

1.8 Sprint Summaries	2
1.1 8.1 Sprint 0	3
1.1.1 8.1.1 Sprint 0: Ceremonies	4
1.1.1.1 Stand-up Meeting 18 March 2021	5
1.1.1.2 Stand-up Meeting 25 March 2021	7
1.1.2 8.1.2 Sprint 0: Documentation	9
1.1.3 8.1.3 Sprint 0: Roles and Responsibilities	14
1.2 8.2 Sprint 1	15
1.2.1 8.2.1 Review of Sprint 1 - Back-end Team	16
1.2.1.1 Coding Update	17
1.2.1.2 Design Update	19
1.2.1.3 Requirements Update	21
1.2.1.4 Task Achieved in Sprint 1	22
1.2.1.5 Testing Update	24
1.2.2 8.2.2 Review of Sprint 1 - Front-end team	26
1.2.2.1 Coding update - Front-end team	27
1.2.2.2 Deployment update - Front-end team	28
1.2.2.3 Design update - Front-end team	29
1.2.2.4 Tasks achieved - Front-end team	35
1.2.2.5 Testing update - Front-end team	37
1.2.3 8.2.3 Sprint 1: Ceremonies	38
1.2.3.1 Stand-up Meeting 15th April 2021	39
1.2.3.2 Stand-up Meeting 22nd April 2021	41
1.2.3.3 Stand-up Meeting 29th April 2021	43
1.2.4 8.2.4 Sprint 1: Documentation	45
1.2.5 8.2.5 Sprint 1: Individual Contribution	47
1.2.5.1 Contribution - Boyang Sun	48
1.2.5.2 Contribution - Chongjing Zhang	49
1.2.5.3 Contribution - Fengrui Zhang	50
1.2.5.4 Contribution - Fu Xie	51
1.2.5.5 Contribution - Haoyu Qin	52
1.2.5.6 Contribution - Jingdan Cui	53
1.2.5.7 Contribution - Jingyu Li	54
1.2.5.8 Contribution - Jinzhe Shan	55
1.2.5.9 Contribution - Pin Wang	56
1.2.5.10 Contribution - Ruofan Zhang	57
1.2.5.11 Contribution - Sai Zhang	58
1.2.5.12 Contribution - Sarah Sultan Alyahya	59
1.2.5.13 Contribution - Yuhang Xie	60
1.2.5.14 Contribution - Zisheng Cheng	61
1.2.5.15 Contribution - Zixin Ye	62
1.3 8.3 Sprint 2	63
1.3.1 8.3.1 Review of Sprint 2 - Back-end Team	64
1.3.1.1 Coding Summary - Back-end team S2	65
1.3.1.2 Deployment Report - Back-end team S2	66
1.3.1.3 Design Update - Back-end team S2	67
1.3.1.4 Requirement Update - Back-end team S2	69
1.3.1.5 Tasks achieved - Back-end team S2	70
1.3.1.6 Testing Report - Back-end team S2	72
1.3.2 8.3.2 Review of Sprint 2 - Front-end team	73
1.3.2.1 Coding summary - Front-end team S2	74
1.3.2.2 Deployment report - Front-end team S2	75
1.3.2.3 Design update - Front-end team S2	77
1.3.2.4 Handover report - Front-end team S2	86
1.3.2.5 Tasks achieved - Front-end team S2	87
1.3.3 8.3.3 Sprint 2: Ceremonies	89
1.3.3.1 Stand-up Meeting 6th May 2021	90
1.3.3.2 Stand-up Meeting 20th May 2021	92
1.3.4 8.3.4 Sprint 2: Documentation	94
1.3.5 8.3.5 Sprint 2: Individual Contribution	96
1.3.5.1 S2 Contribution - Boyang Sun	97
1.3.5.2 S2 Contribution - Chongjing Zhang	98
1.3.5.3 S2 Contribution - Fengrui Zhang	99
1.3.5.4 S2 Contribution - Fu Xie	100
1.3.5.5 S2 Contribution - Haoyu Qin	101
1.3.5.6 S2 Contribution - Jingdan Cui	102
1.3.5.7 S2 Contribution - Jingyu Li	103
1.3.5.8 S2 Contribution - Jinzhe Shan	104
1.3.5.9 S2 Contribution - Pin Wang	105
1.3.5.10 S2 Contribution - Ruofan Zhang	106
1.3.5.11 S2 Contribution - Sai Zhang	107
1.3.5.12 S2 Contribution - Sarah Sultan Alyahya	108
1.3.5.13 S2 Contribution - Yuhang Xie	109
1.3.5.14 S2 Contribution - Zisheng Cheng	110
1.3.5.15 S2 Contribution - Zixin Ye	111
1.3.6 Handover Report	112

8 Sprint Summaries

Table of Contents

- [8.1 Sprint 0](#)
 - [8.1.3 Sprint 0: Roles and Responsibilities](#)
 - [8.1.1 Sprint 0: Ceremonies](#)
 - Stand-up Meeting 18 March 2021
 - Stand-up Meeting 25 March 2021
 - [8.1.2 Sprint 0: Documentation](#)
- [8.2 Sprint 1](#)
 - Individual Contribution
 - [8.2.1 Review of Sprint 1 - Back-end Team](#)
 - Coding Update
 - Design Update
 - Requirements Update
 - Task Achieved in Sprint 1
 - Testing Update
 - [8.2.2 Review of Sprint 1 - Front-end team](#)
 - Deployment update - Front-end team
 - Coding update - Front-end team
 - Design update - Front-end team
 - Tasks achieved - Front-end team
 - Testing update - Front-end team
 - [8.2.3 Sprint 1: Ceremonies](#)
 - Stand-up Meeting 15th April 2021
 - Stand-up Meeting 22nd April 2021
 - Stand-up Meeting 29th April 2021
 - [8.2.4 Sprint 1: Documentation](#)

8.1 Sprint 0

- [8.1.3 Sprint 0: Roles and Responsibilities](#)
- [8.1.1 Sprint 0: Ceremonies](#)
 - Stand-up Meeting 18 March 2021
 - Stand-up Meeting 25 March 2021
- [8.1.2 Sprint 0: Documentation](#)

8.1.1 Sprint 0: Ceremonies

During sprint 0, The team have had two meetings with the client, a frontend group meeting, two stand-up meetings, and three weekly team meetings.

- Meeting with client #1
- Stand-up Meeting 18 March 2021
- Meeting Notes for 23rd March
- Stand-up Meeting 25 March 2021
- Meeting Notes for 28th, March
- Front-end Meeting Notes on 28th March -
- Meeting with Client #2
- Meeting Notes for 30th, March

During sprint 0, The team have had two meetings with the client, a frontend group meeting, two stand-up meetings, and three weekly team meetings.

- Meeting with client #1
- Stand-up Meeting 18 March 2021
- Meeting Notes for 23rd March
- Stand-up Meeting 25 March 2021
- Meeting Notes for 28th, March
- Front-end Meeting Notes on 28th March -
- Meeting with Client #2
- Meeting Notes for 30th, March

Stand-up Meeting 18 March 2021

Date

2021-3-18

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	<ul style="list-style-type: none">• The tasks the team have done in last week• The tasks the team plans to finish next week
Give feedback and Suggestions	Jirat Pasuksmit	<ul style="list-style-type: none">• help to split the frontend team and backend team• suggest what the team should do on the next step

Tasks completion in the Inception phase

Who	What have you done	What you plan to do next
Zhang Sai	<ol style="list-style-type: none">1. Read the project documents the previous project team provided and got familiar with the framework the previous team used before2. Join in Frontend team and found where the first task should be involved3. Listed some requirements that our team needs to re-design, which the client mentioned in the workshop	<ol style="list-style-type: none">1. Break the whole project into frontend team and backend team2. Define team role and responsibility in Frontend team3. make a decision about what can be reused from the previous team.
RuoFan Zhang	<ol style="list-style-type: none">1. Threw out ideas of how this project can be done on Slack2. Gathered team members of the front-end team3. Run the front-end project to evaluate the workload	<ol style="list-style-type: none">1. Have a meeting to discuss the re-design of the current front-end pages2. Deliver a prototype to show the client
Jingyu Li	<ol style="list-style-type: none">1. Review the documents and codes of the previous team2. Review the client's requirements	<ol style="list-style-type: none">1. Have a meeting to discuss the re-design of the current front-end pages2. Do the prototype using Axure
Sarah Sultan	<ol style="list-style-type: none">1. Review the documents from the previous team.2. Review the code from the previous team.3. Understand the requirement of the front-end side.	<ol style="list-style-type: none">1. Have a meeting to discuss the re-design of the current front-end pages.2. Learn knowledge about front-end

Yuhang Xie	<ul style="list-style-type: none"> 1. Review the documents and codes from the last team. 2. Hold a meeting with the whole team and frontend team respectively. 3. Discuss the UI design with the team. 	<ul style="list-style-type: none"> 1. Determine the product structure by XMind. 2. Determine the task assignment for the frontend team. 3. Design a prototype of the product.
Zixin Ye	<ul style="list-style-type: none"> 1. Review the documents of the previous team 2. Contact client for requirements clarification 	<ul style="list-style-type: none"> 1. Set up next meeting 2. Contact client for further information 3. Discuss the details of the prototype with teammates
Fengru Zhang	<ul style="list-style-type: none"> 1. Review the documents of the previous team. 2. Review the client's requirements. 	<ul style="list-style-type: none"> 1. Have a meeting to discuss the re-design of the current front-end pages
Boyan Sun	<ul style="list-style-type: none"> 1. Give out some technology choosing ideas. 2. Give out some workload allocation ideas. 3. Review the previous documentation and give out some suggestions 	<ul style="list-style-type: none"> 1. Read the back-end code and give out deployment instructions. 2. Learn about all the user stories.
Zisheng Cheng	<ul style="list-style-type: none"> 1. Review the documentation of the previous team 2. Read the back-end code 	<ul style="list-style-type: none"> 1. Read the back-end code 2. Summary of the functions that were implemented by the previous team.
Jingdan Cui	<ul style="list-style-type: none"> 1. Choose to be a backend developer 2. Review the documentation of the previous team 3. Learn some basic knowledge of Django 	<ul style="list-style-type: none"> 1. Read the source code. 2. Figure out the functions which are done by the previous team
Haoyu Qin	<ul style="list-style-type: none"> 1. Review of the existing documentation 	<ul style="list-style-type: none"> 2. Assess & migrate Process user stories to Trello
Jinzhe Shan	<ul style="list-style-type: none"> 1. Review the documents of the previous team -especially User Story Part 2. Review the client's requirements. 	
Pin Wang	<ul style="list-style-type: none"> 1. Review the documents of the previous team. 2. Understand the requirement of this project from the back-end side. 	<ul style="list-style-type: none"> 1. Start to read code 2. Summary of the functions that were implemented by the previous team.
Fu Xie	<ul style="list-style-type: none"> 1. Review the documentation of the previous team. 2. Combine previous document and new requirement to design a story card of the product part 	<ul style="list-style-type: none"> 1. Review the documentation of the previous team. 2. Make architectural design diagrams(use case diagram, domain, model, process diagram and communication diagram) according to the existing situation
Chongjing Zhang	<ul style="list-style-type: none"> 1. Review the documentation of the previous team. 2. Review the client's requirement 3. Combine previous document and new requirement to design story card of Individual part and Communication part 	<ul style="list-style-type: none"> 1. Take charge of follow-up tasks 2. Write backlog

Stand-up Meeting 25 March 2021

Date

2021-3-25

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	<ul style="list-style-type: none">• The tasks the team have done in last week• The tasks the team plans to finish next week
Give feedback and Suggestions	Jirat Pasuksmit	<ul style="list-style-type: none">• Plan the next step• Necessary document in Sprint 0

Tasks completion in the Inception phase

Who	What have you done	What you plan to do next
Zhang Sai	<ol style="list-style-type: none">1. Review prototype requirements the client mentioned before2. Discuss with the team for defining tasks and breaking them down into smaller tasks and assigning them to each team member.3. Defined team role	<ol style="list-style-type: none">1. Complete prototype with teammates together2. Show the completed prototype work to our client3. Clarify more detailed UI requirements with the client
RuoFan Zhang	<ol style="list-style-type: none">1. Specified the details of each web page given the requirements of the client with teammatesMonitor the progress of the prototype design	<ol style="list-style-type: none">1. Assign the coding tasks for each member of the front-end team2. Start coding and monitor the progress of the front-end team3. Help teammates to solve problems encountered during the first sprint
Jingyu Li	<ol style="list-style-type: none">1. Have a meeting to discuss the re-design of the current front-end pages2. Review the front-end pages of the previous team3. Work on prototype	<ol style="list-style-type: none">1. Continue doing the prototype using Axure2. Show the prototype to client3. Work on Sprint Summaries and Risk Management documents
Sarah Sultan	<ol style="list-style-type: none">1. Have a meeting to discuss the re-design of the current front-end pages.2. Review the front-end pages of the previous team.3. Have a meeting with a teammate to discuss the individual page.	<ol style="list-style-type: none">1. Work on Sprint Summaries- sprint 0 with the teammate.2. Update confluence page on Sprint Summaries.3. Learn knowledge about Axure and how to use it.

