Sprint 1

Individual Contribution

Back-end Team

Contribution - Boyang Sun

Contribution - Chongjing Zhang

Contribution - Pin Wang

Contribution - Jinzhe Shan

Contribution - Fu Xie

Contribution - Pin Wang

Contribution - Zisheng Cheng

Contribution - Haoyu Qin

Front-end Team

Contribution - Fengrui Zhang

Contribution - Jingyu Li

Contribution -Sai Zhang

Contribution - Sarah Sultan Alyahya

Contribution - Yuhang Xie

Contribution - Zixin Ye

Contribution - Boyang Sun

Act as back-end team leader

- Design the total architecture of the back-end software system, deliver it to the teammates
- Conduct the back-end team weekly meetings and small discussions with teammates who are in charge of different modules
- Workload allocation for the team members
- Define the sprint goal and review for sprint 1
- Finish the sprint 1 assignment submission
- Meeting with the client as back-end team leader and show the current progress

Act as a back-end team member

- In charge of API "Import projects in batch" and "get confluence individual contribution"
    - Import projects in batch API finished
    - get confluence individual contribution 70% progress
- Design the related databases
- Test the two APIs and give out acceptance test output

Contribution - Chongjing Zhang

- Design git_commit_counts table and map it into django model
- Implement Git Commit Count by Date API
    - Invalid url
    - Relation table contains url while git commit table doesn't (the first crawler, acquire all data and sort, count and store them by date. If url is wrong, it will raise "git log excption")
    - Git commit table have data, return them
- Implement a utils function to transform unix timestamp
- Update get_commit functions in github_util package
- Finish acceptance test in postman
- Maintain Trello
- Write some document about my tasks in Sprint1 and Sprint0

Contribution - Fengrui Zhang

- **Update for ProductQualityPage**
  - redesign code matrix form to make better use of space
  - add API of code matrix into the project
- **Implement fetching data from local API**
  - set local API using Mockoon
  - show data from local API successfully
- **Update Design Concept document**
  - Update all the pictures
  - Update text descriptions of any changes

Contribution - Fu Xie

- Test some of the code which is done by previous team (Jira part)
- Create table 'url_config', 'jiracountbytime' and 'individualcontributions'
- Implement API - 'Set Github and Jira Url', and store the data into table 'url_config'
- get the API -' Get Jira TO_DO/IN_PROGRESS/DONE Count by Time' connected with table 'jiracountbytime', make the function execute automatically for every 24 hours
- get the API - 'Get Individual Contribution on Jira DONE Count Tickets' connected with table 'individualcontributions'
- Acceptance test for new API (Jira part)

Contribution - Haoyu Qin

- Re-organized legacy JIRA API code
- Re-implemented legacy JIRA API functionalities
- Re-implemented legacy JIRA API unit tests
- Implementation : Get Individual Contribution on Jira DONE Count Tickets (Fetch, Logic)
- Partial Implementation : Get Jira TO_DO/IN_PROGRESS/DONE Count by Time (Fetch, Logic)
- Unit tests of the JIRA APIs

Contribution - Jingdan Cui

1. Implement 2 APIs
    a. Get the Already Imported Project
    b. Get the Meeting Minutes
2. Use Postman to test the APIs above
3. Create a table to store the title and the Confluence link of a meeting minutes in the Database and update the table on a daily basis

Contribution - Jingyu Li

1. **Implement UI for ProjectConfigurePage**
    - set react form to input url
    - set css to make layout properly
2. **Update UI based on new suggestions**
    - modify react form to show history for input section
3. **Implement fetching data from local API with the help of team leader**
    - set local API using Mockoon
    - fetch data from local API

Contribution - Jinzhe Shan

1. Design the table for "Git Comment Count " API
2. Clarify the "Git Comment Count " Requirement with the client and It has changed.
3. Clarify the "Git Authority for Private Repos" Requirement with the client.
4. Clarify the "Git Metrics" Requirement with the client, and I will achieve it in the next sprint.
5. Write Git API and Database Design Documents
6. Update Testing and Deployment Documents.

Contribution - Pin Wang

- Design APIs for sprint 1
- Implement 3 APIs and test them
    - Get Confluence Space
    - Get User List
    - Get Page Count by Time
- Implement a util function to run methods on a regular basis after the server is running
- Implement the database needed by Get User List and Get Page Count by Time, and update them on a daily basis
- Help others to
    - Deploy
    - Migrate
    - Add new Models
    - Make queries on database

Contribution - Ruofan Zhang

1. modified the ui of process quality webpage
2. implemented the data fetching from api for process quality webpage
3. wrote public components and tools for the project
4. reviewed the pull requests
5. set git/github branching guidelines for the team
6. helped teammates to solve their problems encountered during the app development
7. provided testing tools and mock data for the team

Contribution -Sai Zhang

1. Develop Homepage UI according to the design of Prototype

Version 1:

- Implemented the search bar: The purpose is to help users search the project name they want to import into it
- Implemented temporary storage and display of one or more items selected by the customer
- Implemented Import Button: Click "Button" to Import one or more items selected by the customer at one time
- Implemented the display table of imported data: the purpose is to display the project name and Confluence link of the imported project

Version 2: Based on Jirat's suggestions at Workshop,

1. I removed the Import button and the place where I temporarily stored the selected project name.
2. Added delete button to the table showing Imported Project to directly operate on Imported Project

2. The Local API deployment:

- Created mock data in Mockoon tool
- Set Local URL for testing local API
- According to Ruofan's previous work, try to add the code related to the Homepage for setting Local API

3. Arranged client meeting for reporting work progress of Sprint 1

Contribution - Sarah Sultan Alyahya

Preparing work:

Learned:

- Read and understand the code of the previous team
- Learned how to use react, node.js, react-redux
- Learned about HTML, JavaScript and CSS
- Learned how to use Visual Studio Code
- Learned how to use Github and how to create pull-request

Set up:

- The local development environment with VS Code.

