QuestionRecord” and “Question comment”) are responsible for another feature of our website. The quiz and self-testing feature will need them. Users can create and answer questions, view answer and comment of this question with the support of these three tables. Semantic constraints and data type will be implemented and determined in our actual SQL transaction.

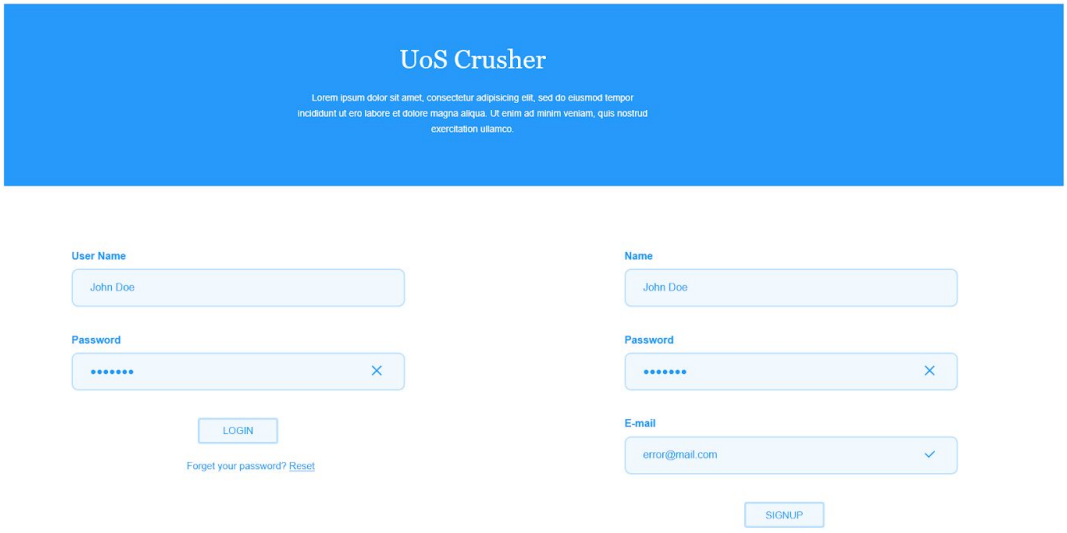
6. Individual Implementation

Login Page and user dashboard (Wayne)

I'm mainly responsible for the design and implementation of login page and user dashboard presentation with proper security measures. The features of these two pages are the following:

In the login page, a user will be able to:
1. Register with university email. After sign-up, they will be redirected to their initial profile page and asked to provide necessary information to complete the sign-up. Once finished they will then be redirected to their profile page.
2. Retrieve password with their email by clicking “forget password?”. In the password reset page a user then will be able to update his password.
3. Login with correct password. For existing user they will be redirected to their profile page.

The mockup of sign-up and login page is the following:



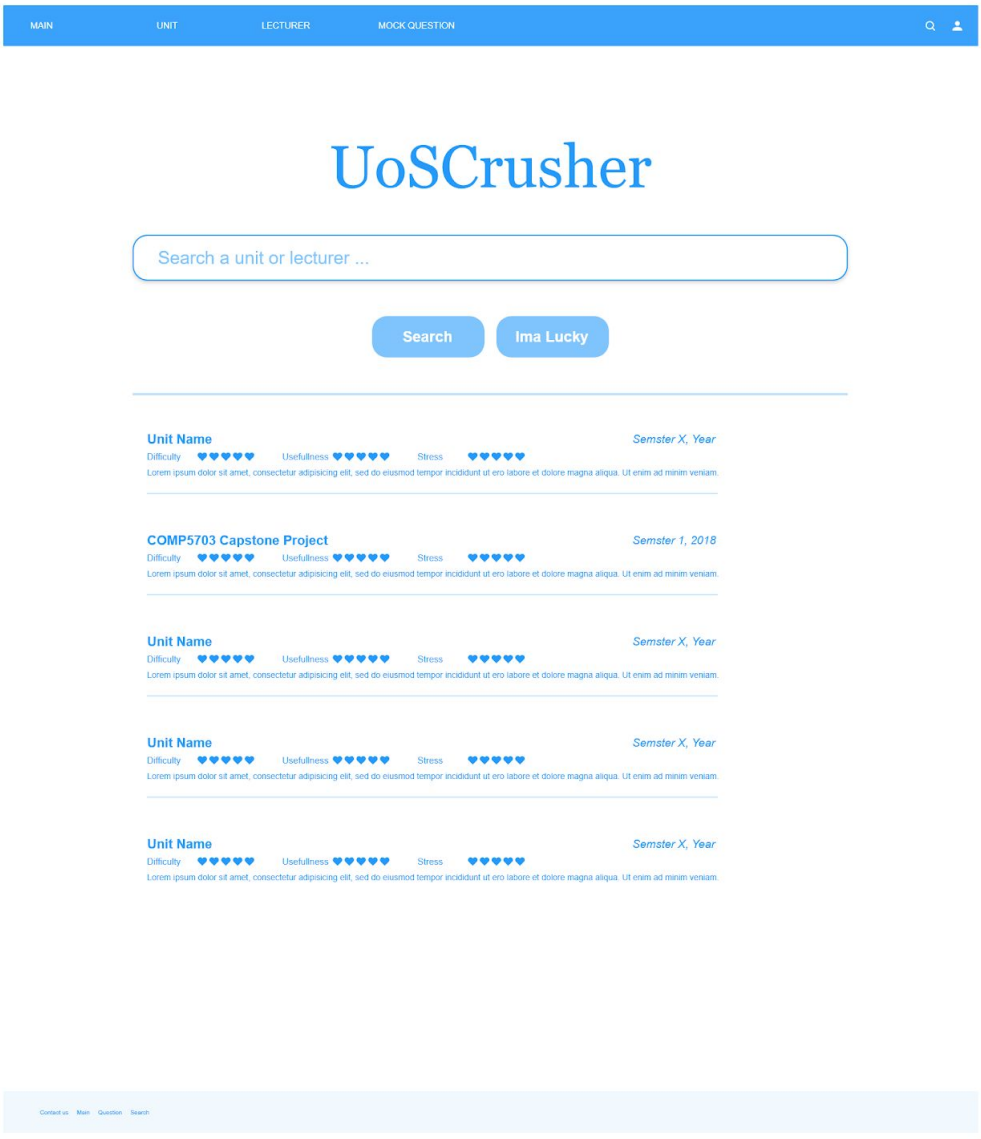
As described by this mockup, user can either login or sign-up. Both action can be done in the same page with their username or email.
In the user profile page, a user will be able to:
1. **View** all his information including avator, degree info, DOB, Email, his skills, chosen units etc. All these information should be provided by user. Refer to the “system data model” for detailed fields. Note that we don’t have permission from university to validate the correctness.

2. **Update** the above information provided by user with “update” button. Some fields should not be modified logically such as year entered or nationality but most of fields are alterable.
3. View his previous given comments and lecture rating.
4. **Modify or delete** previous given comments and lecture rating.
5. Detailed features such as adding or deleting a unit of study cannot be completely covered in this proposal but they’re intuitive, plus the name “dashboard” should be self-explained. So for anything that should be in a dashboard, we’ll implement it at a basic level.

The mockup of user dashboard is the following:
The left column contains user avatar and his basic information. A user can edit any of it in place. The right side column is the user review history including which unit was reviewed or commented by him and detailed review data. Also, a user can choose to edit or delete these review and comment right in place. The top bar includes redirection like homepage that might be useful for users.

Main Landing Page

Following a successfully login, users will be presented with a main landing page where they can search for review pages of a unit or a lecturer. All results related to the user search inputs will be displayed on the same page, if a fussy search is performed. The search results contain a list of hyperlinks to review pages, each of which comes with a description and ratings of the units/lecturers of the page. In addition, an “Ima Lucky” button is provided next to the “Search” button, which on pressed, generates a list of random results.