Yuhang Xie	<ul style="list-style-type: none"> 1. Design a product structure of the system. 2. Task assignment by separating the product structure. 3. Design the product prototype. 	<ul style="list-style-type: none"> 1. Communicate with the client and clarify the requirements. 2. Write the design concept of the product. 3. Modify product prototype.
Zixin Ye	<ul style="list-style-type: none"> 1. Review the documentation of the previous team. 2. Write the requirements part of our documentation. 	<ul style="list-style-type: none"> 1. Finish the requirements of our documentation. 2. Modify Trello to make it consistent with our documentation.
Fengru Zhang	<ul style="list-style-type: none"> 1. Have a meeting to discuss the re-design of the current front-end pages. 2. Review the front-end pages of the previous team. 	<ul style="list-style-type: none"> 1. Write the initial version of the Design Concept document.
Boyan Sun	<ul style="list-style-type: none"> 1. Read the source code provided by the previous team 2. Analysis the previous code quality, test cases and implemented interfaces 3. Meeting with other back-end teammates and discuss reusing the previous code 	<ul style="list-style-type: none"> 1. Write the API documentations 2. Write the Backlog documentations 3. Make sure the team members are working on the right things 4. Sprint 0 workload allocation
Zisheng Cheng	<ul style="list-style-type: none"> 1. Read the source code provided by the previous team 2. Summary of the functions that were implemented. by the previous team 3. Try to deploy a back-end server 	<ul style="list-style-type: none"> 1. Update Confluence page on risk management component 2. Continue reading the back-end code and test API
Jingdan Cui	<ul style="list-style-type: none"> 1. Read the source code provided by the previous team 2. Find out the functions which are developed by the previous team 3. Continue learning some knowledge of Django 	<ul style="list-style-type: none"> 1. Try to deploy the project on the laptop 2. Work on Sprint Summaries and Risk Management documents 3. Continue learning the needed technologies
Haoyu Qin	<ul style="list-style-type: none"> 1. Assess & migrate Process user stories to Trello 	<ul style="list-style-type: none"> 1. Assess testing component 2. Update Confluence page on testing component
Jinzheshan	<ul style="list-style-type: none"> 1. Meeting with other back-end teammates and discuss working items 2. Read related knowledge about Testing and Deployment 3. Write Testing and Deployment documents on Confluence page (process:1 /3) 	<ul style="list-style-type: none"> 1. Learn and use testing tools 2. Complete the first version of Testing and Deployment documents on the Confluence page
Pin Wang	<ul style="list-style-type: none"> 1. Read the back-end code provided by the previous team 2. Understand their structure 3. Learn Django 4. Summary the functions that were implemented. by the previous team 5. Try to deploy a back-end server 	<ul style="list-style-type: none"> 1. Test all APIs implemented by the previous team 2. Write a back-end development doc 3. Fill out our API doc
Fu Xie	<ul style="list-style-type: none"> 1. Read the previous architectural design diagrams and made new architectural design diagrams(use case diagram, domain model, process diagram and communication diagram) based on the current situation 2. Fix the conflict between the current diagram and the previous data schema. 	<ul style="list-style-type: none"> 1. Work and communication with the frontend team to finalize the architectural design. Make sure there is consistent between the front-end and back-end part 2. Work with frontend team members to finalize the design concept document.
Chongjing Zhang	<ul style="list-style-type: none"> 1. Understand tasks content and assign priorities 	<ul style="list-style-type: none"> 1. Finish backlog part

8.1.2 Sprint 0: Documentation

Title		Creator	Modified
1 Project Requirement		Boyang Sun	Zixin Ye
	1.1 Project Overview	Zixin Ye	
	1.2 Quality Requirements	Zixin Ye	
	1.3.0 Persona Draft	Zixin Ye	
	1.3.1 Coordinator Persona	Zixin Ye	
	1.4 Goal Models	Zixin Ye	
	1.5 User stories	Zixin Ye	
	1.6 Project Analysis Metrics	Zixin Ye	
2 Design Concepts		Boyang Sun	Fengrui Zhang Fu Xie
3 Architecture Design		Boyang Sun	Fu Xie
	3.1 Use Case Diagram	Fu Xie	
	3.2 Domain Model	Fu Xie	
	3.3 Process Diagram	Fu Xie	
	3.4 Communication Diagram	Fu Xie	
4 Prototypes		Boyang Sun	Ruofan Zhang
5 Backlogs		Boyang Sun	
	5.1 Sprint 0: Backlog	Boyang Sun	Boyang Sun Chongjing ZHANG
6 API documents		Boyang Sun	Pin Wang
	6.1.0 Database schema	Boyang Sun	Fu Xie
	6.1.1 Subject	Boyang Sun	Pin Wang
	6.1.2 Project	Boyang Sun	Pin Wang
	6.1.3 Team	Boyang Sun	Pin Wang
	6.1.4 Confluence	Boyang Sun	Pin Wang
	6.1.5 Jira	Boyang Sun	
	6.1.6 Git	Boyang Sun	
	6.2.4 Data Sample	Jingdan Cui	Jingdan Cui
7 Deployment		Boyang Sun	Jinzhe Shan Pin Wang
8 Sprint Summaries		Boyang Sun	Sarah Sultan A AL YAHYA
8.1 Sprint 0		Jingdan Cui	
	8.1.3 Sprint 0: Roles and Responsibilities	Jingdan Cui	Boyang Sun Jingdan Cui Yuhang Xie
	8.1.1 Sprint 0: Ceremonies	Jingdan Cui	
	8.1.2 Sprint 0: Documentation	Sarah Sultan A AL YAHYA	Sarah Sultan A AL YAHYA
9 Testing		Boyang Sun	Jinzhe Shan
	9.1 Functional Testing Structure	Haoyu Qin	
	9.2 Non-Functional Testing Structure	Haoyu Qin	
	9.3 Acceptance Testing	Jinzhe Shan	
10 Meetings		Boyang Sun	
10.1 Meeting Notes for Project Team		Boyang Sun	
	Meeting Notes for 23rd March	Boyang Sun	Boyang Sun Jingdan Cui Pin Wang
	Meeting Notes for 28th, March	Boyang Sun	Boyang Sun Jingdan Cui
	Meeting Notes for 30th, March	Boyang Sun	Boyang Sun ZISHENG CHENG
10.2 Meeting with Clients		Boyang Sun	

	Meeting with client #1	Jirat Pasuksmit	Pin Wang Ruofan Zhang Jingdan Cui
	Meeting with Client #2	Boyang Sun	Boyang Sun Sai Zhang
10.3 Meeting Note for Front-end Team		Sarah Sultan A AL YAHYA	
	Front-end Meeting Notes on 28th March -	Fengrui Zhang	Fengrui Zhang Boyang Sun Sarah Sultan A AL YAHYA
10.4 Meeting Note for Back-end Team		Sarah Sultan A AL YAHYA	
11 General Process Documentation		Boyang Sun	
12 Risk Management		Boyang Sun	Jingyu LI ZISHENG CHENG Jingdan Cui

	Title	Creator	Modified
1 Project Requirement		Boyang Sun	Zixin Ye
	1.1 ProjectOverview	Zixin Ye	
	1.2 Quality Requirements	Zixin Ye	
	1.3.0 Persona Draft	Zixin Ye	
	1.3.1 Coordinator Persona	Zixin Ye	
	1.4 Goal Models	Zixin Ye	
	1.5 User stories	Zixin Ye	
	1.6 Project Analysis Metrics	Zixin Ye	
2 Design Concepts		Boyang Sun	Fengrui Zhang Fu Xie
3 Architecture Design		Boyang Sun	Fu Xie
	3.1 Use Case Diagram	Fu Xie	
	3.2 Domain Model	Fu Xie	
	3.3 Process Diagram	Fu Xie	
	3.4 Communication Diagram	Fu Xie	
4 Prototypes		Boyang Sun	Ruofan Zhang
5 Backlogs		Boyang Sun	
	5.1 Sprint 0: Backlog	Boyang Sun	Boyang Sun Chongjing ZHANG
6 API documents		Boyang Sun	Pin Wang
	6.1.0 Database schema	Boyang Sun	Fu Xie
	6.1.1 Subject	Boyang Sun	Pin Wang
	6.1.2 Project	Boyang Sun	Pin Wang
	6.1.3 Team	Boyang Sun	Pin Wang
	6.1.4 Confluence	Boyang Sun	Pin Wang
	6.1.5 Jira	Boyang Sun	
	6.1.6 Git	Boyang Sun	
	6.2.4 Data Sample	Jingdan Cui	Jingdan Cui
7 Deployment		Boyang Sun	Jinzhe Shan Pin Wang
8 Sprint Summaries		Boyang Sun	Sarah Sultan A AL YAHYA
8.1 Sprint 0		Jingdan Cui	
	8.1.3 Sprint 0: Roles and Responsibilities	Jingdan Cui	Boyang Sun Jingdan Cui Yuhang Xie
	8.1.1 Sprint 0: Ceremonies	Jingdan Cui	
	8.1.2 Sprint 0: Documentation	Sarah Sultan A AL YAHYA	Sarah Sultan A AL YAHYA
9 Testing		Boyang Sun	Jinzhe Shan
	9.1 Functional Testing Structure	Haoyu Qin	
	9.2 Non-Functional Testing Structure	Haoyu Qin	

	9.3 Acceptance Testing	Jinzhe Shan	
10 Meetings		Boyang Sun	
10.1 Meeting Notes for Project Team		Boyang Sun	
	Meeting Notes for 23rd March	Boyang Sun	Boyang Sun Jingdan Cui Pin Wang
	Front-end Meeting Notes on 28th March -	Fengrui Zhang	Fengrui Zhang Boyang Sun Sarah Sultan A AL YAHYA
	Meeting Notes for 28th, March	Boyang Sun	Boyang Sun Jingdan Cui
	Meeting Notes for 30th, March	Boyang Sun	Boyang Sun ZISHENG CHENG
10.2 Meeting with Clients		Boyang Sun	
	Meeting with client #1	Jirat Pasuksmit	Pin Wang Ruofan Zhang Jingdan Cui
	Meeting with Client #2	Boyang Sun	Boyang Sun Sai Zhang
11 General Process Documentation		Boyang Sun	
12 Risk Management		Boyang Sun	Jingyu LI ZISHENG CHENG Jingdan Cui

Title		Creator	Modified
1 Project Requirement		Boyang Sun	Zixin Ye
	1.1 Project Overview	Zixin Ye	
	1.2 Quality Requirements	Zixin Ye	
	1.3.0 Persona Draft	Zixin Ye	
	1.3.1 Coordinator Persona	Zixin Ye	
	1.4 Goal Models	Zixin Ye	
	1.5 User stories	Zixin Ye	
	1.6 Project Analysis Metrics	Zixin Ye	
2 Design Concepts		Boyang Sun	Fengrui Zhang Fu Xie
3 Architecture Design		Boyang Sun	Fu Xie
	3.1 Use Case Diagram	Fu Xie	
	3.2 Domain Model	Fu Xie	
	3.3 Process Diagram	Fu Xie	
	3.4 Communication Diagram	Fu Xie	
4 Prototypes		Boyang Sun	Ruofan Zhang
5 Backlogs		Boyang Sun	
	5.1 Sprint 0: Backlog	Boyang Sun	Boyang Sun Chongjing ZHANG
6 API documents		Boyang Sun	Pin Wang
	6.1.0 Database schema	Boyang Sun	Fu Xie
	6.1.1 Subject	Boyang Sun	Pin Wang
	6.1.2 Project	Boyang Sun	Pin Wang
	6.1.3 Team	Boyang Sun	Pin Wang
	6.1.4 Confluence	Boyang Sun	Pin Wang
	6.1.5 Jira	Boyang Sun	
	6.1.6 Git	Boyang Sun	
	6.2.4 Data Sample	Jingdan Cui	Jingdan Cui
7 Deployment		Boyang Sun	Jinzhe Shan Pin Wang
8 Sprint Summaries		Boyang Sun	Sarah Sultan A AL YAHYA
8.1 Sprint 0		Jingdan Cui	
	8.1.3 Sprint 0: Roles and Responsibilities	Jingdan Cui	Boyang Sun Jingdan Cui Yuhang Xie
	8.1.1 Sprint 0: Ceremonies	Jingdan Cui	

	8.1.2 Sprint 0: Documentation	Sarah Sultan A AL YAHYA	Sarah Sultan A AL YAHYA
9 Testing		Boyang Sun	Jinzhe Shan
	9.1 Functional Testing Structure	Haoyu Qin	
	9.2 Non-Functional Testing Structure	Haoyu Qin	
	9.3 Acceptance Testing	Jinzhe Shan	
10 Meetings		Boyang Sun	
10.1 Meeting Notes for Project Team		Boyang Sun	
	Meeting Notes for 23rd March	Boyang Sun	Boyang Sun Jingdan Cui Pin Wang
	Meeting Notes for 28th, March	Boyang Sun	Boyang Sun Jingdan Cui
	Meeting Notes for 30th, March	Boyang Sun	Boyang Sun ZISHENG CHENG
10.2 Meeting with Clients		Boyang Sun	
	Meeting with client #1	Jirat Pasuksmit	Pin Wang Ruofan Zhang Jingdan Cui
	Meeting with Client #2	Boyang Sun	Boyang Sun Sai Zhang
10.3 Meeting Note for Front-end Team		Sarah Sultan A AL YAHYA	
	Front-end Meeting Notes on 28th March -	Fengrui Zhang	Fengrui Zhang Boyang Sun Sarah Sultan A AL YAHYA
10.4 Meeting Note for Back-end Team		Sarah Sultan A AL YAHYA	
11 General Process Documentation		Boyang Sun	
12 Risk Management		Boyang Sun	Jingyu LI ZISHENG CHENG Jingdan Cui