Develop UI:

- Used the existing style of the website for the page, such as the heading, banner
- Used doughnut chart to depict the data, where the existing recharts library was used to perform the task. Although the legends for the graph were shown on the right for better readability
- Used three-button (from the existing implementation) to show the confluence, git and Jira data
- For each tool, data for all students as well as each student can be filtered
- For the purpose of showing student filter, a drop-down was used, whereupon selecting the relevant student, the data was populated for the corresponding student

Develop Local API:

- Local API was implemented and deployed to get the data for individual contributions for each tool and the data from the API was formatted to suit the doughnut chart data format
- In order to implement the local APIs, the user.service, the user.actions, user.constants and user.render files were changed including the individualContributionPage

Confluence page:

- Create Sprint 1 in 8 Sprint Summaries
- Create Sprint 1 Documentation and update daily
- Reorganise 10 Meeting page

Contribution - Yuhang Xie

1. **Do the preparing work for frontend work of the project**
    a. Reviewed HTML, JavaScript, CSS
    b. Learned React, node.js, react-redux
    c. Learned the code from previous team
2. **Development the UI design of the project homepage**
    a. student name
    b. student ID
    c. student profile
    d. student email address
    e. page banner
    f. page header
3. **Development the Mockoon local API**
    a. modified user constant file
    b. modified user reducer file
    c. modified user action file
    d. modified user service file
    e. modified the data input method in project homepage

Contribution - Zisheng Cheng

1.Design student_commit_counts table and map it into django model

2.Implement Git Individual Commit Count API

3.Complete the task of regularly grabbing data from github and storing it in the database

4.Finish acceptance test in postman

5.Write some document about my tasks in Sprint1 and Sprint0

Contribution - Zixin Ye

My efforts in Sprint 1:

1. Finish UI of communication page.
2. Finish all functions required to interact with backend.
3. Contribute some efforts to the development of the API functions in coordinator home page.

Review of Sprint 1 - Back-end Team

Coding Update

Confluence

Import Projects in Batch

- Pull Request: https://github.com/pete965/COMP90082_Software_Project_Database_Backend/pull/13/files
- Added Functions:
    - import_project_in_batch()
    - import_project()
- Edited Functions: No implementation in the older system.

Get Meeting Minutes

- Pull Request:
- Added Functions:
- Edited Functions:

Git

Get Individual Commit Count

- Pull Request: https://github.com/pete965/COMP90082_Software_Project_Database_Backend/pull/5
- Added Functions:
    - class StudentCommitCounts(models.Model)
    - update_individual_commits
- Edited Functions:
    - get_git_commits
    - test_get_individual_commits

Get Commit Count by Date

- Pull Request: https://github.com/pete965/COMP90082_Software_Project_Database_Backend/pull/5
- Added Functions:
    - class GitCommitCounts(models.Model)
    - transformTimestamp
- Edited Functions:
    - get_git_commits
    - get_commits
    - get_git_pr
    - test_git_date
    - test_get_total_commits

Design Update

System Design Update

Get Individual Contribution on Confluence Pages

- Older design: Older design is when you call this API, it will get the data from confluence RESTFUL API and it usually takes a few seconds.
- New design: We will retrieve the data from the confluence API every 24 hours and store it in the database. Each time the API is called, we give out the output based on the data in the database. This design enhanced the total efficiency and can react in a few milliseconds

Import Projects In Batch

- Older design: No design and implementation for this API in older software.
- New design: API receives the coordinator name and the project list. Then this information will be added into our database one by one, please refer to the database design: Database Design

Get Imported Project

- Older design: The older version calls the Atlassian Confluence API and gets all the Confluence space's data.
- New design: The title and name of a Confluence space is retrieved from the database.

Get Meeting Minutes

- Older design: No design and implementation for this API in the older version.
- New design: The title and weblink of a meeting minute is retrieved from the database. The system calls the Confluence API every 24 hours to get the meeting minutes information of the Confluence space, and the meeting minutes data will be stored and updated n the database.

Get Git Commit by Date

- Older design: Use POST method to require date from github and it usually takes some time.
- New design: Use GET method now, and we will retrieve the data from the confluence API every 24 hours and store it in the database. Each time the API is called, we give out the output based on the data in the database. This design enhanced the total efficiency and can react in a few milliseconds. If there are not data in db, it will try to get url in relation table to access git url to crawl data and then store. If it is invalid url or space key doesn't exist in relation table, it will raise specific exception. DB design: Database Design

Requirements Update

Get user list

If the user count is more than 50, we need to display that there is too much users.

Get Meeting Minutes

There will be a warning shown on the system if the keywords of all page titles do not include "**meeting**".

Get Git Individual Commit Count

Update to use GET method, and improve logic to use db data first, if there are not, crawl data from github. It will greatly speed up access. Also set a timed 24h crawler to acquire data to ensure real-time property.

Get Git Commit Count by Date

Update to use GET method, and improve logic to use db data first, if there are not, crawl data from github. It will greatly speed up access. Also set a timed 24h crawler to acquire data to ensure real-time property.

Get Git Comment Count

This requirement has canceled after communicating with the client.

Get Git Metrics

By clarifying with the client, we need to clone the git repo and then do code analysis such as code lines, comment lines, the number of functions, the number of classes, and so on.

Task Achieved in Sprint 1

Confluence

Import Projects in Batch

- Previous system: There are no importing projects functionalities and no relationship between coordinator and projects
- Our work: Design and finish the implementation of this API and functionality. User can now import projects(such as COMP90082SP and COMP90092) in batch
- Related User Story: 1.5 User stories US - 1

Github

Get Commit Counts by Date

- Previous system: In the previous design, the api obtains data by directly entering the corresponding Github link and returned it
- Our work: Using GET method to replace POST. Design the corresponding database. The user obtains the corresponding data by space_key. DB data will update every 24h to make sure real-time by crawler. Get data from the database first, if not, the crawler will access github to get the data and store it in the database.
- Related User Story: 1.5 User stories US - 9

Get Individual Commit Counts

- Previous system: In the previous design, the api obtains data by directly entering the corresponding Github link and returned it
- Our work: Using GET method to replace POST. Design the corresponding database. The user obtains the corresponding data by space_key. Write the part of the data previously obtained from Github into the timing task.
- Related User Story: 1.5 User stories US - 16