Unit Review (Kun)

One of the core functions of our web application is to let users browse and writer reviews about IT-related units at University of Sydney. The main features of the component are presented as follows:

1. Show key static information of a unit.

The key information is excerpted from CUSP, including short description, lecturer(s), assumed knowledge, etc. We found that CUSP unit description pages might contain too much information, making it difficult for users, especially potential students, to locate the what they need. So instead, we exclude some not-so-relevant data, like attributes and objectives, for the purpose of clarity and simplicity.

2. Show unit reviews.

Reviews of a unit will be displayed orderly. The users can choose to sort the reviews by their creation time, numbers of “upvotes” and so forth. Each review component consists of ratings and a description, as well as buttons allowing the reviewer to modify or delete the review.

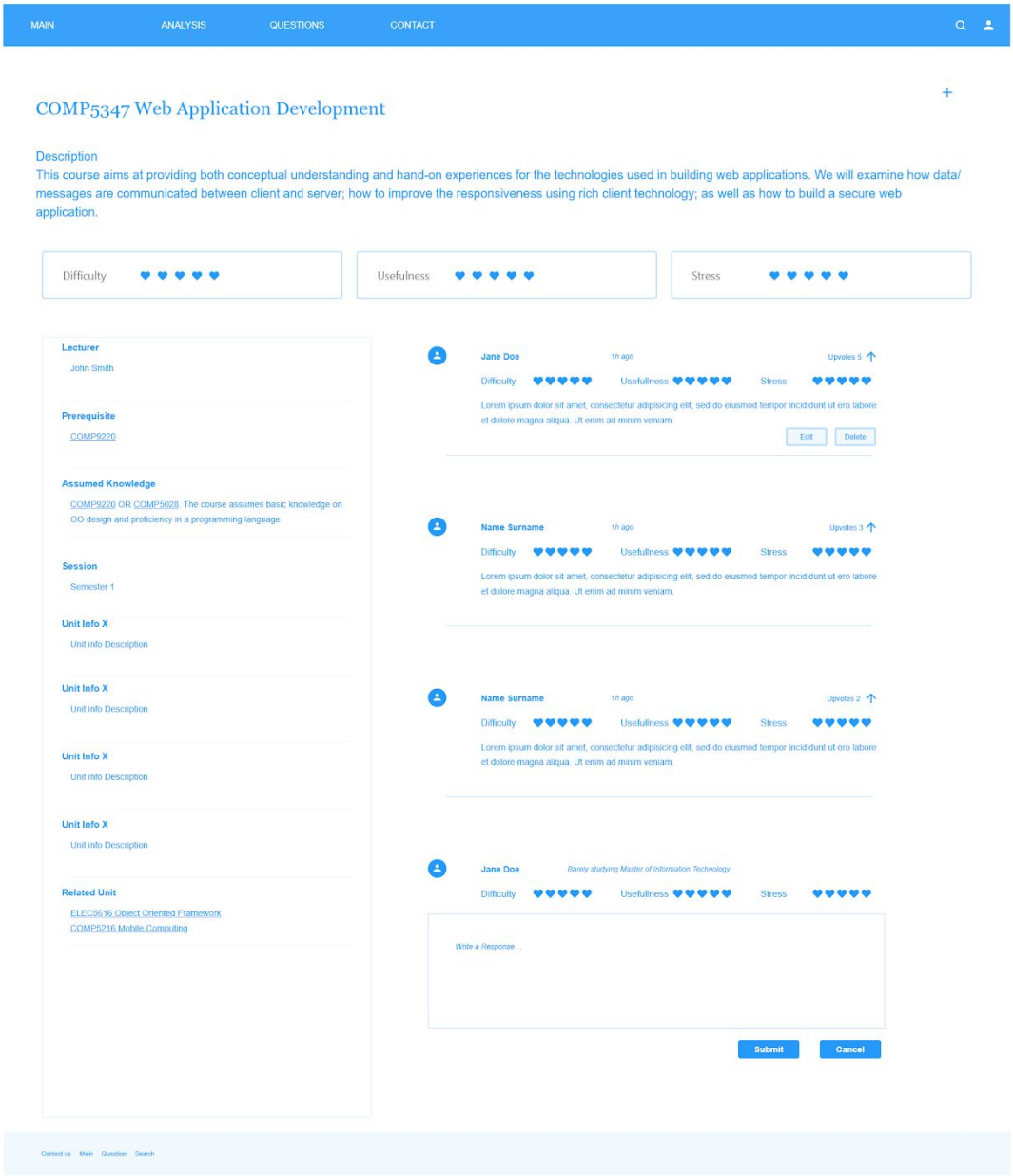
3. Allow users comment on as well as rate the unit

on a scale of 1 to 5, with respect to the following:

- Difficulty: how hard is it to get satisfactory assessment outcomes.
- Stress: measures the workload of the unit with a rating 5 being very stressful.
- Practicality: describes how well a unit prepares the reviewer for job hunting.
  For instance, suppose COMP5347 were difficult, not stressful and very useful, a user would rate it 5, 3 and 5 respectively.

4. Allow users to edit and delete their reviews.

On the page, users can only modify and delete their own reviews. In other words, “Edit” and “Delete” button of a review will not be visible to the current user, if the review was made by other users.


MAIN
ANALYSIS
QUESTIONS
CONTACT
COMP5347 Web Application Development
Description
This course aims at providing both conceptual understanding and hand-on experiences for the technologies used in building web applications. We will examine how data/ messages are communicated between client and server; how to improve the responsiveness using rich client technology; as well as how to build a secure web application.
Difficulty
Usefulness
Stress
Lecturer
John Smith
Prerequisite
COMP9220
Assumed Knowledge
COMP9220 OR COMP5028. The course assumes basic knowledge on OO design and proficiency in a programming language
Session
Semester 1
Unit Info X
Unit info Description
Unit Info X
Unit info Description
Unit Info X
Unit info Description
Unit Info X
Unit info Description
Related Unit
ELEC5616 Object Oriented Framework
COMP5216 Mobile Computing
Jane Doe
1h ago
Upvotes 5
Difficulty
Usefullness
Stress
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut ero labore et dolore magna aliqua. Ut enim ad minim veniam.
Edit
Delete
Name Surname
1h ago
Upvotes 3
Difficulty
Usefullness
Stress
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut ero labore et dolore magna aliqua. Ut enim ad minim veniam.
Name Surname
1h ago
Upvotes 2
Difficulty
Usefullness
Stress
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut ero labore et dolore magna aliqua. Ut enim ad minim veniam.
Jane Doe
Barely studying Master of Information Technology
Difficulty
Usefullness
Stress
Write a Response ...
Submit
Cancel
Contact us
Main
Question
Search

Lecturer Display & Review (Rufeng Tu)

1. Display Lecturer Information

1.1 User story:
I as a student who are interested in a certain lecturer, want to know how is the lecturer is rated by others, so that I could avoid irresponsible lecturer and choose the one that suits me best.
I as a students who took an lecturer, want to post my views about the lecturer, so that other students could get my help.

1.2 Features Description
The lecturers in SIT could be displayed when user viewing the info page of the selected lecturer. The information of the lecturers includes:

- Avatar
- Lecturer's Name
- Nationality
- Teaching History
- Email
- Overall Teaching Ability
- Overall Responsiveness
- Reviews on the Lecturer

Among which, the Overall Teaching Ability and Overall Responsiveness are calculated from the reviews by the user to the lecturer. The Reviews on the Lecturer is collected from the Reviews submitted by other students.

2. Submitting Reviews on the Lecturer

2.1 User Story

I as a student who knows a certain lecturer, would like to give rating and some comments to the lecturer, so that I can help other students to know about this lecture.

