MCI project First Milestone Report

Team number: 07
Project Title: Curriculum mapping tool

Milestone 1	Activities	Planned Outputs	Achieved Outputs
Complete the website's homepage based on the UI/UX design and link the homepage to the degree structure page. • The homepage will display the titles of the six Computer Science degrees*. • By clicking one of the titles, the browser will be directed to a degree structure page. • The degree structure page can be a blank web page.	Collect user stories and create requirement documentation.	Requirement documentation will be created. It will list the detailed client requirements.	Same as planned.
	Learn essential skills for web development.	Each member will obtain essential skills, including HTML, CSS, JavaScript, Node.js, Express, and MySQL.	Same as planned.
	Design web application architecture to describe the client layer, the API layer, the application layer, and the database layer, showing their relationships, and how they will interact with each other.	A diagram of software architecture design will be created. The diagram will indicate the major services provided by our website.	Same as planned.
	UI/UX design for a homepage, a degree structure page, and a course relationships page^.	UI/UX design for these three web pages will be created. This design will determine the appearance of our website.	The designs of the homepage and the degree structure page are achieved as planned. But since our front-end members got COVID-19 and took sick leave for one week, we have not yet finished the UI/UX designs for the course relationships page.
	Design a database to specify what data will be required and in what structure they will be stored in the database.	A diagram of database design will be created. The diagram will contain two or more tables and show the relationship between tables.	Same as planned.
	Collect data of degree structure and course.	A CSV file will be created. The CSV file will contain information about degree titles, core courses and elective courses of each degree, pre-requisite courses and incompatible courses of each course.	Same as planned. We created three CSV files to store data.
	Establish a database to store the collected data.	A database file will be created by using the data in the CSV file. The database will store the data in a way that is specified in the database design.	Same as planned.
	Set up APIs to request degree titles from the database.	An app.js file will be created. The app.js file will contain APIs for getting data from the database.	Instead of using APIs, we took a simpler approach to query database. We imported an NPM package called 'mysql' to create a connection between our server and database. This package also provided functions to query data from database, which saved us much development effort.
	Create a server that can process the degree titles, making them suitable for browser use.	A server and some new functions will be created in app.js. These new functions will be able to process data.	Same as planned.

	Add new APIs to render degree titles to the browser.	New APIs will be added to app.js file so that app.js can render data to the browser.	Same as planned.
	Create our website's homepage in the browser, showing the degree titles as specified by the UI/UX design.	An index.ejs file will be created, serving as the website's homepage. Some CSS files and JavaScript files will be created to realize the UI/UX design of the homepage.	Same as planned.
	Set up a link from the homepage to the degree structure page.	When the client clicks a degree title on the homepage, the client will be directed to a degree structure page. Also, a new API will be added in the app.js file, which will render a degree structure page to the browser.	Same as planned.

*According to the client requirement, the six Computer Science degrees are: Bachelor of Computer Science, Bachelor of Information Technology, Master of Computer Science, Master of Computing and Innovation, Master of Cyber Security, Master of Artificial Intelligence and Machine Learning.

^A homepage will display the title of each academic degree. A degree structure page will display the core courses and elective courses of each academic degree. A course relationship page will display the pre-requisite courses and incompatible courses of each course.

Team reflection on progress	Provide some comments below regarding the completion of this milestone specifically around: 1. How is the project progressing? 2. Are there any differences between projected and actual outputs

1. Project Progress

- Data operationality. Crawling raw course data from course website, we have established a database and stored all the needed data for our use.
- Architecture design. After a few weeks of studying and analyzing, we generated the architecture diagram of our application based on our deepened understanding of Web-Application.
- Documentation. Up to the Milestone 1, we have documented more than 120 files including the requirements, GitHub task cards, meeting minutes etc. to provide a clear and traceable development process.
- Internal organization. The team internal meetings i.e., sprint planning, iteration review and other standardized Agile approaches have been executed to align with the goal of quality delivery to our client.

2. Differences and Revised Action

However, a few project activities we planned fall short. Mainly, the design of Web Page and database.

- Although finished the homepage before Milestone 1, the front-end team omitted the designing and discussion process to catch up the project timeline for the developing was behind the schedule.
- Same problem exists in the design of database as well, the back-end designing decisions couldn't be made early due to the lag of the web page developing.
- We are trying to avoid same mistakes by efforts listed below:
    - Iteratively posting the design draft.
    - Enhance the communication among members. When a group member is developing his/her own features, instead of going silent assumption, he/she also needs to list possible requirements that needed other members to implement.

Team reflection on managing problems	Have you encountered any problems to date? If so, how have you managed them?

Problem 1: Two front-end team members got Covid-19 and took sick leave, resulting in the team being one week behind schedule.
Solutions:

- As soon as the two members got back to normal development, we first prioritized some of the tasks. After the group discussion, we decided to focus on the Homepage and degree structure page. In that case, we successfully delivered the products as we promised in the Milestone 1 plan.
- We realized that it is necessary to have risk management. We listed potential risks and analyzed their types, probabilities, effects, and corresponding strategies. If any problem occurs during the development phase, we can respond quickly and reduce the severity of the problem.
- To catch up with the schedule, we voluntarily took the second week of the mid-break as a normal working week in sprint 4.

In summary, with the collaborative effort of the whole team, our achieved outputs are same as planned outputs.

Problem 2: As the project moved forward, an instant messenger alone could not meet our demand for effective communication, clear work breakdown, and team collaboration.
Solutions:

- We take Discord as our main communication tool. It has strong functions to support us to have our online scrum meeting. We also set up several channels according to their usage, such as assignment channel, front-end channel and back-end channel etc. We also share the knowledge we learnt and update our daily scrum with other team members through Discord.
- We use GitHub as a project management tool to assign tasks to members in every sprint. At the beginning of every sprint, the scrum master confirms the to-do tasks in that sprint. And each member is allocated the tasks that need to be completed. During the phase of development, we push our code to our own branch before merging it into the master branch, which can prevent the code from crashing.
- We also utilize OneDrive for file management. Editing files on GitHub is not convenient. With OneDrive by file sharing, multiple members can simultaneously work on the same file, which can improve the efficiency of the whole team.

To conclude, instead of using one messaging tool, we have Discord as our communication tool in both oral communications and written communication. And we use GitHub as our code hosting platform for version control and collaboration. Besides, we share a well-organized OneDrive which contains the files that we need.

Supervisor assessment	Please, rate your team (1) effort, (2) project progress and (3) their self-reflection for milestone 1 Rating scale 1-10 as per standard marking scheme, ie 5 is a Pass and 7 is a credit. Add some comments to explain your rating
Effort: Progress: Reflection:	