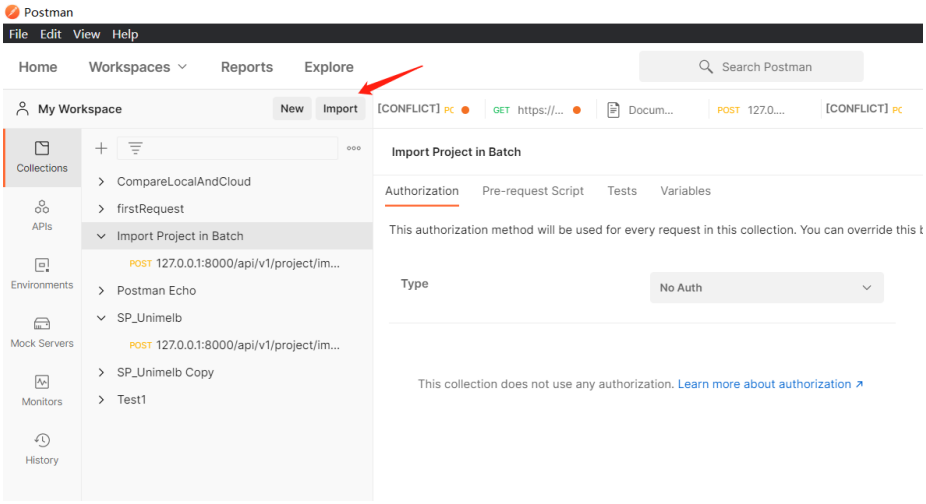
Jira

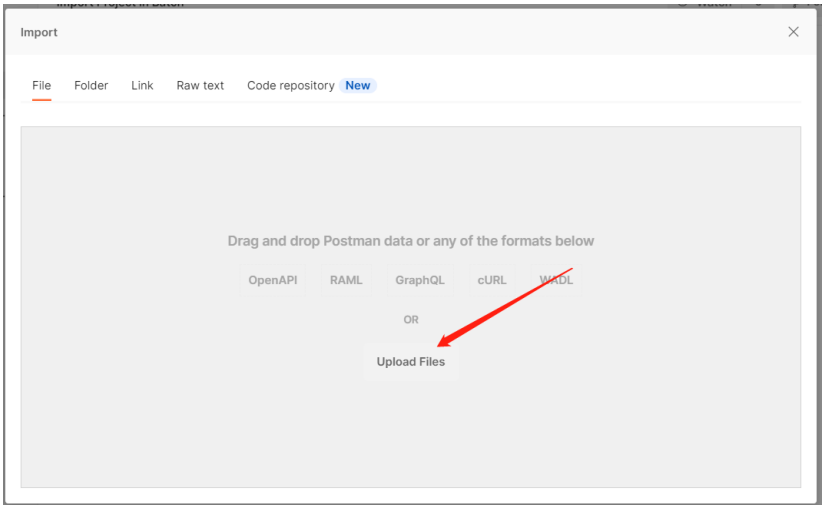
Testing Update

We choose to test the APIs using postman since they can be easily imported and tested with postman clients. We provided postman settings in the confluence.

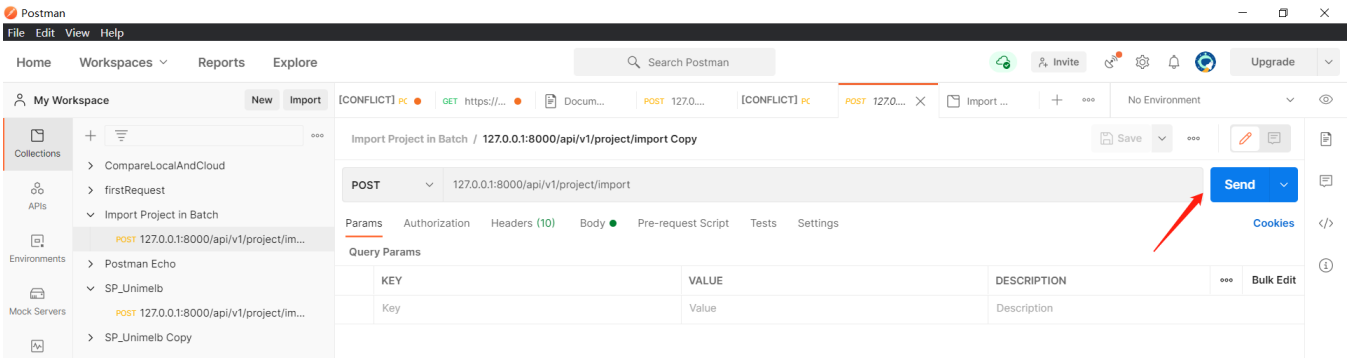
Follow the instructions here before you use the real postman request to test:

- Install postman: https://www.postman.com/downloads/
- Deploy the back-end system: 7 Deployment
- Import the collection below