2.2 Feature Description

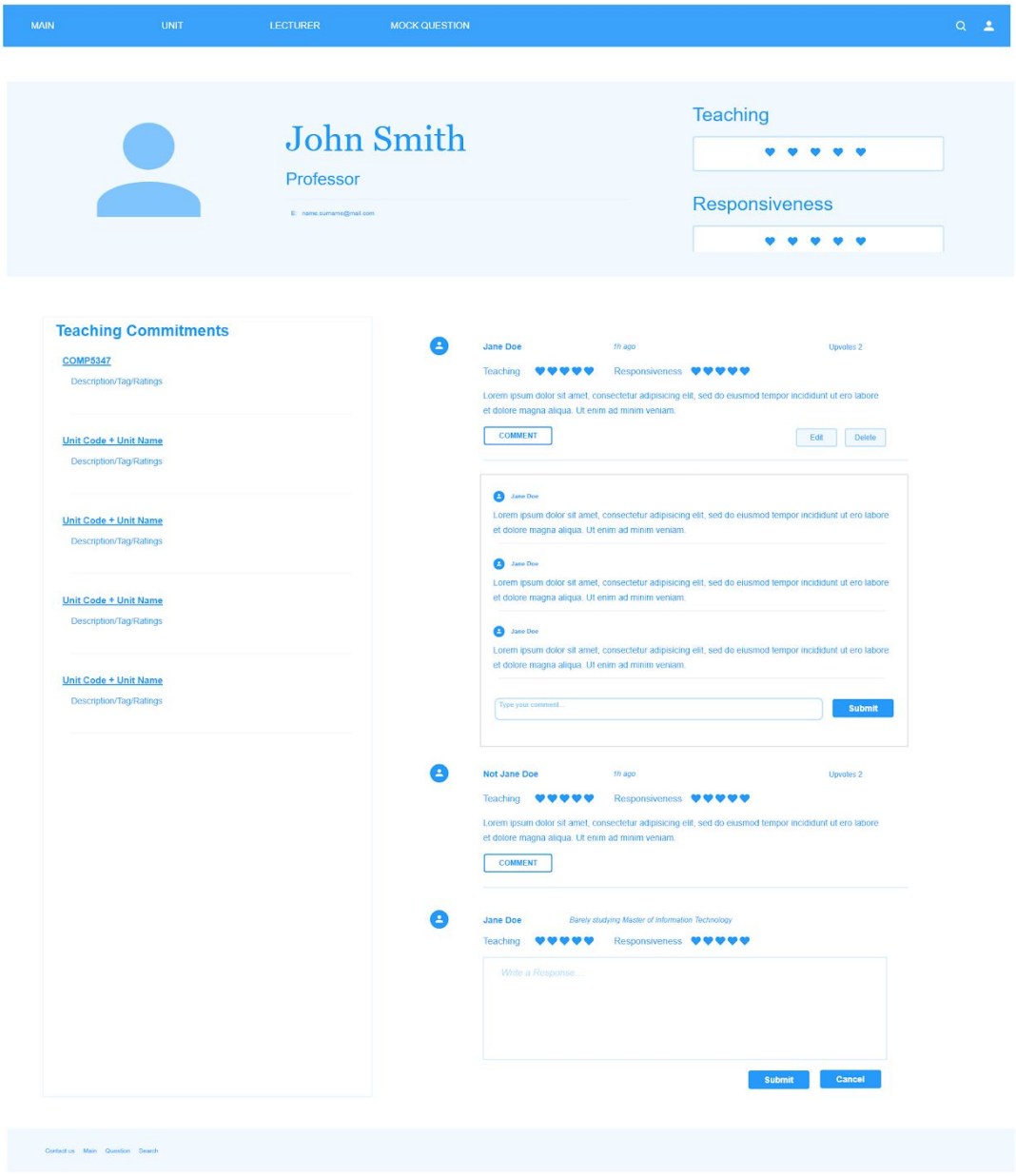
The user in SIT who knows the lecturer could submit reviews about this lecture. The reviews includes:

- Year of the Review
- Teaching Ability Rating
- Responsiveness Rating
- Comments

Among which, the Teaching Ability and Responsiveness will contribute to the overall rating of this lecturer. And please be noticed that the year of review is not the same as the time of submission.

3. Mockup

This mockup is the page to display the information of the lecturer and submit reviews. It includes the elements mentioned in the 1.2 section and the subission area mentioned in 2.2 section.