Title		Creator	Modified
1 Project Requirement		Boyang Sun	Zixin Ye
	1.1 ProjectOverview	Zixin Ye	
	1.2 Quality Requirements	Zixin Ye	
	1.3.0 Persona Draft	Zixin Ye	
	1.3.1 Coordinator Persona	Zixin Ye	
	1.4 Goal Models	Zixin Ye	
	1.5 User stories	Zixin Ye	
	1.6 Project Analysis Metrics	Zixin Ye	
2 Design Concepts		Boyang Sun	Fengrui Zhang Fu Xie
3 Architecture Design		Boyang Sun	Fu Xie
	3.1 Use Case Diagram	Fu Xie	
	3.2 Domain Model	Fu Xie	
	3.3 Process Diagram	Fu Xie	
	3.4 Communication Diagram	Fu Xie	
4 Prototypes		Boyang Sun	Ruofan Zhang
5 Backlogs		Boyang Sun	
	5.1 Sprint 0: Backlog	Boyang Sun	Boyang Sun Chongjing ZHANG
6 API documents		Boyang Sun	Pin Wang
	6.1.0 Database schema	Boyang Sun	Fu Xie
	6.1.1 Subject	Boyang Sun	Pin Wang
	6.1.2 Project	Boyang Sun	Pin Wang
	6.1.3 Team	Boyang Sun	Pin Wang
	6.1.4 Confluence	Boyang Sun	Pin Wang
	6.1.5 Jira	Boyang Sun	

	6.1.6 Git	Boyang Sun	
	6.2.4 Data Sample	Jingdan Cui	Jingdan Cui
7 Deployment		Boyang Sun	Jinzhe Shan Pin Wang
8 Sprint Summaries		Boyang Sun	Sarah Sultan A AL YAHYA
8.1 Sprint 0		Jingdan Cui	
	8.1.3 Sprint 0: Roles and Responsibilities	Jingdan Cui	Boyang Sun Jingdan Cui Yuhang Xie
	8.1.1 Sprint 0: Ceremonies	Jingdan Cui	
	8.1.2 Sprint 0: Documentation	Sarah Sultan A AL YAHYA	Sarah Sultan A AL YAHYA
9 Testing		Boyang Sun	Jinzhe Shan
	9.1 Functional Testing Structure	Haoyu Qin	
	9.2 Non-Functional Testing Structure	Haoyu Qin	
	9.3 Acceptance Testing	Jinzhe Shan	
10 Meetings		Boyang Sun	
10.1 Meeting Notes for Project Team		Boyang Sun	
	Meeting Notes for 23rd March	Boyang Sun	Boyang Sun Jingdan Cui Pin Wang
	Front-end Meeting Notes on 28th March -	Fengrui Zhang	Fengrui Zhang Boyang Sun Sarah Sultan A AL YAHYA
	Meeting Notes for 28th, March	Boyang Sun	Boyang Sun Jingdan Cui
	Meeting Notes for 30th, March	Boyang Sun	Boyang Sun ZISHENG CHENG
10.2 Meeting with Clients		Boyang Sun	
	Meeting with client #1	Jirat Pasuksmit	Pin Wang Ruofan Zhang Jingdan Cui
	Meeting with Client #2	Boyang Sun	Boyang Sun Sai Zhang
11 General Process Documentation		Boyang Sun	
12 Risk Management		Boyang Sun	Jingyu LI ZISHENG CHENG Jingdan Cui

8.1.3 Sprint 0: Roles and Responsibilities

Name	Role	Responsibility
Yuhang Xie	Scrum Master	<ul style="list-style-type: none">• Ensure that the product is completed in the final submissions• Manage the team to make sure the requirements in every sprint are met• Help team members to allocate tasks to themselves• Coordinate with the product owner to achieve the client's expectation
Sai Zhang	Product Owner	<ul style="list-style-type: none">• Communicate and negotiate with the client• Act as a "bridge" between the client and the team
Ruofan Zhang	Frontend Team Lead	<ul style="list-style-type: none">• Lead the frontend team which is working on the frontend tasks• Provide help when team members encounter difficulties in writing code• Responsible for merging and maintaining the branches of the code• Control the code version in the repository
Boyang Sun	Backend Team Lead	<ul style="list-style-type: none">• Lead the backend team which is working on the database and backend tasks• Provide help when team members encounter difficulties in writing code• Responsible for merging and maintaining the branches of the code• Control the code version in the repository
Pin Wang, Fu Xie	Design and Architecture Lead	<ul style="list-style-type: none">• Responsible for the design model of the project• Adjust the architecture when needed
Jingdan Cui, Jingyu Li	Risk Management Lead	<ul style="list-style-type: none">• Monitor risks during the development process and document when risks occur
Jinzhe Shan, Haoyu Qin	Testing Lead	<ul style="list-style-type: none">• Responsible for performing various tests, such as unit testing, integration testing, system testing and acceptance testing.• Related Documentation Writing

8.2 Sprint 1

- Individual Contribution
- 8.2.1 Review of Sprint 1 - Back-end Team
 - Coding Update
 - Design Update
 - Requirements Update
 - Task Achieved in Sprint 1
 - Testing Update
- 8.2.2 Review of Sprint 1 - Front-end team
 - Deployment update - Front-end team
 - Coding update - Front-end team
 - Design update - Front-end team
 - Tasks achieved - Front-end team
 - Testing update - Front-end team
- 8.2.3 Sprint 1: Ceremonies
 - Stand-up Meeting 15th April 2021
 - Stand-up Meeting 22nd April 2021
 - Stand-up Meeting 29th April 2021
- 8.2.4 Sprint 1: Documentation

8.2.1 Review of Sprint 1 - Back-end Team

Design Update

Requirements Update

Task Achieved in Sprint 1

Testing Update

Coding Update

7 Deployment

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 Imported Projects

- Pull Request: https://github.com/pete965/COMP90082_Software_Project_Database_Backend/pull/15
- Added Functions:
 - get_imported_project
- Edited Function: No implementation in the older system.

Get Meeting Minutes

- Pull Request: https://github.com/pete965/COMP90082_Software_Project_Database_Backend/pull/8
- Added Functions:
 - get_meeting_minutes
 - update_meeting_minutes
- Edited Functions: No implementation in the older system.

Get Spaces by Keyword

- Pull Request: https://github.com/pete965/COMP90082_Software_Project_Database_Backend/pull/3
- Added Functions:
 - get_spaces_by_key
- Edited Functions: No implementation in the older system.

Get User List

- Pull Request: https://github.com/pete965/COMP90082_Software_Project_Database_Backend/pull/7
- Added Functions:
 - get_user_list
 - update_user_list
 - get_user
- Edited Functions: No implementation in the older system.

Get Page Count by Time

- Pull Request: https://github.com/pete965/COMP90082_Software_Project_Database_Backend/pull/9
- Added Functions:
 - get_page_count_by_time
 - update_page_history
- Edited Functions: No implementation in the older system.

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

Jira

Get Jira TO_DO/IN_PROGRESS/DONE Count by Time

- Pull Request: https://github.com/pete965/COMP90082_Software_Project_Database_Backend/pull/14
- Added Functions:
 - class JiraCountByTime(models.Model)
 - get_ticket_count_team_timestamp
 - auto_get_ticket_count_team_timestamped
 - get_ticket_count_team_timestamped_withoutupdate
- Edited Functions: No implementation in the older system.

Get Individual Contribution on Jira DONE Count Tickets

- Pull Request: https://github.com/pete965/COMP90082_Software_Project_Database_Backend/pull/12
- Added Functions:
 - class IndividualContributions(models.Model)
 - get_contributions
- Edited Functions: No implementation in the older system.

Set Github and Jira Url

- Pull Request: https://github.com/pete965/COMP90082_Software_Project_Database_Backend/pull/14
- Added Functions:
 - class Urlconfig(models.Model)
 - setGithubJiraUrl
- Edited Functions: No implementation in the older system.

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: [6.2.3 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 pages from the Confluence space, and the meeting minutes data will be stored and updated in the database.

Get Spaces by Keyword

- Older design: The group names is used to distinguish groups in the older design. There is an implementation of API for searching the group name that contains the keyword.
- New design: The Confluence spaces is used to distinguish groups. Use Confluence REST API to retrieve the real time space key list, and return the ones that contains the keyword.

Get User List

- Older design: The user list is the group members in the specific Confluence group. There is an implementation of API for getting the group members. However, when you call this API, it will get the data from confluence RESTFUL API and it usually takes a few seconds.
- New design: The user list is the members that have permissions to the specific Confluence space. The user list is retrieved from the database. The system calls the Confluence API every 24 hours to get the user list from the Confluence space, and the user list data will be stored and updated in the database. This design enhanced the total efficiency and can react in a few milliseconds

Get Page Count by Time

- Older design: No design and implementation for this API in the older version.
- New design: The page counts in the Confluence space are calculated everyday since this space is created. The page counts is retrieved from the database. The system calls the Confluence API every 24 hours to get the page counts from the Confluence space, and the page counts data will be stored and updated in 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: [6.2.3 Database Design](#)

Get Jira TO_DO/IN_PROGRESS/DONE Count by Time

- Older design: Just get TO_DO/IN_PROGRESS/DONE count, no timestamp.
- New design: Get TO_DO/IN_PROGRESS/DONE count with a timestamp for each record. It will retrieve the data from the Jira API every 24 hours and store it in the database. DB design: [6.2.3 Database Design](#)

Get Individual Contribution on Jira DONE Count Tickets

- Older design: No interaction with database
- New design: The data obtained is written into the database once the method executes. DB design: [6.2.3 Database Design](#)

Set Github and Jira Url

- Older design: No design and implementation for this API in older software.
- New design: POST method to get api calls from the front-end and write the data into database. DB design: [6.2.3 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](#)

Get Imported Projects

- Previous system: The Confluence space data is directly retrieved from calling Confluence API.
- Our work: Search each data in project_coordinator_relation table in the database. Select and return all the projects which are imported by the coordinator.
- Related User Story: [1.5 User stories US - 2](#)

Get Meeting Minutes

- Previous system: There is no implementation of meeting minutes in the previous version.
- Our work: By given a specific space key, all the data in the meeting_minutes table in the database can be retrieved. The chosen titles and links of the meeting minutes will be sent to the frontend.
- Related User Story: [1.5 User stories US - 13](#)

Get Spaces by Keyword

- Previous system: The group names is used to distinguish groups in the older design and there exists relative implementations.
- Our work: By given a keyword, a list of space keys that contain the keyword is sent to the frontend.
- Related User Story: [1.5 User stories US - 1](#)

Get User List

- Previous system: The group names is used to distinguish groups in the older design and there exists relative implementations.
- Our work: By given a specific space key, all the data in the user_list table in the database can be retrieved to reduce the response time and impove the user experience. The list of users that have permissions to that space is sent to the frontend. And the database is updated regurlarly.
- Related User Story: [1.5 User stories US - 1](#)

Get Page Count by Time

- Previous system: There is no implementation of meeting minutes in the previous version.
- Our work: By given a specific space key, all the data in the page_history table in the database can be retrieved to reduce the response time and impove the user experience. The page counts in the Confluence space on each day is sent to the frontend. And the database is updated regurlarly.
- Related User Story: [1.5 User stories US - 7](#)

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

Get Jira TO_DO/IN_PROGRESS/DONE Count by Time

- Previous system: In the previous design, the api obtains TO_DO/IN_PROGRESS/DONE count without the timestamp. It does not fetch data every day. It does not interact with database.
- Our work: Adding a timestamp to each records obtained from Jira. The method can automatically execute after a period of time. The data obtained is written into the database once the method executes.
- Related User Story: [1.5 User stories US - 8](#)

Get Individual Contribution on Jira DONE Count Tickets

- Previous system: In the previous design, It does not interact with database.
- Our work: The data obtained is written into the database once the method executes.
- Related User Story: [1.5 User stories US - 15](#)

Set Github and Jira Url

- Previous system: There is no implementation of meeting minutes in the previous version.
- Our work: Developed a post method which will be called by the front-end, and send the data to the database once the method is executed.
- Related User Story: [1.5 User stories US - 22](#)

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

The screenshot shows the Postman interface with the 'Import' tab selected in the top navigation bar. On the left, the 'My Workspace' sidebar lists collections, APIs, environments, mock servers, monitors, and history. In the main area, a 'Import Project in Batch' dialog is open. It shows a list of projects from a local environment, including 'CompareLocalAndCloud', 'firstRequest', and 'Import Project in Batch'. Below this, there's a section for 'Authorization' with a dropdown set to 'No Auth'. A note states: 'This collection does not use any authorization. [Learn more about authorization](#)'.