- Click the send button in the corresponding request



- Finally, you will see the output like these: Acceptance Test

Confluence

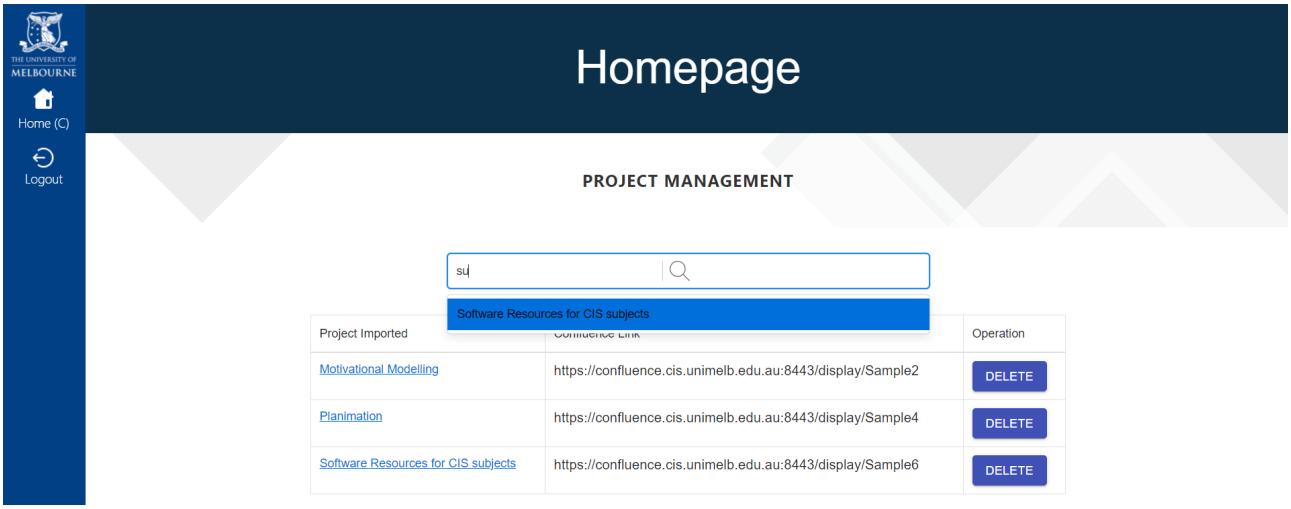
Import Project in Batch

Review of Sprint 1 - Front-end team

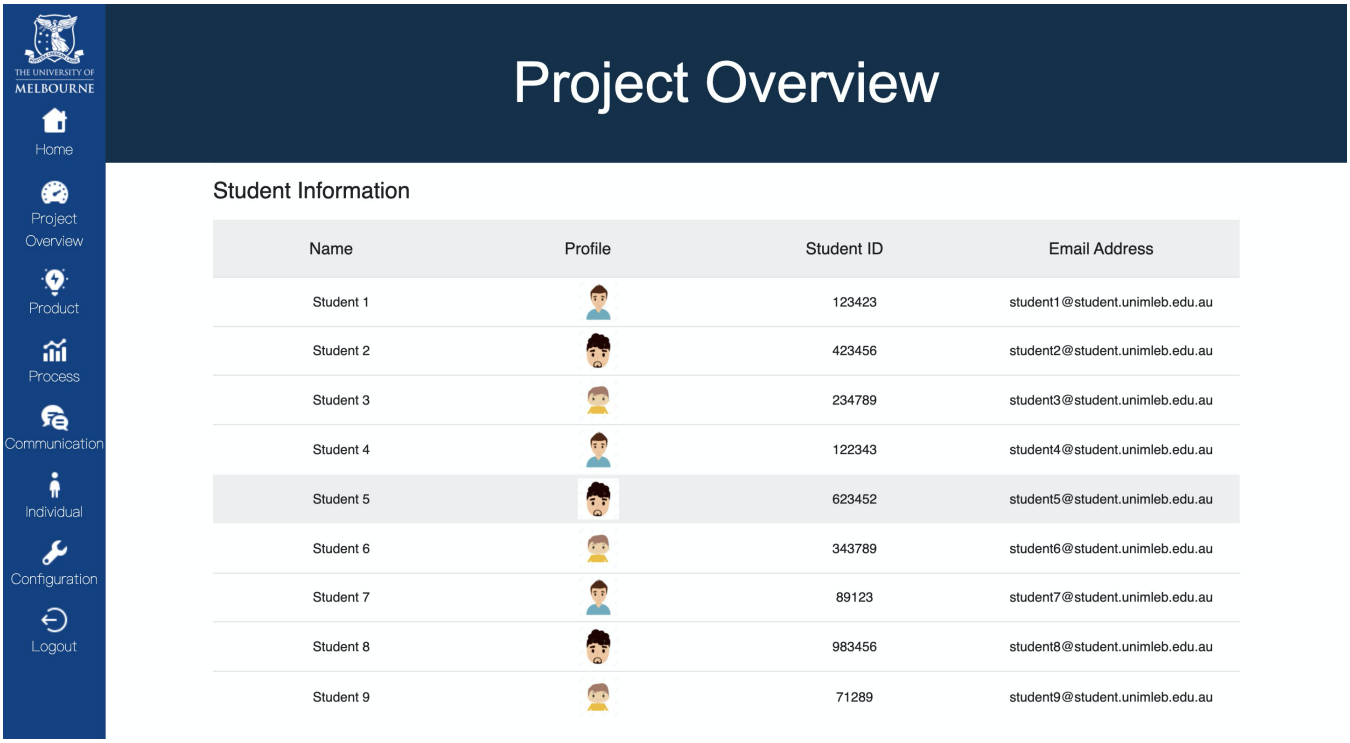
Coding update - Front-end team

Design update - Front-end team

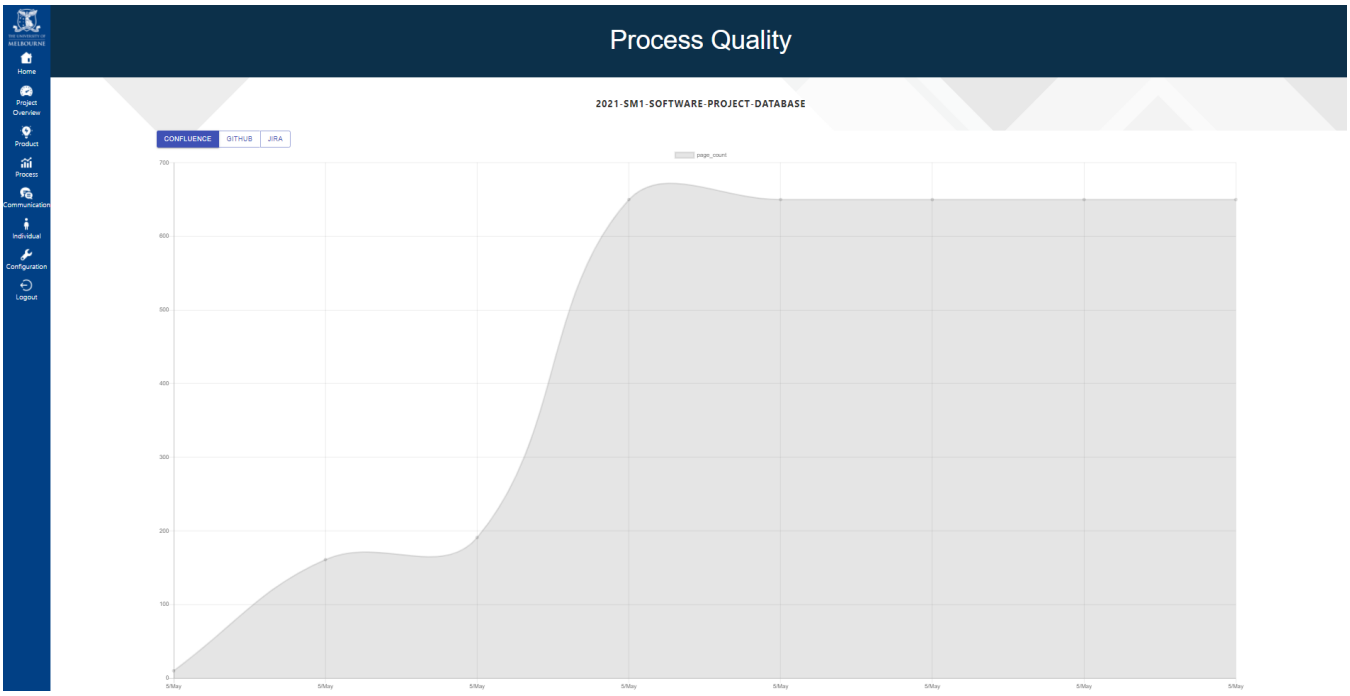
1. Coordinator homepage