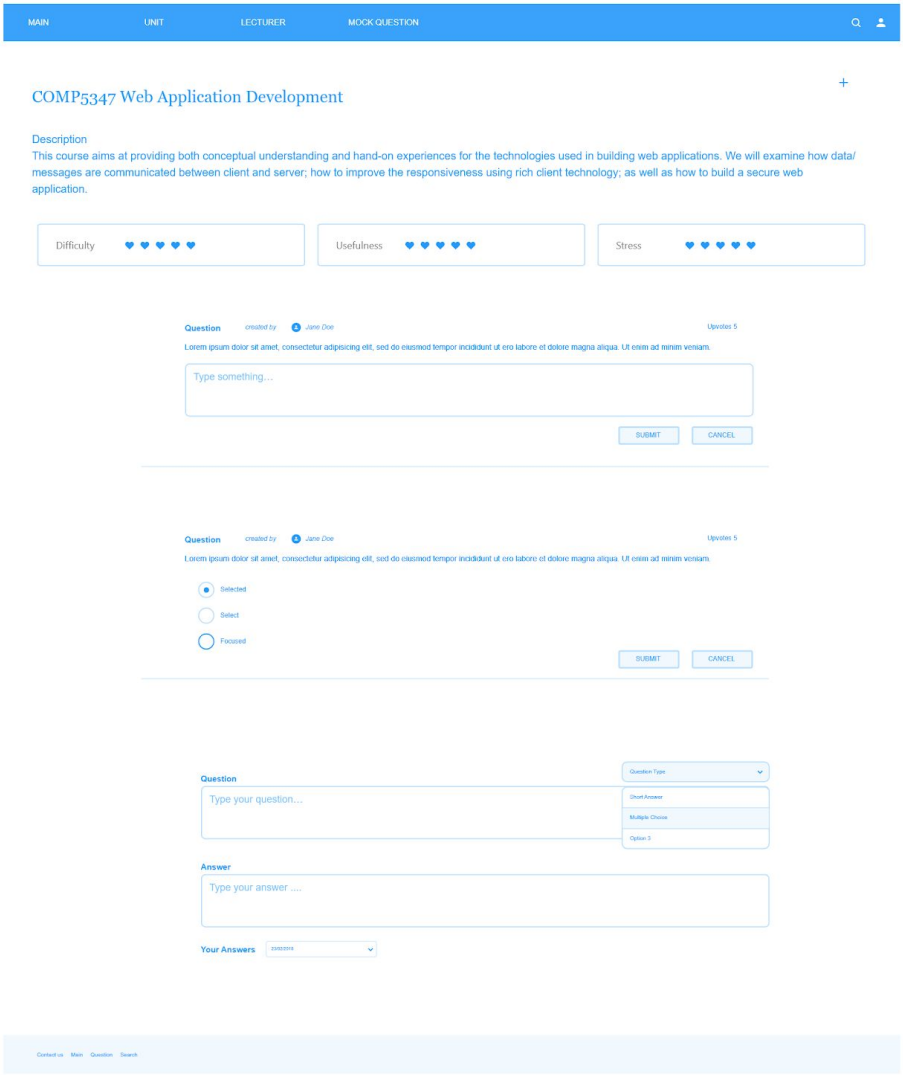


Reviewing questions(Ziqi Wang)

Features:

1. Reviewing question creation

When users are learning a course, he can create question that he thinks is useful for future use. Either he thinks it will helps him to understand the content of the course or the question may occur in a future exam, he can use our website to create a list of questions and save them to the database. Also he can decide whether to share these questions with other people. The question type that can be created using our tools includes single choice, multiple choice, short answer and analysis question. After putting in the question you also need to provide a sample answer for the question. A more clear demonstration can be found in our wireframe.