The screenshot shows the 'Import' dialog window. At the top, tabs for 'File', 'Folder', 'Link', 'Raw text', and 'Code repository' are visible, with 'Code repository' currently selected. Below this is a large text input area with the placeholder 'Drag and drop Postman data or any of the formats below'. Underneath the input area are buttons for 'OpenAPI', 'RAML', 'GraphQL', 'cURL', and 'WADL'. Below these buttons is the text 'OR' and a red arrow points to the 'Upload Files' button.

- Click the send button in the corresponding request

The screenshot shows the Postman interface with a request in progress. The 'Import Project in Batch' collection is selected. A specific POST request is highlighted in the list, showing its URL as '127.0.0.1:8000/api/v1/project/import'. The 'Send' button is highlighted with a red arrow. The request details panel shows 'Params', 'Authorization', 'Headers (10)', 'Body', 'Pre-request Script', 'Tests', and 'Settings' tabs. The 'Body' tab is active, showing a table for 'Query Params' with one entry: 'Key' under 'KEY' and 'Value' under 'VALUE'. The 'Send' button is located at the bottom right of the request details panel.

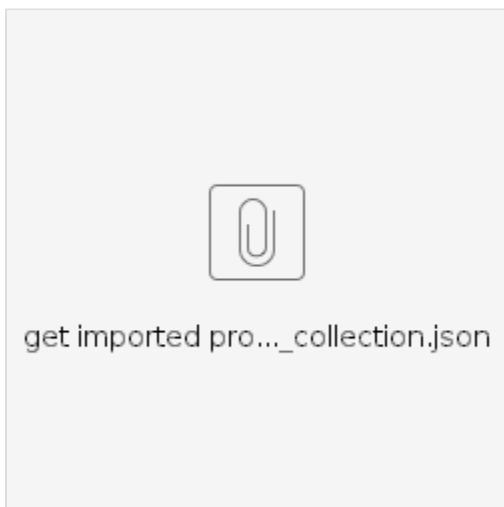
- Finally, you will see the output like these: [6.2.1 Acceptance Test](#)

Confluence

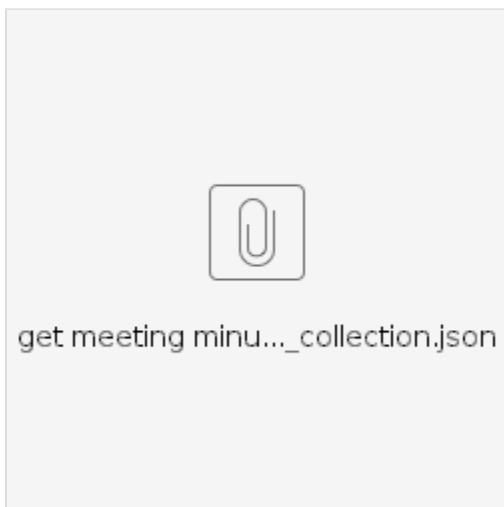
Import Project in Batch



Get Imported Projects



Get Meeting Minutes



8.2.2 Review of Sprint 1 - Front-end team

[Coding update - Front-end team](#)

[Deployment update - Front-end team](#)

[Design update - Front-end team](#)

[Tasks achieved - Front-end team](#)

[Testing update - Front-end team](#)

Coding update - Front-end team

76 changed files with 21,550 additions and 4,502 deletions since last sprint

Major changes can be found in the following files:

1. UI implementation
 - CoordinatorHomePage.js
 - ProjectHomePage.js
 - ProductQualityPage.js
 - ProcessQualityPage.js
 - CommunicationQualityPage.js
 - IndividualContributionPage.js
 - ProjectSettingsPage.js
2. API data fetching and Redux implementation
 - user.service.js
 - user.actions.js
 - user.reducer.js
3. Public tools
 - Alert.js
 - Banner.js
 - ButtonGroup.js
 - DonutChart.js
 - LineChart.js
 - Table.js
 - formatDonutChartData.js
 - formatLineChartData.js
 - formatProjectInfo.js
 - formatSearchResult.js
 - toast.js
 - unixToDate.js

Details of all changed files can be found in <https://github.com/ruofzhang/COMP90082-SM1-2021-SP-Frontend/compare/INCEPTION...main>

Deployment update - Front-end team

Steps to run the project:

1. ensure Git and Node.js are installed in your machine
2. enter 'git clone <https://github.com/ruofzhang/COMP90082-SM1-2021-SP-Frontend.git>' in command line or terminal to download the project
3. In command line or terminal, enter 'cd COMP90082-SM1-2021-SP-Frontend'
4. In command line or terminal, enter 'npm install' to download the dependencies
5. In command line or terminal, enter 'npm start' to run the app. By default, it will run on <http://localhost:3000>.

Design update - Front-end team

- 1. Coordinator homepage
- 2. Project overview
- 3. Process quality
- 4. Product quality
- 5. Communication quality
- 6. Individual contribution page
- 7. Configuration

1. Coordinator homepage

- Previous:

The screenshot shows a dark-themed web application. On the left is a vertical sidebar with icons for Home (C), Import Projects, Manage Supervisors, and Logout. The main header reads "Coordinator Homepage". Below the header is a search bar with dropdowns for "Choose Subject" (set to SWEN90013) and "Choose Year" (set to 2019). A table below the search bar has columns: ID, NAME, PROJECT_NAME, SUPERVISOR, SECONDARY_SUPERVISOR, and YEAR. A "SAVE CHANGES" button is at the bottom of the table.

- Current:

The screenshot shows a dark-themed web application. On the left is a vertical sidebar with icons for Home (C), Import Projects, Manage Supervisors, and Logout. The main header reads "Homepage". Below the header is a section titled "PROJECT MANAGEMENT". A search bar contains the letter "su". A table lists imported projects with columns: Project Imported, COMMUNICATIVE LINK, and Operation. Each row includes a "DELETE" button. The table rows are:

Project Imported	COMMUNICATIVE LINK	Operation
Motivational Modelling	https://confluence.cis.unimelb.edu.au:8443/display/Sample2	<button>DELETE</button>
Planimation	https://confluence.cis.unimelb.edu.au:8443/display/Sample4	<button>DELETE</button>
Software Resources for CIS subjects	https://confluence.cis.unimelb.edu.au:8443/display/Sample6	<button>DELETE</button>

2. Project overview

- Previous:

The screenshot shows the TQ Project Homepage. On the left is a vertical sidebar with the University of Melbourne logo at the top. Below it are several icons with labels: 'Project Home' (house), 'Process' (chart), 'Communication' (speech bubble), 'Individual' (person), 'Configuration' (gear), and 'Logout' (key). In the center, the title 'TQ Project Homepage' is displayed above a large white area. At the bottom of this central area is a green button labeled 'GET TEAM LIST'.

- Current:

The screenshot shows the Project Overview page. The sidebar on the left includes the University of Melbourne logo, 'Home' (house), 'Project Overview' (chart), 'Product' (lightbulb), 'Process' (chart), 'Communication' (speech bubble), 'Individual' (person), 'Configuration' (gear), and 'Logout' (key). The main content area has a dark blue header with the title 'Project Overview'. Below the header is a section titled 'Student Information' containing a table with 10 rows of student data.

Name	Profile	Student ID	Email Address
Student 1		123423	student1@student.unimleb.edu.au
Student 2		423456	student2@student.unimleb.edu.au
Student 3		234789	student3@student.unimleb.edu.au
Student 4		122343	student4@student.unimleb.edu.au
Student 5		623452	student5@student.unimleb.edu.au
Student 6		343789	student6@student.unimleb.edu.au
Student 7		89123	student7@student.unimleb.edu.au
Student 8		983456	student8@student.unimleb.edu.au
Student 9		71289	student9@student.unimleb.edu.au

3. Process quality

- Previous:

 THE UNIVERSITY OF MELBOURNE

Project Home

Process

Communication

Individual

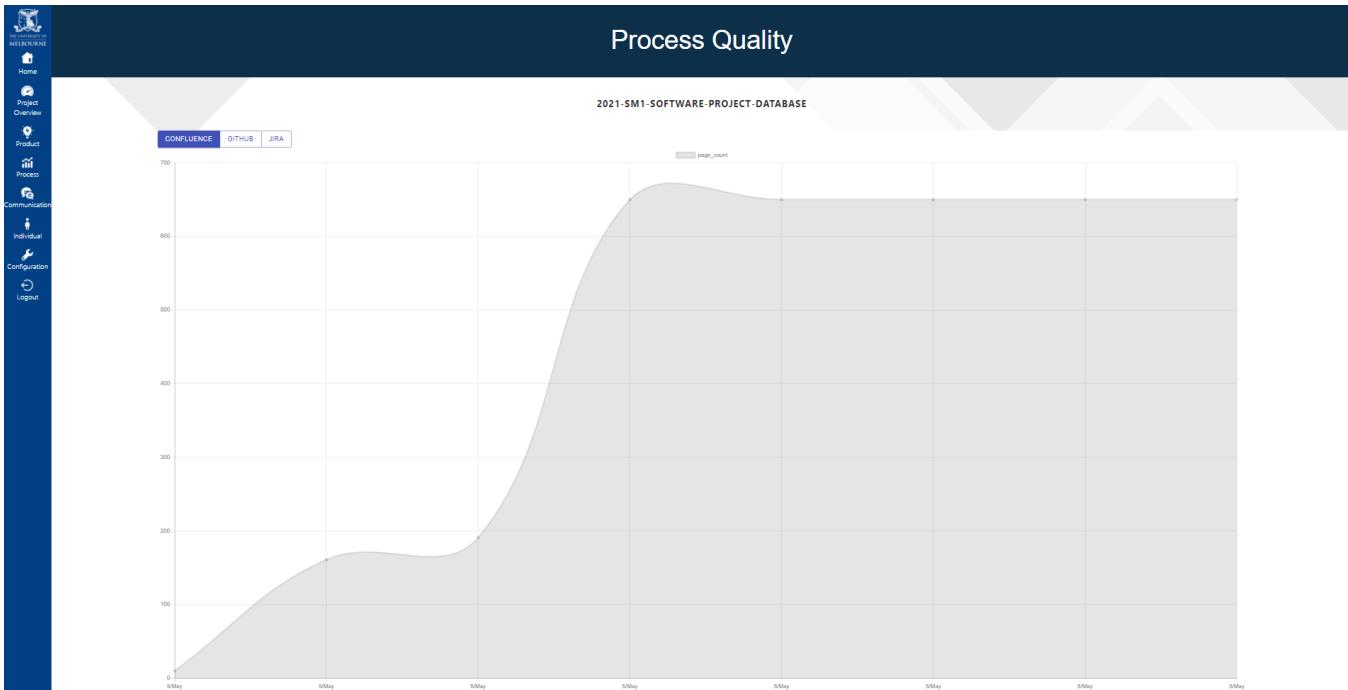
Configuration

Logout

Process Quality

TEAM JIRA

- Current:



4. Product quality

- Previous:

Not implemented in the previous work;

- Current:

The screenshot shows a dark-themed web application interface. On the left is a vertical sidebar with icons and labels: Home, Project Overview, Product, Process, Communication, Individual, Configuration, and Logout. The main title "Product Quality" is at the top, followed by the subtitle "2021-SM1-SOFTWARE-PROJECT-DATABASE". Below this are two tables of code metrics.

Number of all lines	Number of classes	Number of declarable statements	Number of executable statements
111	222	0	333

Number of files	Number of functions	Number of preprocessor lines	Ratio of comment lines to code lines
0	444	0	555

5. Communication quality

- Previous:

The screenshot shows a dark-themed web application interface. On the left is a vertical sidebar with icons and labels: Project Home, Process, Communication, Individual, Configuration, and Logout. The main title "Communication Quality" is at the top. In the center, there is a green button labeled "FETCH SLACK" and a dropdown menu labeled "Choose Sprint" with "Sprint 0" selected.