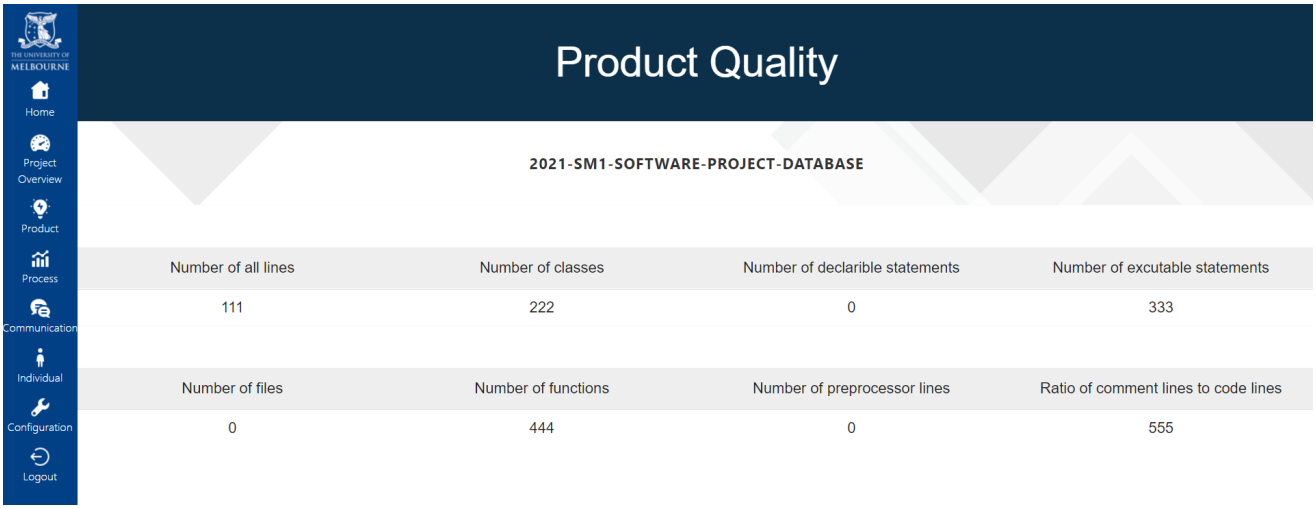
2. Project overview



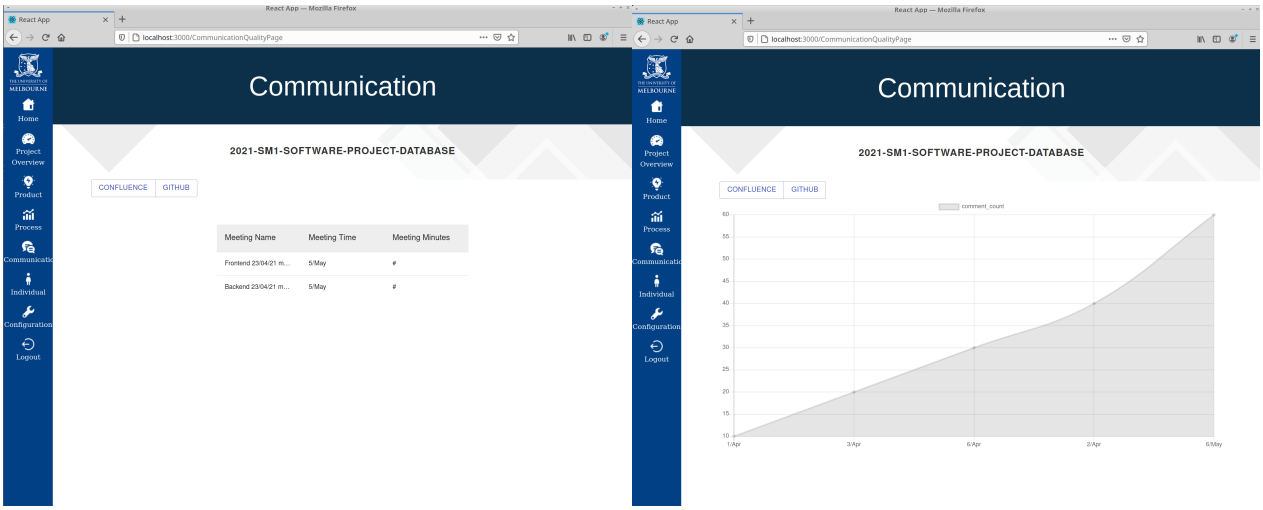
3. Process quality



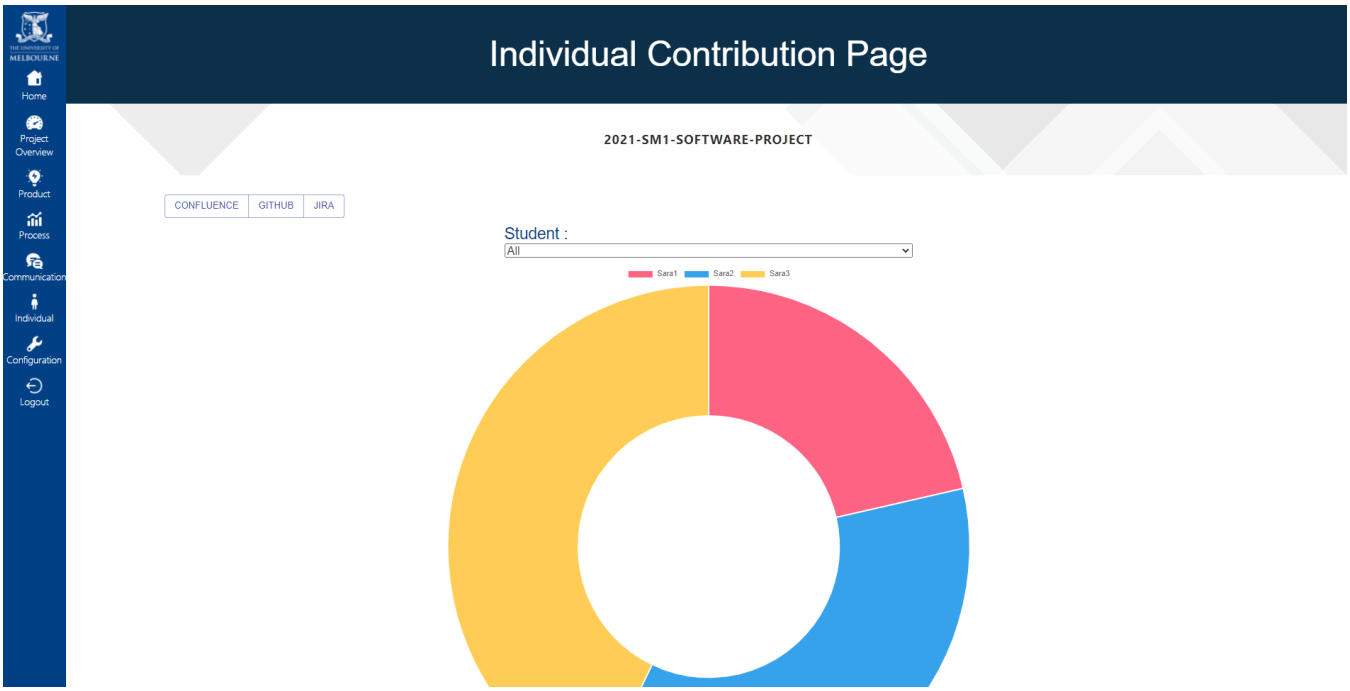
4. Product quality



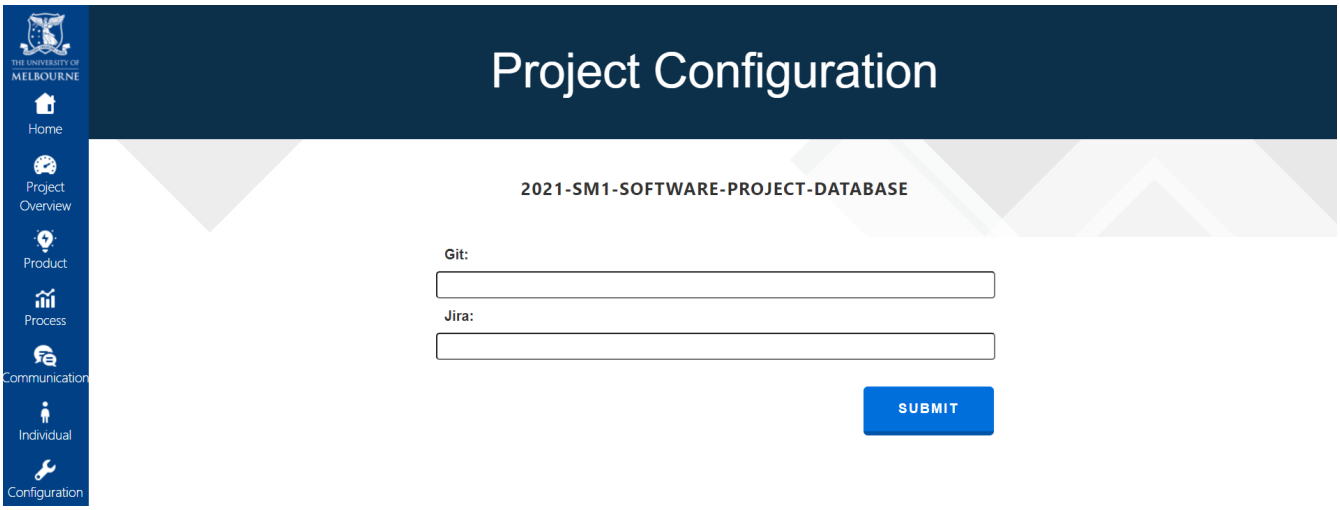
5. Communication



6. Individual page



7. Configure


THE UNIVERSITY OF MELBOURNE
Home
Project Overview
Product
Process
Communication
Individual