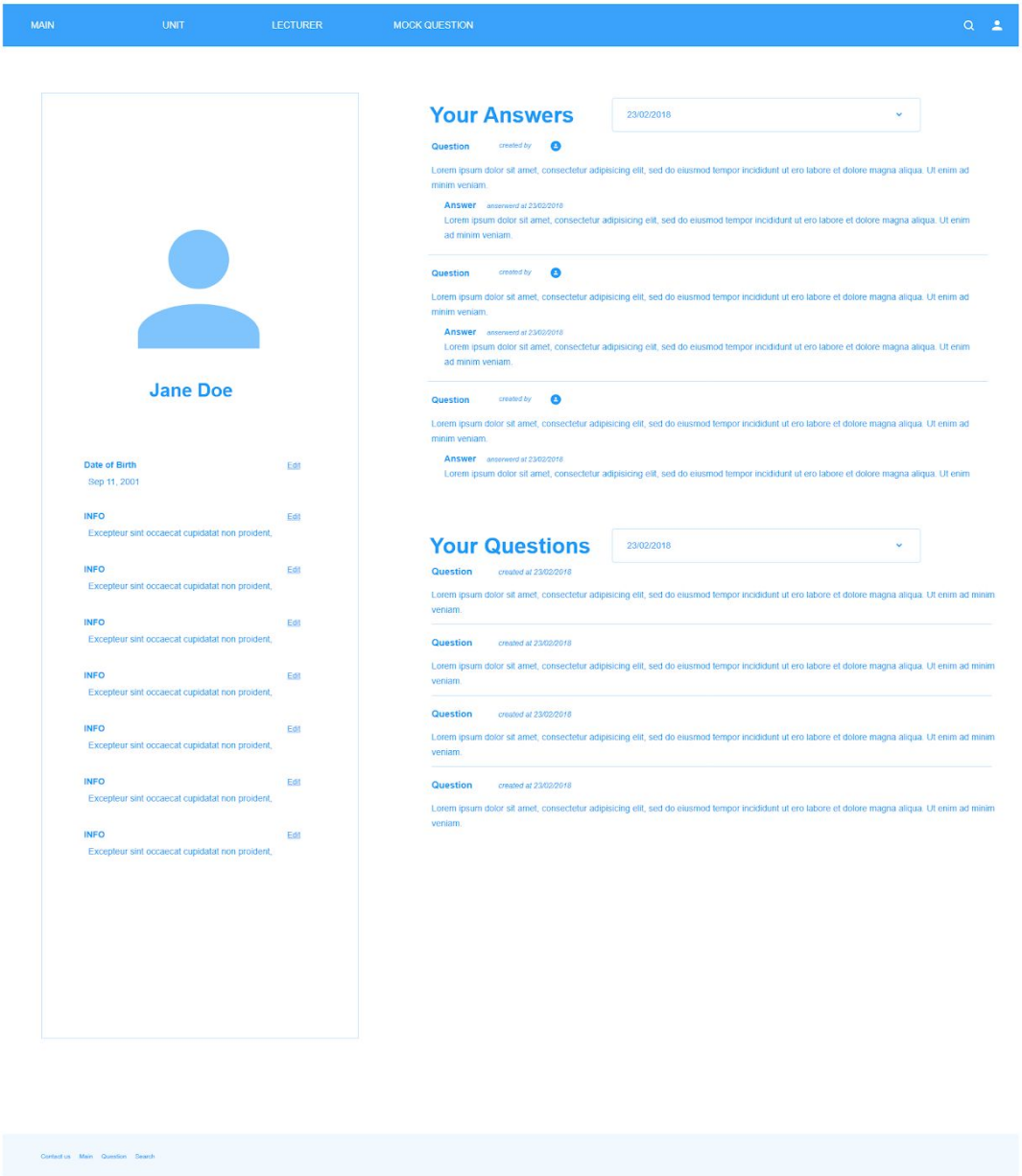


2. Taking quiz

As mentioned before, users are allowed to create questions for a course. Also they can take quiz consisting of the questions created by themselves or from our public database. You can compare your answer to the sample answer as benchmark. Besides that you can go to our discussion board to find more opinions from others for the question if you think the sample answer is questionable.