- Current:

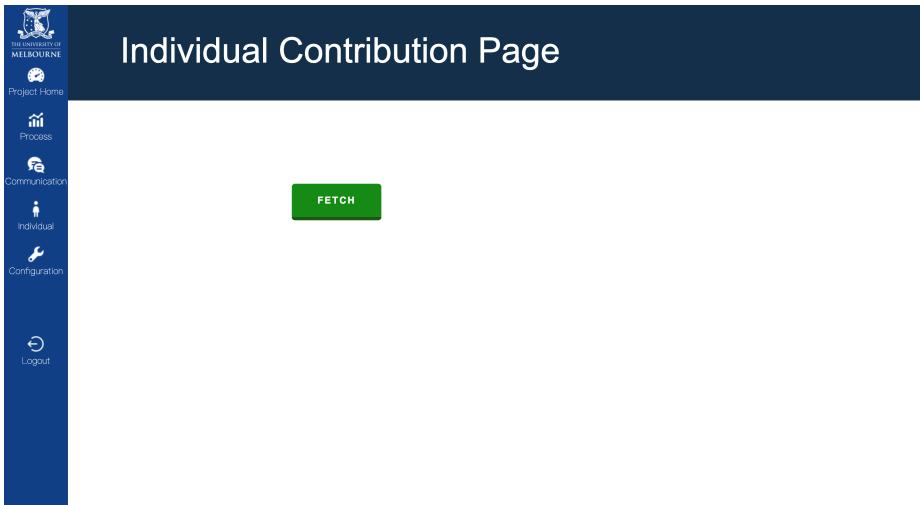
The screenshot shows a dark-themed web application interface. On the left is a vertical sidebar with icons and labels: Home, Project Overview, Product, Process, Communication, Individual, Configuration, and Logout. The main title "Communication" is at the top, followed by the subtitle "2021-SM1-SOFTWARE-PROJECT-DATABASE". There are two tabs at the top: CONFLUENCE and GITHUB. The left panel displays a table of meeting logs:

Meeting Name	Meeting Time	Meeting Minutes
Frontend 23/04/21 m...	5/May	#
Backend 23/04/21 m...	5/May	#

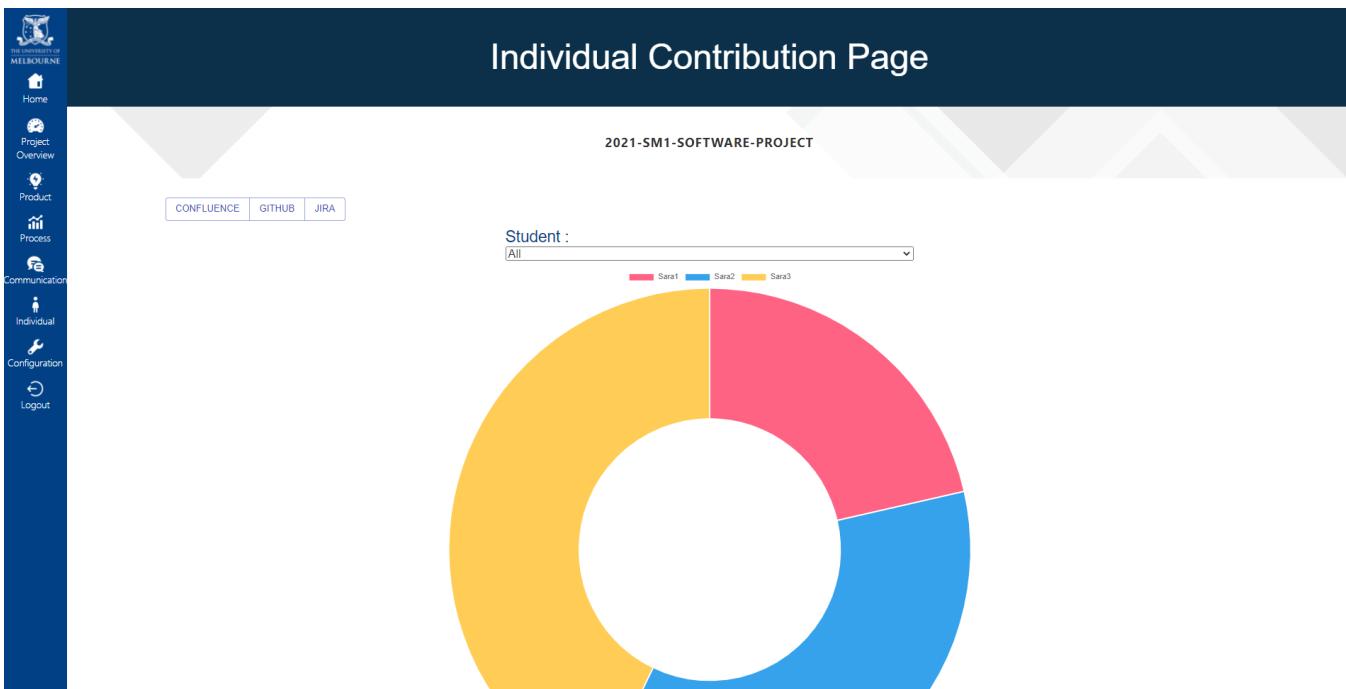
The right panel displays a line chart titled "comment_count" showing the number of comments over time. The x-axis represents time from 1 Apr to 5 May, and the y-axis represents the count from 0 to 60. The chart shows a sharp increase in comments starting around April 25th.

6. Individual contribution page

- Previous:

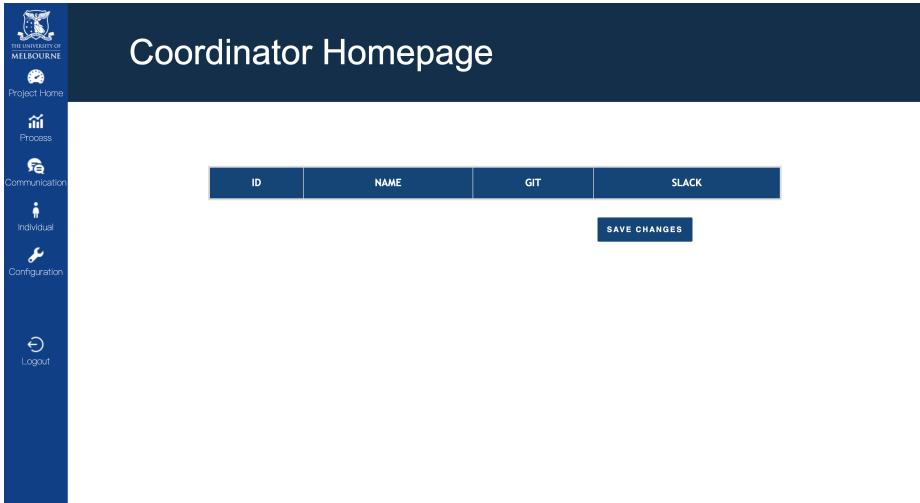


- Current:



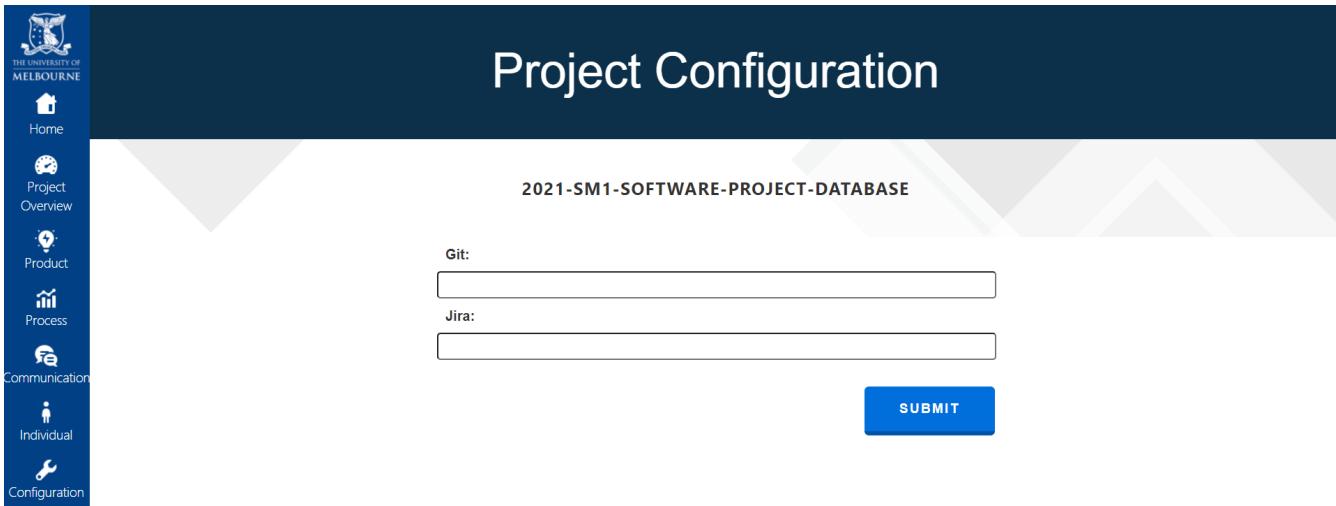
7. Configuration

- Previous:



The Coordinator Homepage interface features a dark blue header with the title "Coordinator Homepage". On the far left is a vertical sidebar with the University of Melbourne logo at the top. Below the logo are several navigation icons with labels: "Project Home" (house icon), "Process" (chart icon), "Communication" (speech bubble icon), "Individual" (person icon), and "Configuration" (gear wrench icon). At the bottom of the sidebar is a "Logout" button with a user icon. The main content area contains a table with columns "ID", "NAME", "GIT", and "SLACK". Below the table is a "SAVE CHANGES" button.

- Current:



The Project Configuration interface has a dark blue header with the title "Project Configuration". Below the header is a section titled "2021-SM1-SOFTWARE-PROJECT-DATABASE". The main body contains two input fields: one for "Git" and one for "Jira", both with placeholder text. To the right of these fields is a "SUBMIT" button.

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, [4 Prototypes](#);
2. Implemented the data fetching and display functionality using Redux and visualization components.

In general, except for the header and the navigation bar, we discarded the UI implementation done by the previous team and wrote the UI from scratch using new components and styling.

The code for fetching the data from API and passing the data to each web page was partially re-used with cleaning, refactoring and adding. See details in [Coding update - Front-end team](#).

The UI improvements based on the work done by the previous team can be perceived in the [Design update - Front-end team](#).

User stories checklist

Epic	ID	As a	I want	So that	Priority	Story Points	Completed (Front-end)
Home Page	U S _1	Coodinator	To import student projects from confluence	I can assess students' work	Must have	2	
	U S _2	Coodinator	To see those imported projects	I can view each project	Must have	2	
	U S _3	Coodinator	to delete a project that does not belong to my subject.	I can only keep the projects from my subject(s).	Could have	1	
Product Quality	U S _4	Coodinator	to see some metrics from a third-party code analysis tool	I can assess the product quality of each team based on the summary	Must have	3	
	U S _5	Coodinator	to view up to date statistical data	I can view the progress they made each week	Must have	4	
Process Quality	U S _6	Coodinator	to see the process quality summary of my project teams	I can assess the process quality of each team based on the summary	Must have	12 (break down)	
	U S _7	Coodinator	to view graphical data on a project team's Confluence activities	I can measure and assess a project team's process quality in terms of task management	Must have	4	
	U S _8	Coodinator	to view graphical visualisations on a project team's JIRA task activities (i.e., total vs. completed vs. remaining tasks)	I can measure and assess a project team's process quality in terms of task management	Must have	4	
	U S _9	Coodinator	to view graphical visualizations on the number of code commits with proper code reviews and Continuous Integrations tests in GitHub /Bitbucket/GitLab	I can measure and assess a project team's process quality in terms of codebase management	Must have	4	
Communication Quality	U S _10	Coodinator	to see the communication summary of my project teams	I can assess the communication of each team based on the summary	Must have	10 (break down)	
	U S _11	Coodinator	the system to filter out irrelevant messages (emoji only messages) when gathering accessible communication data for each project team	I can measure and assess a project team's communication quality in Slack accurately by excluding irrelevant messages.	Should have	2	
	U S _12	Coodinator	to view statistical data on the total number of messages sent over assessable Slack channels for each project team	I can measure and assess a project team's communication quality in terms of chat frequency in Slack	Should have	2	
	U S _13	Coodinator	to view each meeting information including meeting minutes for each project team	I can measure and assess a project team's meeting frequency and meeting quality	Must have	3	
	U S _14	Coodinator	to view statistical data on comments on GitHub	I can measure and assess a project team's communication quality in GitHub	Must have	3	
Individual Contribution	U S _15	Coodinator	to see the individual contribution summary of each team member in my project teams	I can assess the individual contribution of each team member based on the summary	Must have	13 (break down)	
	U S _16	Coodinator	to view statistical data on the number of pull requests involving peer reviews made per team member in GitHub/Bitbucket/GitLab	I can assess the individual contribution of each team member based on their pull requests	Must have	3	

	U S = 47	Coo rdin ator	to view statistical data on the number of lines of executable code made per team member	I can assess the individual contribution of each team member based on the number of lines of code they have written.	Coul d have	5	
	U S = 18	Coo rdin ator	to view statistical data on the number of completed tasks/tickets on JIRA by each team member	I can assess the individual contribution of each team member based on their task management /completion	Mus t have	3	
	U S = 19	Coo rdin ator	to view statistical data on the number of complete tasks from JIRA by each team member	I can assess the individual contribution of each team member based on their individual contribution	Mus t have	3	
	U S = 20	Coo rdin ator	to view statistical data on publishing and editing activity made by each team member on Confluence	I can assess the individual contribution of each team member based on their documentation work	Mus t have	8	
	U S = 24	Coo rdin ator	to view statistical data on meeting attendance of each team member	I can assess the individual contribution of each team member based on their attendance in meetings	Can have	5	
Configuration	U S = 22	Coo rdin ator	to change the URLs of JIRA, Confluence and Git.	I can make any changes to the links of JIRA, Confluence and Git in case some project teams might want to change their JIRA/Confluence /Git destination.	Mus t have	3	

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.