Configuration
Project Configuration
2021-SM1-SOFTWARE-PROJECT-DATABASE
Git:
Jira:
SUBMIT

Tasks achieved - Front-end team

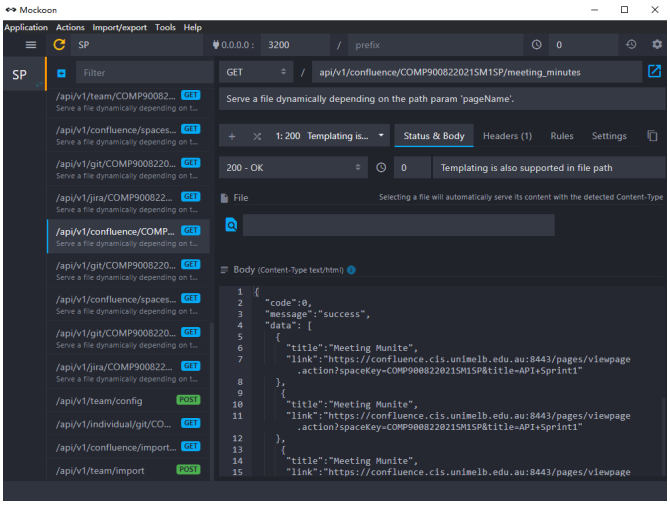
In Sprint 1, our team mainly achieved two tasks:

1. Modified the UI of the previous app according to the prototype created in Sprint 0;
2. Implemented the data fetching and display functionality using Redux and visualization components.