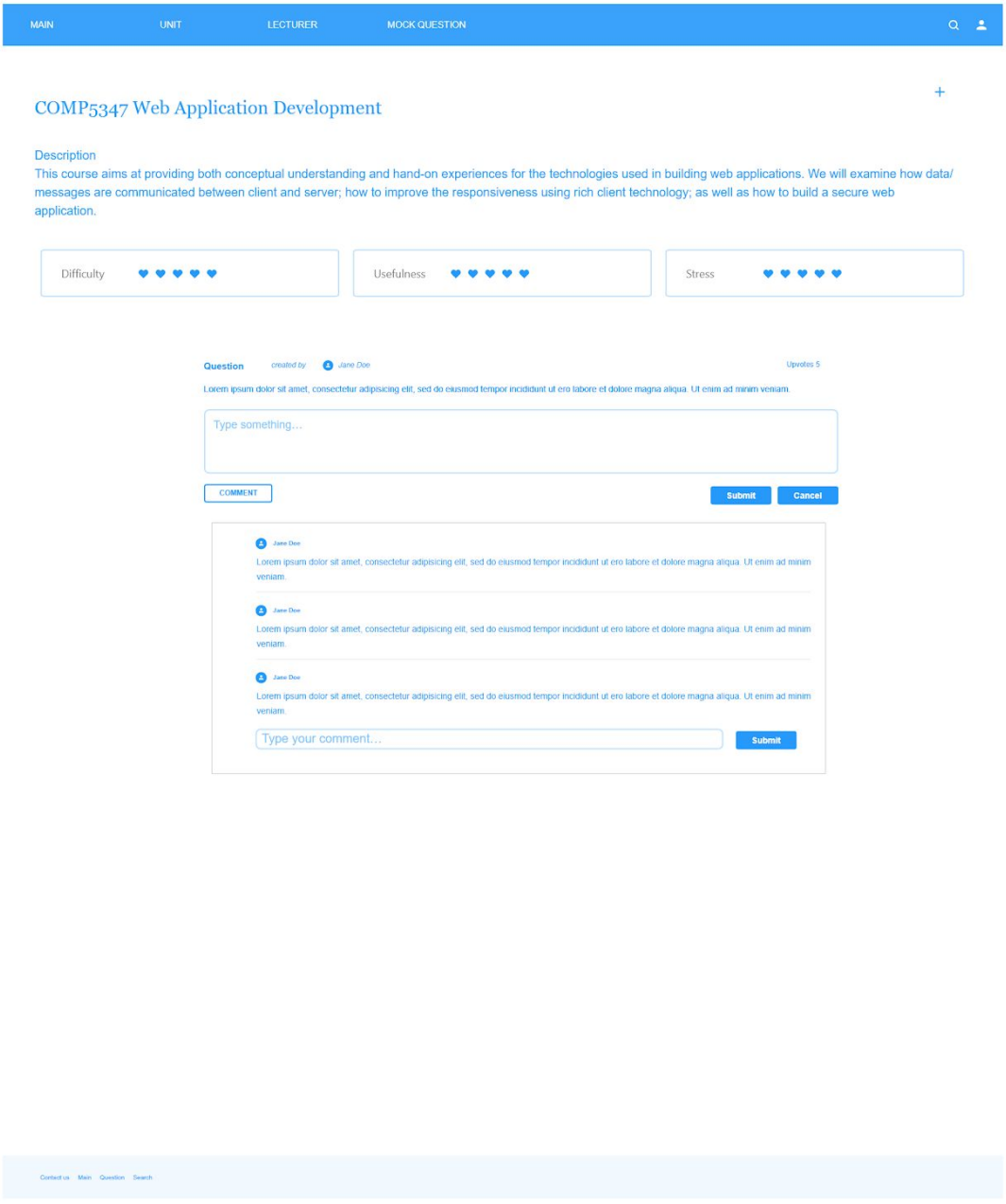
3. Viewing your questions

After a user create his own questions. He is allowed to view the list of the questions he created. Also he is allowed to change the detail of the questions and the sample answers.



4. Question discussion board

For each question in public database, there will be a discussion board for it. People can publish their opinions in this area and give thumb up or thumb down to the sample answer, and one user can only give one from either thumb up or thumb down. And future users can take the thumb up and thumb down number as a reference to see whether the sample answer is good enough.



7. Conclusion

In this proposal, the demands and feasibility of our project CUSP Crachser were discussed and the schedule, implementation plans and component distribution were elaborated. Some proper quality management schemes were applied in the illustration of mockup, data model and system architecture.

To conclude, this proposed project is unique and feasible. The deliverables could help to achieve our aims. It could be suggested that the our group (Group 24) should continue this project with the proposed plans.

Reference

[1] https://studentvip.com.au/usyd/subjects