The screenshot shows the Mockoon application interface. At the top, there's a navigation bar with 'Application', 'Actions', 'Import/export', 'Tools', and 'Help'. Below that, the URL is set to '0.0.0.0 : 3200' and the prefix is '/'. A search bar contains the path '/api/v1/confluence/COMP9008221SM1SP/meeting_minutes'. The main area displays a list of API endpoints under the 'SP' category:

- /api/v1/team/COMP90082... **GET**: Serve a file dynamically depending on ...
- /api/v1/confluence/spaces... **GET**: Serve a file dynamically depending on ...
- /api/v1/git/COMP9008220... **GET**: Serve a file dynamically depending on ...
- /api/v1/jira/COMP900822... **GET**: Serve a file dynamically depending on ...
- /api/v1/confluence/COMP... **GET**: Serve a file dynamically depending on ...
- /api/v1/git/COMP9008220... **GET**: Serve a file dynamically depending on ...
- /api/v1/confluence/spaces... **GET**: Serve a file dynamically depending on ...
- /api/v1/git/COMP9008220... **GET**: Serve a file dynamically depending on ...
- /api/v1/jira/COMP900822... **GET**: Serve a file dynamically depending on ...
- /api/v1/team/config **POST**
- /api/v1/individual/git/CO... **GET**
- /api/v1/confluence/import... **GET**
- /api/v1/team/import **POST**

Below the list, there's a 'Status & Body' tab selected. It shows a status of '200 - OK' and a count of '0'. A note says 'Templating is also supported in file path'. Under the 'Body' section, the Content-Type is 'text/html' and the body content is:

```
1 {
2   "code":0,
3   "message":"success",
4   "data": [
5     {
6       "title": "Meeting Munite",
7       "link": "https://confluence.cis.unimelb.edu.au:8443/pages/viewpage
8       .action?spaceKey=COMP900822021SM1SP&title=API+Spring1"
9     },
10    {
11      "title": "Meeting Munite",
12      "link": "https://confluence.cis.unimelb.edu.au:8443/pages/viewpage
13      .action?spaceKey=COMP900822021SM1SP&title=API+Spring1"
14    },
15    {
16      "title": "Meeting Munite",
17      "link": "https://confluence.cis.unimelb.edu.au:8443/pages/viewpage
18      .action?spaceKey=COMP900822021SM1SP&title=API+Spring1"
19    }
20  ]
```



8.2.3 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	<ul style="list-style-type: none">• The tasks the team have done in last week• The tasks the team plans to finish next week
Give feedback and Suggestions	Jirat Pasuksmit	<ul style="list-style-type: none">• 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 Sprint 1 phase:

Who	What have you done	What you plan to do next
Zhang Sai	<ul style="list-style-type: none">• Set up local development environment for developing homepage UI and run• learn react knowledge• started to develop some feature via react in VSC	<ul style="list-style-type: none">• try to complete homepage UI development• learn Mockoon tool
Ruofan Zhang	<ul style="list-style-type: none">• read the previous code and learn how to use react framework	<ul style="list-style-type: none">• modify the ui of process quality webpage and offer public components for the team

Jingyu Li	<ul style="list-style-type: none"> Set up the local development environment with VSCode and install node.js run the project in the local machine learn React 	<ul style="list-style-type: none"> continue learning React Doing UI for Project Configure Page
Sarah Sultan	<ul style="list-style-type: none"> Set up the local dev environment with VS Code and run the front-end project on my machine. 	<ul style="list-style-type: none"> Figure out the project and learn how to use the React framework and JavaScript syntax.
Yuhang Xie	<ul style="list-style-type: none"> confirmed the tasks need to be done in Sprint 1 Review the code from last team 	<ul style="list-style-type: none"> Review HTML, JavaScript and CSS Learn react, node.js
Zixin Ye	<ul style="list-style-type: none"> Set up the environment for future development Learn React 	<ul style="list-style-type: none"> Develop UI of communication page
Fengrui Zhang	<ul style="list-style-type: none"> Set up local development environment for developing and run the previous project learn to code using react 	<ul style="list-style-type: none"> Develop a static UI of product quality page
Boyang Sun		
Zisheng Cheng	<ul style="list-style-type: none"> Do the deployment read code 	<ul style="list-style-type: none"> Test the data base related function Try to write some features
Jingdan Cui	<ul style="list-style-type: none"> Do the deployment Find some problems, because of the directory structure, during deployment and report them to the backend team 	<ul style="list-style-type: none"> Try to finish some functions before next week meeting
Haoyu Qin	<ul style="list-style-type: none"> Test out deployment environment Legacy code review 	<ul style="list-style-type: none"> Learn Jira API
Jinzhe Shan		
Pin Wang	<ul style="list-style-type: none"> confirm new APIs with front-end 	<ul style="list-style-type: none"> Finalize API document
Fu Xie	<ul style="list-style-type: none"> Set up the environment for future development 	<ul style="list-style-type: none"> Try to deploy the code Review the code from the previous team
Chongjing Zhang	<ul style="list-style-type: none"> Set up the environment for future development 	<ul style="list-style-type: none"> Try to deploy the code Review the code from the previous team

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	<ul style="list-style-type: none">• The tasks the team have done in last week• The tasks the team plans to finish next week
Give feedback and Suggestions	Jirat Pasuksmit	<ul style="list-style-type: none">• the Backend team should speed up the progress

Tasks completion in the Sprint 1 phase:

Who	What have you done	What you plan to do next
Zhang Sai	<ul style="list-style-type: none">• Show complete Homepage UI based on prototype design• Showed UI data (hardcoding data)• Learn about how to set local API using Mockoon	<ul style="list-style-type: none">• Improve UI according to Jirat's suggestions• Try to deployment local API via mock data• Learn redux
Ruofan Zhang	<ul style="list-style-type: none">• done the ui modification of process quality webpage• helped teammates to solve problems• offered public components for the team• set git/github branching guidelines	<ul style="list-style-type: none">• implement the api data fetching part of the process quality page• offer testing tools and mock data for the team
Jingyu Li	<ul style="list-style-type: none">• Complete UI for Project Configure Page based on previous design	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">▪ Figure out the project and learn how to use the React framework and JavaScript syntax.▪ Create Sprint 1 Documentation in confluence.	<ul style="list-style-type: none">▪ Deliver the UI of the "Individual Contribution Page" according to the prototype.▪ Update Sprint 1 Documentation in confluence.
Yuhang Xie	<ul style="list-style-type: none">▪ Reviewed html, javascript and css▪ Learned React, node.js▪ Finished UI for project homepage	<ul style="list-style-type: none">▪ Deploy local Mockoon API to test frontend page▪ Write test data in json file▪ Modify API files for project homepage

Zixin Ye	<ul style="list-style-type: none"> • Complete UI of communication page 	<ul style="list-style-type: none"> • Complete API of communication page • Write some mock data for testing API functions
Fengrui Zhang	<ul style="list-style-type: none"> • Show complete Product Quality page UI based on prototype design • Showed UI data (hardcoding) 	<ul style="list-style-type: none"> • Learn about how to set local API to test front-end page using Mockoon • Finish the API of code matrix
Boyang Sun	<ul style="list-style-type: none"> • Discuss with group members to solve the implementation issues • Design DBs 	<ul style="list-style-type: none"> • Migrate Database • Implement API "Import Project in Batch" • Conduct back-end team progress tracking
Zisheng Cheng	<ul style="list-style-type: none"> • Test the api from Github • read related code 	<ul style="list-style-type: none"> • Implement some APIs • complete its test
Jingdan Cui	<ul style="list-style-type: none"> • 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. 	<ul style="list-style-type: none"> • Discuss with the backend group about our DB • Talk to the frontend team to ensure the consistency
Haoyu Qin	<ul style="list-style-type: none"> • Re-organized & Re-implemented legacy API code 	<ul style="list-style-type: none"> • Implement new Jira APIs
Jinzhe Shan		
Pin Wang	<ul style="list-style-type: none"> • Finish API document 	<ul style="list-style-type: none"> • Implement some APIs
Fu Xie	<ul style="list-style-type: none"> • Deployed the system on my local pc • Confirmed what tasks I need to do • Tested some of the apis (for Jira part) 	<ul style="list-style-type: none"> • Communicate with my teammate and implement some Jira APIs
Chongjing Zhang	<ul style="list-style-type: none"> • Update Trello user story • Deploy system • Test some previous apis • learn django frame • Discuss with Jingdan Cui about overlapping tasks 	<ul style="list-style-type: none"> • Implement APIs • Design db table

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	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• Try to do the integration test before 30th April 2021

Tasks completion in the Sprint 1 phase:

Who	What have you done	What you plan to do next
Zhang Sai	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• Improved Homepage UI based on Client's suggestions• Complete API setting with backend
Ruofan Zhang	<ul style="list-style-type: none">• Done the api fetching functionality for process quality webpage• offered testing tools and mock data for the team• helped teammates to solve problems	<ul style="list-style-type: none">• list tasks to do in the next sprint and assign them to each team member according to their workload in Sprint 1
Jingyu Li	<ul style="list-style-type: none">• Finish fetch data from local API with the help of team leader for Project Configure Page	<ul style="list-style-type: none">• Add new functions like setting username and password for Project Configure Page based on suggestions from client.
Sarah Sultan	<ul style="list-style-type: none">• Deliver the UI of the "Individual Contribution Page" according to the prototype.• Update Sprint 1 Documentation in confluence.• Local API "Individual Contribution Page"	<ul style="list-style-type: none">• Test API• Update Sprint 1 Documentation in confluence.

Yuhang Xie	<ul style="list-style-type: none"> ▪ Deployed local API using Mockoon ▪ Make test data in json file ▪ Modify the API files for project homepage 	<ul style="list-style-type: none"> ▪ Modify the student profile in table ▪ Optimalise the table size and pop up the student member list.
Zixin Ye	<ul style="list-style-type: none"> • Complete UI and all functions required to interact with backend 	<ul style="list-style-type: none"> • Adjust any changes according to any changes from backend team/client • Help other people in frontend team out if necessary
Fengrui Zhang	<ul style="list-style-type: none"> • Complete all functions required to interact with the backend for Product Quality Page • redesign the code matrix form • Passed the Mockoon test 	<ul style="list-style-type: none"> • Update for the documentation • Find a better method to visualize the code matrix • Make the banner interactive
Boyang Sun	<ul style="list-style-type: none"> • Complete the "Import Projects in Batch" API • Check the back-end team progress 	<ul style="list-style-type: none"> • Prepare for the documentation • Do the assignment submission • Work allocation for submission
Zisheng Cheng	<ul style="list-style-type: none"> • Complete the git APIs and the corresponding database design • Test API 	<ul style="list-style-type: none"> • Update Sprint 1 Documentation
Jingdan Cui	<ul style="list-style-type: none"> • Complete GET_IMPORTED_PROJECT function • Complete GET_MEETING_MINUTES function • Complete UPDATE_MEETING_MINUTES function • Using POSTMAN to test the functions above 	<ul style="list-style-type: none"> • Update Sprint 1 Documentation • Constantly update existing functions
Haoyu Qin	<ul style="list-style-type: none"> • Get Jira TO_DO/IN_PROGRESS/DONE Count by Time • Get Individual Contribution on Jira DONE Count Tickets 	<ul style="list-style-type: none"> • Update Sprint 1 Documentation • Refine behavior of Get Jira TO_DO/IN_PROGRESS /DONE Count by Time
Jinzhe Shan		
Pin Wang	<ul style="list-style-type: none"> • Complete the implementation of 3 APIs and the corresponding database design • Test API 	<ul style="list-style-type: none"> • Modify some business logic.
Fu Xie	<ul style="list-style-type: none"> • Tested some of the APIs from the previous team • Set up models to create tables for storing data • Complete the API Set Git_url and Jira_url • Work with my teammate on other APIs 	<ul style="list-style-type: none"> • Refine behavior of Get Jira TO_DO/IN_PROGRESS /DONE Count by Time • Constantly update existing functions • Update Sprint 1 Documentation
Chongjing Zhang	<ul style="list-style-type: none"> • Finish git_commit api • Finish Git commit count by date api • Finish GitCommitCount table • Connect with Relation table • Implement some useful functions • Update test cases 	<ul style="list-style-type: none"> • Update Sprint 1 Documentation • Improve code and logic • Use postman to present data example

8.2.4 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	
	5.2 Sprint 1: Backlog	Boyang Sun	Pin Wang Jinzhe Shan Jingdan Cui
6 API documents		Boyang Sun	Pin Wang
	6.1.5 Jira	Boyang Sun	Pin Wang
	6.1.6 Git	Boyang Sun	Pin Wang
	6.9 Issues to be Fixed	Boyang Sun	
6.2 Sprint 1: API		Pin Wang	Pin Wang Fu Xie Boyang Sun
	6.2.1 Acceptance Test	Boyang Sun	Jingdan Cui Pin Wang Boyang Sun Jinzhe Shan
	6.2.2 Confluence API Design	Boyang Sun	
	6.2.3 Database Design	Boyang Sun	Boyang Sun Pin Wang Chongjing ZHANG
	6.2.5 Mock API Construction Instructions	Boyang Sun	
	6.2.6 Workload Separation	Boyang Sun	Boyang Sun
	Data Sample	Jingdan Cui	Jingdan Cui Pin Wang Chongjing ZHANG
7 Deployment		Boyang Sun	Jinzhe Shan Pin Wang
8 Sprint Summaries		Boyang Sun	Sarah Sultan A AL YAHYA
8.2 Sprint 1		Sarah Sultan A AL YAHYA	
Individual Contribution		Boyang Sun	
Review of Sprint 1 - Back-end Team		Boyang Sun	
	Coding Update	Boyang Sun	Boyang Sun Pin Jingdan Cui Fu Xie Chongjing ZHANG ZISHENG CHENG
	Design Update	Boyang Sun	ZISHENG CHENG Fu Xie Boyang Sun Jingdan Cui
	Requirements Update	Boyang Sun	Chongjing ZHANG Jingdan CuiZISHENG CHENG Jinzhe Shan
	Testing Update	Boyang Sun	
	Task Achieved in Sprint 1	Boyang Sun	
8.2.2 Review of Sprint 1 - Front-end team		Ruofan Zhang	
	Coding update - Front-end team	Ruofan Zhang	Sarah Sultan A AL YAHYA Ruofan Zhang Yuhang Xie Zixin Ye Sai Zhang Fengrui Zhang Jingyu Li
	Design update - Front-end team	Ruofan Zhang	Sarah Sultan A AL YAHYA Ruofan Zhang
	Deployment update - Front-end team	Ruofan Zhang	
	Tasks achieved - Front-end team	Ruofan Zhang	
	Testing update - Front-end team	Ruofan Zhang	Ruofan Zhang
8.2.3 Sprint 1: Ceremonies		Jingdan Cui	Ruofan Zhang
	Stand-up Meeting 15th April 2021	Jingdan Cui	Jingdan Cui
	Stand-up Meeting 22nd April 2021	Jingdan Cui	Jingdan Cui
8.2.4 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

8.2.5 Sprint 1: Individual Contribution

Back-end Team

[Contribution - Boyang Sun](#)

[Contribution - Chongjing Zhang](#)

[Contribution - Pin Wang](#)

[Contribution - Jinzhe Shan](#)

[Contribution - Fu Xie](#)

[Contribution - Jingdan Cui](#)

[Contribution - Zisheng Cheng](#)

[Contribution - Haoyu Qin](#)

Front-end Team

[Contribution - Ruofan Zhang](#)

[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 exception")
 - 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
4. Write the functions-related documentations.