The improvements based on the work done by the previous team can be perceived in Design update - Front-end team.

Testing update - Front-end team

We used Mockoon to test our data fetching code. The mockoon-mock-api.json file can be imported into the Mockoon and anyone can run the mock api on their own machines.



Sprint 1 Ceremonies

During sprint 1, the team have had one meeting with the client, two meetings for a front-end team, two meetings for a back-end team, three stand-up meetings, and three weekly team meetings.

- Stand-up Meeting 15th April 2021
- Front-end Meeting Notes on 19th April
- Back-end Meeting Notes 19th April
- Stand-up Meeting 22nd April 2021
- Back-end Meeting Notes 24th April
- Front-end Meeting Notes on 26th April
- Stand-up Meeting 29th April 2021
- Meeting with Client #3

Stand-up Meeting 15th April 2021

Date:

2021-4-15

Time:

14:15 - 15:15

Attendees:

- Jirat Pasuksmit
- Zhang Sai
- Jingyu Li
- Sarah Sultan
- Yuhang Xie
- Zixin Ye
- Fengrui Zhang
- Zisheng Cheng
- Jingdan Cui
- Haoyu Qin
- Pin Wang
- Fu Xie
- Chongjing Zhang

Goals:

- Discuss what each team member contributed last week and what they plan to do next

Discussion items:

Items	who	Notes
Demonstrate the project process	Each Team Member	• The tasks the team have done in last week • The tasks the team plans to finish next week
Give feedback and Suggestions	Jirat Pasuksmit	• the content on Trello should be more specific • remember to do the Peer Review • assign a person to do the code review will be good • mockup data • provide protocol data to the Backend team

Tasks completion in the Inception phase:

Who	What have you done	What you plan to do next
Zhang Sai	• Set up local development environment for developing homepage UI and run • learn react knowledge • started to develop some feature via react in VSC	• try to complete homepage UI development • learn Mockoon tool
Ruofan Zhang	• read the previous code and learn how to use react framework	• modify the ui of process quality webpage and offer public components for the team

Jingyu Li	• Set up the local development environment with VSCode and install node.js • run the project in the local machine • learn React	• continue learning React • Doing UI for Project Configure Page
Sarah Sultan	• Set up the local dev environment with VS Code and run the front-end project on my machine.	• Figure out the project and learn how to use the React framework and JavaScript syntax.
Yuhang Xie	▪ confirmed the tasks need to be done in Sprint 1 ▪ Review the code from last team	▪ Review HTML, JavaScript and CSS ▪ Learn react, node.js
Zixin Ye		
Fengrui Zhang		
Boyang Sun		
Zisheng Cheng	• Do the deployment • read code	• Test the data base related function • Try to write some features
Jingdan Cui	• Do the deployment • Find some problems, because of the directory structure, during deployment and report them to the backend team	• Try to finish some functions before next week meeting
Haoyu Qin		
Jinzhe Shan		
Pin Wang	• confirm new APIs with front-end	• Finalize API document
Fu Xie		
Chongjing Zhang		

Stand-up Meeting 22nd April 2021

Date:

2021-4-22

Time:

14:15 - 15:15

Attendees:

- All team members
- Jirat Pasuksmit

Goals:

- Discuss what each team member contributed last week and what they plan to do next

Discussion Items:

Items	who	Notes
Demonstrate the project process	Each Team Member	• The tasks the team have done in last week • The tasks the team plans to finish next week
Give feedback and Suggestions	Jirat Pasuksmit	• the Backend team should speed up the progress

Tasks completion in the Inception phase:

Who	What have you done	What you plan to do next
Zhang Sai	• Show complete Homepage UI based on prototype design • Showed UI data (hardcoding data) • Learn about how to set local API using Mockoon	• Improve UI according to Jirat's suggestions • Try to deployment local API via mock data • Learn redux
Ruofan Zhang	• done the ui modification of process quality webpage • helped teammates to solve problems • offered public components for the team • set git/github branching guidelines	• implement the api data fetching part of the process quality page • offer testing tools and mock data for the team
Jingyu Li	• Complete UI for Project Configure Page based on previous design	• Modify UI for Project Configure Page based on new suggestions to achieve showing history function • Learn redux • Do fetching data from local API
Sarah Sultan	▪ Figure out the project and learn how to use the React framework and JavaScript syntax. ▪ Create Sprint 1 Documentation in confluence.	▪ Deliver the UI of the "Individual Contribution Page" according to the prototype. ▪ Update Sprint 1 Documentation in confluence.
Yuhang Xie	▪ Reviewed html, javascript and css ▪ Learned React, node.js ▪ Finished UI for project homepage	▪ Deploy local Mockoon API to test frontend page ▪ Write test data in json file ▪ Modify API files for project homepage

Zixin Ye		
Fengrui Zhang		
Boyang Sun		
Zisheng Cheng	• Test the api from Github • read related code	• Implement some APIs • complete its test
Jingdan Cui	• Help group members to solve the deployment problem • Communicate with the frontend team and get a better understand of the project • Discuss the task assignment with some members since there are some overlapping tasks.	• Discuss with the backend group about our DB • Talk to the frontend team to ensure the consistency
Haoyu Qin		
Jinzhe Shan		
Pin Wang	• Finish API document	• Implement some APIs
Fu Xie		
Chongjing Zhang		

Stand-up Meeting 29th April 2021

Date:

2021-4-29

Time:

14:15 - 15:30

Attendees:

- All team members
- Jirat Pasuksmit

Goals:

- Discuss what each team member contributed last week and what they plan to do next