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 of Individual contribution page:

- 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 for Individual contribution page:

- 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:

- Worked on [8 Sprint Summaries](#)
- Create [8.2.4 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
- 4. Update the document of Design Concept for Sprint 1**
- 5. Arrange team leaders meeting for Sprint 1**

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.

8.3 Sprint 2

- [8.3.1 Review of Sprint 2 - Back-end Team](#)
- [8.3.2 Review of Sprint 2 - Front-end team](#)
- [8.3.3 Sprint 2: Ceremonies](#)
- [8.3.4 Sprint 2: Documentation](#)
- [8.3.5 Sprint 2: Individual Contribution](#)

8.3.1 Review of Sprint 2 - Back-end Team

[Coding Summary - Back-end team S2](#)

[Deployment Report - Back-end team S2](#)

[Design Update - Back-end team S2](#)

[Handover Report- Back-end team S2](#)

[Requirement Update - Back-end team S2](#)

[Tasks achieved - Back-end team S2](#)

[Testing Report - Back-end team S2](#)

Coding Summary - Back-end team S2

Confluence

Get Imported Projects

- Pull Request: https://github.com/pete965/COMP90082_Software_Project_Database_Backend/pull/43
- Edited Function:
 - get_imported_project

Get Meeting Minutes

- Pull Request: https://github.com/pete965/COMP90082_Software_Project_Database_Backend/pull/44
- Added Functions:
 - insert_meeting_minutes
- Edited Functions:
 - update_meeting_minutes

Get Individual Contribution

Git

- Git Commit Count by Date API
 - Speed up visit time from more than 30s to less than 1s
- Git Authentication API
 - Verify user's identity
 - Restrict invalid access
 - Allow to visit private repository
- Improve Git Pull and Git Commit APIs
 - Speed up
 - Remove unnecessary and irrelevant data request
- Git Individual commit API
 - Speed up
- Daily update

Jira

- Pull Request: https://github.com/pete965/COMP90082_Software_Project_Database_Backend/pull/83
- Jira To_Do/In_Progress/Done Count By Time
 - implement jira-agile-metrics to retrieve data, speed up the time
 - auto update data when user configure Jira URL
 - change the time format stored in the database from YYYY-MM-DD to unix time
- Jira Get Individual Contribution on Jira Count By Tickets
 - auto update data when user configure Jira URL
 - get the data by provided URL, rather than Jira key
- Set Github and Jira URL
 - Change the time format stored in database as required
 - Enable auto updating data when users set up the URL
- Daily update

Deployment Report - Back-end team S2

Private Ec2 Deployment: Finished 28th April

Client VM Deployment: Finished 28th May

Design Update - Back-end team S2

Confluence

1. Get Meeting Minutes

- Older 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 pages from the Confluence space, and the meeting minutes data will be stored and updated in the database.
- New design: The meeting minutes will be imported and stored into the database when the coordinator imports a project. And coordinator can see the meeting minutes immediately instead of waiting 24 hours for updates.

Git

1. Get Git Commit Count by Date

- Older design:
 - Use POST method.
 - Every request data are fetched from GitHub directly which is so slow.
 - Request a lot of unnecessary and irrelevant data.
 - Not beautiful.
 - Can't access private repository.
- New design:
 - Use GET method
 - Daily update
 - Use database to cache in advance to greatly improve performance.
 - Add git authentication function to access private repository and restrict invalid access.
 - Sort and organize data according to past 30 days commit count so that x-time-axis will not be too long and can display more beautiful
- Some gitutil tools

2. Get Git Metrics by SciTools Understand [Jinzhe Shan](#)

- Older design:
 - Use Python API + cmd command lines to get code metrics
 - It works on a single python but meets a QT thread exception when it runs in our Django Server which is really hard to solve because it is the conflict between Django Structure and Understand Tool
- New design:
 - Extract Understand Part from Django to a single python file
 - Use JSON file to store code metrics of projects
 - Read corresponding JSON file in Django and package them to Git MetricsAPI

3. Get Git individual Commit

- Older design:
 - Not implemented
- New design:
 - Daily update
 - Use database to cache in advance to greatly improve performance.
 - Exception handling

Confluence

Jira

1. Get Jira TO_DO/IN_PROGRESS/DONE Count by Time

- Older design:
 - Every request data are fetched from Jira directly which is so slow.
 - Request a lot of unnecessary and irrelevant data.
- New design:
 - Daily update
 - Use database to cache in advance to greatly improve performance.
 - Implement jira-agile-metrics to retrieve data from Jira
 - Auto update when users set up Jira URLs

2. Get Individual Contribution on Jira DONE Count Tickets

- Older design:
 - Get the project name. then use it to retrieve data
 - Every request data are fetched from Jira directly which is so slow.
 - Time format in data base is yyyy-mm-dd
- New design:
 - Daily update

- Use database to cache in advance to greatly improve performance.
- Use Jira URL instead of project to retrieve data as required
- Auto update when users set up Jira URLs
- Change the time format in database to unix time as required

3. Set Github and Jira Url

- Older design:
 - Github and JiraUrls are stored in different tables
- New design:
 - Store Jira and GithubUrls in the same table

Requirement Update - Back-end team S2

Confluence

Git

- Git Commit Count by Date
 - Only show last 30 days

Jira

- Set Github and Jira Url
 - Store Jira and Github URLs in the same table

Tasks achieved - Back-end team S2

Confluence

Get Imported Projects

- Previous system: Search each data in project_coordinator_relation table in the database. Select and only return all the projects which are imported by the coordinator whose coordinator id is 1.
- Our work: Select and return all the projects which are imported by the coordinator who currently login.
- Related User Story: [1.5 User stories US - 2](#)

Daily Update Meeting Minutes

- Previous system: Fetch the relevant meeting minutes data from Confluence and store them into the database every 24 hours. However, the meeting information cannot be retrieved when the coordinator imports a project.
- Our work: Not only the meeting information is updated daily, but also the coordinator can see the meeting minutes as long as he/she imports a project.
- Related User Story: [1.5 User stories US - 13](#)

Git

Get Git Commit Count by Date

- Previous system: Search each data in project_coordinator_relation table in the database.
- Our work: Select and return project commit data chosen by currently login coordinator.
- Implement daily update function
- Add git authentication function
- Fix bugs of Get Git Commits APIs
- Related User Story: [1.5 User stories US - 9](#)

Get Git Metrics by SciTools Understand Jinzhe Shan

- Researched SciTools Understand software, which is a kind of professional code analysis tool.
- To implement Git Metrics API there are 4 steps
 - Deploy the Understand software as a sub-system of our project on our Linux Server, which can refer the document :[7.2 Scitools Understand Deployment](#)
 - Get software source code using git methods.
 - Use Understand Python API & Und Command Lines to generate indexes for describing the product quality of a software project and store them in JSON files
 - Read corresponding JSON file in Django and package them to Git Metrics API
- Fix bugs of Get Git Metrics and Get Git Commits APIs

Get Git Individual Commit

- Previous system: Not implemented
- Our work: Shows the total commit count of each student.
- Implement daily update function
- Exception handling
- Related User Story: [1.5 User stories US - 15](#)

Jira

Get Jira TO_DO/IN_PROGRESS/DONE Count by Time

- Previous system: Search each data in project_coordinator_relation table in the database. Select and only return all the projects which are imported by the coordinator whose coordinator id is 1. Get one record per request from Jira, slow performance.
- Our work: Select and return all the projects which are imported by the coordinator who currently login. Implement jira-agile-metrics to fetch data, significantly increase the efficiency; Auto update when users setup URLs and everyday.
- Related User Story: [1.5 User stories US - 6](#)

Get Individual Contribution on Jira DONE Count Tickets

- Previous system: Take project name and use it to fetch data
- Our work: Take Jira URLs and use it to fetch data; Auto update when users setup URLs and everyday.
- Related User Story: [1.5 User stories US - 18](#)

Set Github and Jira Url

- Previous system: Store Github and Jira URLs in different tables
- Our work: Store Github and Jira URLs in the same tables; Auto update when users setup URLs and everyday.
- Related User Story: [1.5 User stories US - 22](#)

Testing Report - Back-end team S2

Refer to : [9.4 Testing Progress](#)

8.3.2 Review of Sprint 2 - Front-end team

[Coding summary - Front-end team S2](#)

[Deployment report - Front-end team S2](#)

[Design update - Front-end team S2](#)

[Handover report - Front-end team S2](#)

[Tasks achieved - Front-end team S2](#)

Coding summary - Front-end team S2

58 changed files with 2,985 additions and 2,169 deletions since last sprint

Details of all changed files can be found in https://github.com/ruofzhang/COMP90082-SM1-2021-SP-Frontend/compare/COMP90082_2021_RLSE_SP_1.0...COMP90082_2021_RLSE_SP_FINAL

Major changes can be found in the following files:

1. Functionality implementation
 - CoordinatorHomePage.js
 - LoginPage.js
 - ProjectSettingsPage.js
2. UI re-design
 - navbar.js
 - navbar.module.css
3. API data fetching and Redux implementation
 - user.actions.js
 - store.js
 - user.reducer.js
 - src/index.js
 - src/_reducers/index.js
4. Public tools
 - Alert.js
 - DonutChart.js
 - DropdownMenus.js
 - LineChart.js
 - ReverseTable.js
 - formatDonutChartData.js
 - formatDrawerData.js
 - formatImportedProjectData.js
5. Routing configuration
 - Home.js
 - ProtectedRoute.js
 - PublicRoute.js
6. Code cleaning
 - ProjectHomePage.js
 - CommunicationQualityPage.js
 - ProductQualityPage.js

Deployment report - Front-end team S2

The app has been deployed on the client's server using Nginx and it can be accessed by visiting <http://sp.unimelb.site>

Detailed instructions for deployment:

Step 1 git clone <https://github.com/ruofzhang/COMP90082-SM1-2021-SP-Frontend.git>

Step 2 cd COMP90082-SM1-2021-SP-Frontend

Step 3 Under the project directory, enter the command "npm run build"

A folder named "build" will be generated

.git	2021/5/28 17:14	文件夹
build	2021/5/25 15:57	文件夹
data sample	2021/5/28 17:14	文件夹
docs	2021/4/1 23:10	文件夹
node_modules	2021/5/19 20:15	文件夹
prototypes	2021/4/1 23:05	文件夹
public	2021/4/29 12:27	文件夹
src	2021/5/27 13:32	文件夹
tests	2021/3/30 23:51	文件夹
.gitignore	2021/3/30 23:51	Git Ignore 源文件 1 KB
jest.config	2021/3/30 23:51	JavaScript 源文件 7 KB
package	2021/5/20 1:22	JSON 源文件 2 KB
package-lock	2021/5/19 20:15	JSON 源文件 746 KB
README	2021/5/27 13:32	Markdown File 20 KB
webpack.config	2021/5/18 3:40	JavaScript 源文件 1 KB

And the content is like

icons	2021/5/25 15:56	文件夹
static	2021/5/25 15:57	文件夹
asset-manifest	2021/5/25 15:57	JSON 源文件 2 KB
favicon	2021/4/29 12:27	JPG 文件 127 KB
index	2021/5/25 15:57	Chrome HTML D... 3 KB
logo192	2021/3/30 23:51	PNG 文件 6 KB
logo512	2021/3/30 23:51	PNG 文件 10 KB
manifest	2021/3/30 23:51	JSON 源文件 1 KB
precache-manifest.dc5c7f228455dc3c...	2021/5/25 15:57	JavaScript 源文件 1 KB
robots	2021/3/30 23:51	文本文档 1 KB
service-worker	2021/5/25 15:57	JavaScript 源文件 2 KB