Discussion Items:

Items	who	Notes
Demonstrate the project process	Each Team Member	• The tasks the Front-end team have done in last week • The tasks the Back-end team have done in last week • The tasks the team plans to finish next week
Give feedback and Suggestions	Jirat Pasuksmit	• Try to do the integration test before 30th April 2021

Tasks completion in the Inception phase:

Who	What have you done	What you plan to do next
Zhang Sai	• Showed latest Homepage UI based on Jirat's suggestion last week (removed imported button and temporary area for storing selected projects) • Arranged client meeting on 30th April. • Setting part of Local API work	• Improved Homepage UI based on Client's suggestions • Complete API setting with backend
Ruofan Zhang	• Done the api fetching functionality for process quality webpage • offered testing tools and mock data for the team • helped teammates to solve problems	• list tasks to do in the next sprint and assign them to each team member according to their workload in Sprint 1
Jingyu Li	• Finish fetch data from local API with the help of team leader for Project Configure Page	• Add new functions like setting username and password for Project Configure Page based on suggestions from client.
Sarah Sultan	• Deliver the UI of the "Individual Contribution Page" according to the prototype. • Update Sprint 1 Documentation in confluence. • Local API "Individual Contribution Page"	• Test API • Update Sprint 1 Documentation in confluence.

Yuhang Xie	▪ Deployed local API using Mockoon ▪ Make test data in json file ▪ Modify the API files for project homepage	▪ Modify the student profile in table ▪ Optimalise the table size and pop up the student member list.
Zixin Ye	• Complete UI and all functions required to interact with backend	• Adjust any changes according to any changes from backend team/client • Help other people in frontend team out if necessary
Fengrui Zhang		
Boyang Sun		
Zisheng Cheng	• Complete the git APIs and the corresponding database design • Test API	• Update Sprint 1 Documentation
Jingdan Cui	• Complete GET_IMPORTED_PROJECT function • Complete GET_MEETING_MINUTES function • Complete UPDATE_MEETING_MINUTES function • Using POSTMAN to test the functions above	• Update Sprint 1 Documentation • Constantly update existing functions
Haoyu Qin		
Jinzhe Shan		
Pin Wang	• Complete the implementation of 3 APIs and the corresponding database design • Test API	• Modify some business logic.
Fu Xie		
Chongjing Zhang		

Sprint 1 Documentation

Title		Creator	Modified
1 Project Requirement		Boyang Sun	Zixin Ye
2 Design Concepts		Boyang Sun	Fengrui Zhang Fu Xie Yuhang Xie
3 Architecture Design		Boyang Sun	Fu Xie
4 Prototypes		Boyang Sun	Ruofan Zhang
5 Backlogs		Boyang Sun	
	Sprint 1 Backlog	Boyang Sun	Pin Wang Jinzhe Shan Jingdan Cui
6 API documents		Boyang Sun	Pin Wang
	6.5 Jira	Boyang Sun	Pin Wang
	6.6 Git	Boyang Sun	Pin Wang
	6.9 Issues to be Fixed	Boyang Sun	
API Sprint1		Pin Wang	Pin Wang Fu Xie Boyang Sun
	Acceptance Test	Boyang Sun	Jingdan Cui Pin Wang Boyang Sun Jinzhe Shan
	Confluence API Design	Boyang Sun	
	Database Design	Boyang Sun	Boyang Sun Pin Wang Chongjing ZHANG
	Mock API Construction Instructions	Boyang Sun	
	Workload Seperation	Boyang Sun	Boyang Sun
	Data Sample	Jingdan Cui	Jingdan Cui Pin Wang
7 Deployment		Boyang Sun	Jinzhe Shan Pin Wang
8 Sprint Summaries		Boyang Sun	Sarah Sultan A AL YAHYA
Sprint 1		Sarah Sultan A AL YAHYA	
Individual Contribution		Boyang Sun	
Review of Sprint 1 - Back-end Team		Boyang Sun	
	Design Update	Boyang Sun	
	Requirements Update	Boyang Sun	
	Coding Update	Boyang Sun	
	Testing Update	Boyang Sun	
	Task Achieved in Sprint 1	Boyang Sun	
Sprint 1 Ceremonies		Jingdan Cui	
	Stand-up Meeting 15th April 2021	Jingdan Cui	Jingdan Cui
	Stand-up Meeting 22nd April 2021	Jingdan Cui	Jingdan Cui
Sprint 1 Documentation		Sarah Sultan A AL YAHYA	Sarah Sultan A AL YAHYA
9 Testing		Boyang Sun	Jinzhe Shan
10 Meeting		Sarah Sultan A AL YAHYA	
10.1 Meeting Notes for Project Team		Boyang Sun	
	Meeting Notes for 1st, April (Workshop Stand-up Meeting)	Sai Zhang	Sai Zhang Boyang Sun Jinzhe Shan
	Meeting Notes for 6th, April	Boyang Sun	
	Meeting Notes for 13th April	Sai Zhang	Sai Zhang
	Stand-up Meeting 15th April 2021	Jingdan Cui	Jingdan Cui
	Meeting Notes for 15th April	Sai Zhang	Sai Zhang Jingdan Cui
	Stand-up Meeting 22nd April 2021	Jingdan Cui	Jingdan Cui
	Stand-up Meeting 29th April 2021	Jingdan Cui	Jingdan Cui
10.2 Meeting with Clients		Boyang Sun	

	Meeting with Client #3	Sarah Sultan A AL YAHYA	
10.3 Meeting Note for Front-end Team		Sarah Sultan A AL YAHYA	
	Front-end Meeting Notes on 19th April	Fengrui Zhang	Fengrui Zhang Zixin Ye
	Front-end Meeting Notes on 26th April	Fengrui Zhang	Sarah Sultan A AL YAHYA
10.4 Meeting Note for Back-end Team		Sarah Sultan A AL YAHYA	
	Back-end Meeting Notes 19th April	Boyang Sun	
	Back-end Meeting Notes 24th April	Jinzhe Shan	
11 Team and role		Boyang Sun	
12 Risk Management		Boyang Sun	Jingyu LI ZISHENG CHENG Jingdan Cui