Step 4 Add these files in a zip file

Step 5 Install Nginx on the server. Suppose the OS of the server is Ubuntu 18.04, run the following commands:

```
sudo apt update
```

```
sudo apt install nginx
```

Step 6 Config Nginx. Run the following commands:

```
sudo mkdir -p /var/www/<your domain name>/html
```

This folder is root directory of the website. Put and unzip the zip file generated in Step 4 here.

```
sudo nano /etc/nginx/sites-available/<your domain name>
```

Edit and paste the following content into /etc/nginx/sites-available/<your domain name> and save

```
server
{
listen 80;
server_name <your domain name>;
index index.html;
root /var/www/<your domain name>/html;
location /
{
try_files $uri /index.html;
}
location /api/
{
proxy_pass http://<your domain name>:<back-end running port>;
proxy_set_header Host <your domain name>;
}
}
```

```
sudo ln -s /etc/nginx/sites-available/<your domain name> /etc/nginx/sites-enabled/
```

```
sudo systemctl restart nginx
```

Final step Visit <http://<your domain name>> to test if the configuration works

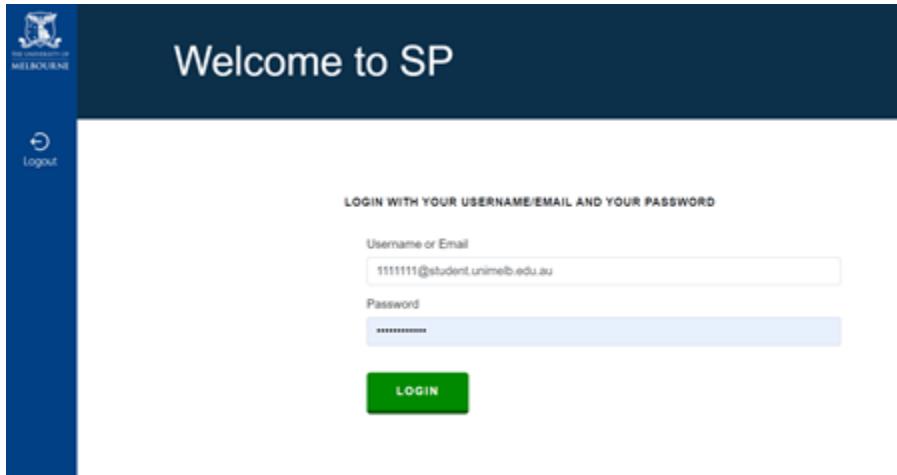
Design update - Front-end team S2

Sprint 1:

[Design update - Front-end team S1](#)

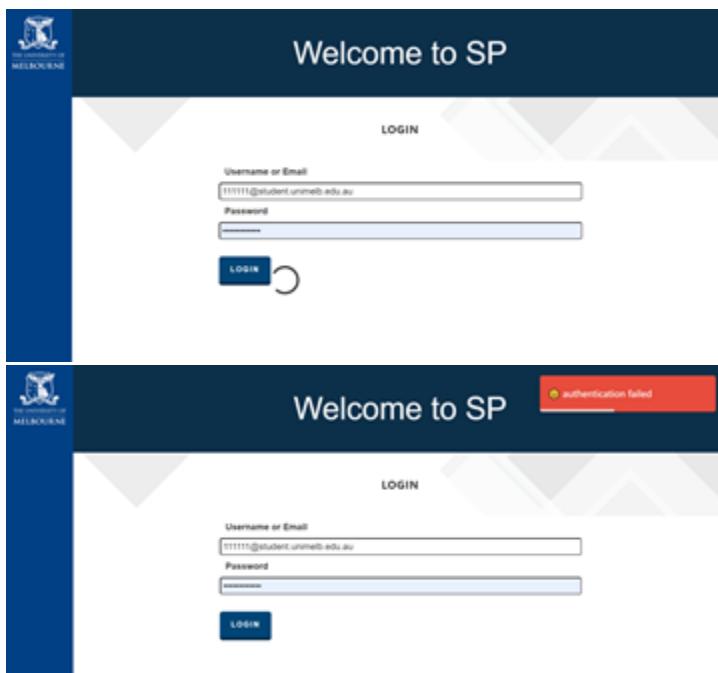
1. Login page

Previous:



The previous login page has a dark blue header with the University of Melbourne logo and the text "Welcome to SP". On the left, there's a vertical sidebar with a "Logout" button. The main area contains a "LOGIN WITH YOUR USERNAME/EMAIL AND YOUR PASSWORD" instruction, followed by two input fields: "Username or Email" containing "111111@student.unimelb.edu.au" and "Password" containing a masked password. A green "LOGIN" button is at the bottom.

Current:



The current login page design is similar to the previous one but includes a progress indicator. It features a dark blue header with the University of Melbourne logo and the text "Welcome to SP". The main area has a "LOGIN" button above the input fields. The "Username or Email" field contains "111111@student.unimelb.edu.au" and the "Password" field is masked. Below the inputs is a blue "LOGIN" button with a white circular progress icon. In the top right corner of the header, there is a red notification box with a yellow exclamation mark and the text "authentication failed".

2. Coordinator Home

Previous:

The screenshot shows a dark blue header with the University of Melbourne logo and a navigation bar on the left containing 'Home (C)' and 'Logout' buttons. The main area has a light gray background with a title 'Homepage' and a subtitle 'PROJECT MANAGEMENT'. Below this is a search bar with the placeholder 'su' and a magnifying glass icon. A table titled 'Software Resources for CIS subjects' lists four entries. Each entry includes a 'Project Imported' column, a 'CONFERENCE LINK' column with a URL, and an 'Operation' column with a 'DELETE' button.

Project Imported	CONFERENCE LINK	Operation
Motivational Modelling	https://confluence.cis.unimelb.edu.au:8443/display/Sample2	<button>DELETE</button>
Panimation	https://confluence.cis.unimelb.edu.au:8443/display/Sample4	<button>DELETE</button>
Software Resources for CIS subjects	https://confluence.cis.unimelb.edu.au:8443/display/Sample6	<button>DELETE</button>

Current:

THE UNIVERSITY OF
MELBOURNE

Home



Logout

Coordinator Home

PROJECT MANAGEMENT

Project Name	Confluence Link	Operation
ActiveBoard	https://confluence.cis.unimelb.edu.au:8443/display/SWEN900162019S1ATeam/Home	<button>DELETE</button> <button>VIEW</button>
COMP90082-2021-SM1-SP	https://confluence.cis.unimelb.edu.au:8443/display/COMP900822021SM1SP/Home	<button>DELETE</button> <button>VIEW</button>

THE UNIVERSITY OF
MELBOURNE

Home



Logout

Coordinator Ho

PROJECT MANAGEMENT

Project Name	Confluence Link
ActiveBoard	https://confluence.cis.unimelb.edu.au:8443/display/SWEN900162019
COMP90082-2021-SM1-SP	https://confluence.cis.unimelb.edu.au:8443/display/COMP900822021

COMP90082-2021-SM1-SP

Student Name	Student Email
Pin Wang	pinwang@student.unimelb.edu.au
Zixin Ye	yezy2@student.unimelb.edu.au
Jingdan Cui	jingdanc1@student.unimelb.edu.au
Jirat Pasuksmit	pasuksmitj@student.unimelb.edu.au
Ruofan Zhang	ruofzhang@student.unimelb.edu.au
Sarah Sultan A AL YAHYA	salyahya@student.unimelb.edu.au
Boyang Sun	boyangs@student.unimelb.edu.au
Chongjing ZHANG	chongjingz@student.unimelb.edu.au
Fengrui Zhang	fengrzhang@student.unimelb.edu.au
Haoyu Qin	haoyuq@student.unimelb.edu.au
Jingyu Li	jingyul5@student.unimelb.edu.au

3. Project Homepage

Previous:



THE UNIVERSITY OF
MELBOURNE

-  Project Home
-  Process
-  Communication
-  Individual
-  Configuration
-  Logout

TQ Project Homepage

GET TEAM LIST

Current:



THE UNIVERSITY OF
MELBOURNE

-  Home
-  Project
-  Product
-  Process
-  Communication
-  Individual
-  Configuration
-  Logout

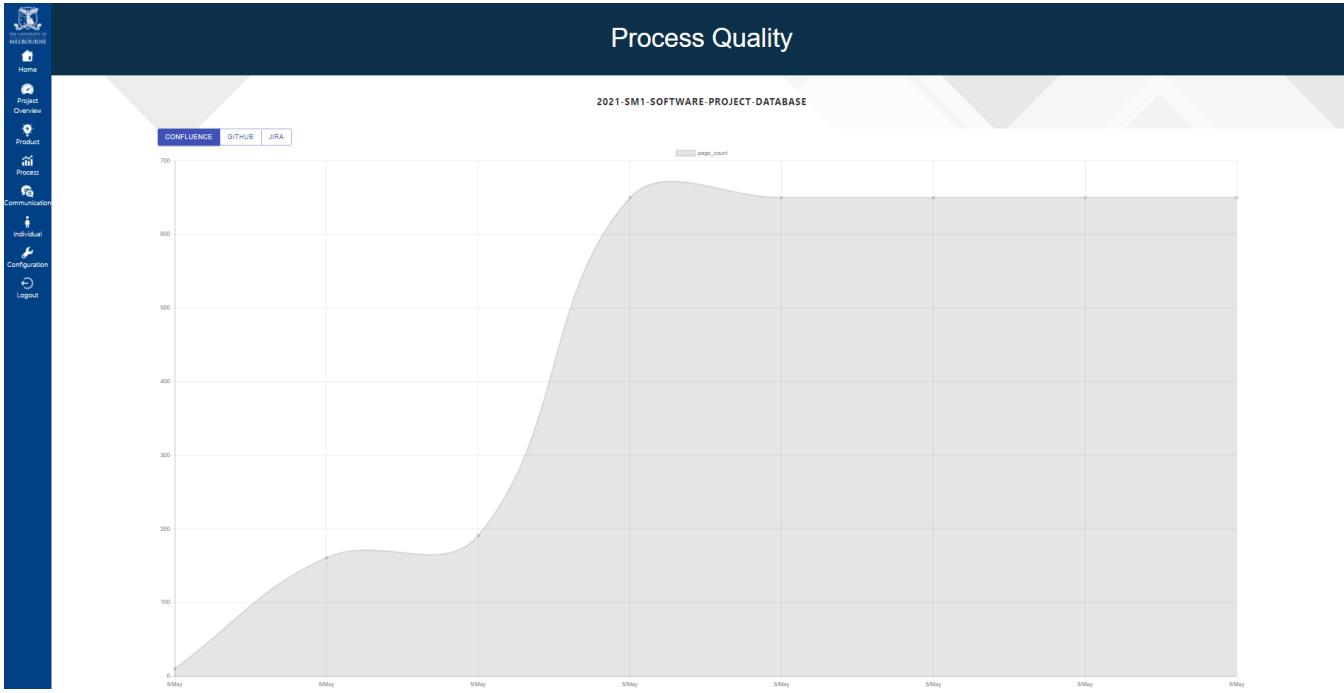
Project Overview

COMP90082-2021-SM1-SP

Name	Profile	Student ID	Email Address
Boyang Sun		boyangs	boyangs@student.unimelb.edu.au
Chongjing ZHANG		chongjingz	chongjingz@student.unimelb.edu.au
Fengrui Zhang		fengrzhang	fengrzhang@student.unimelb.edu.au
Haoyu Qin		haoyuq	haoyuq@student.unimelb.edu.au
Jingyu Li		jingyul5	jingyul5@student.unimelb.edu.au
Jinzie Shan		jinzes	jinzes@student.unimelb.edu.au
Sai Zhang		saiz	saiz@student.unimelb.edu.au
Fu Xie		xiefx	xiefx@student.unimelb.edu.au

4.Process Quality

Previous:





Product Quality

2021-SM1-SOFTWARE-PROJECT-DATABASE

Number of all lines	Number of classes	Number of declarable statements	Number of executable statements
---------------------	-------------------	---------------------------------	---------------------------------

111	222	0	333
-----	-----	---	-----

Number of files	Number of functions	Number of preprocessor lines	Ratio of comment lines to code lines
-----------------	---------------------	------------------------------	--------------------------------------

0	444	0	555
---	-----	---	-----

Current:



Product Quality

COMP90082-2021-SM1-SP

Metric	Number
--------	--------

Lines	846
-------	-----

Classes	12
---------	----

Files	12
-------	----

Functions	53
-----------	----

Comment Lines	253
---------------	-----

Comment Lines / Code Lines	0.3
----------------------------	-----

Declarable Statements	126
-----------------------	-----

Executable Statements	718
-----------------------	-----

6. Communication Page

Previous:

React App — Mozilla Firefox

localhost:3000/CommunicationQualityPage

THE UNIVERSITY OF MELBOURNE

Home

Project Overview

Product

Process

Communication

Individual

Configuration

Logout

Communication

2021-SM1-SOFTWARE-PROJECT-DATABASE

CONFLUENCE GITHUB

Meeting Name	Meeting Time	Meeting Minutes
Frontend 23/04/21 m...	5/May	#
Backend 23/04/21 m...	5/May	#

Current:

THE UNIVERSITY OF MELBOURNE

Home

Project

Product

Process

Communication

Individual

Configuration

Logout

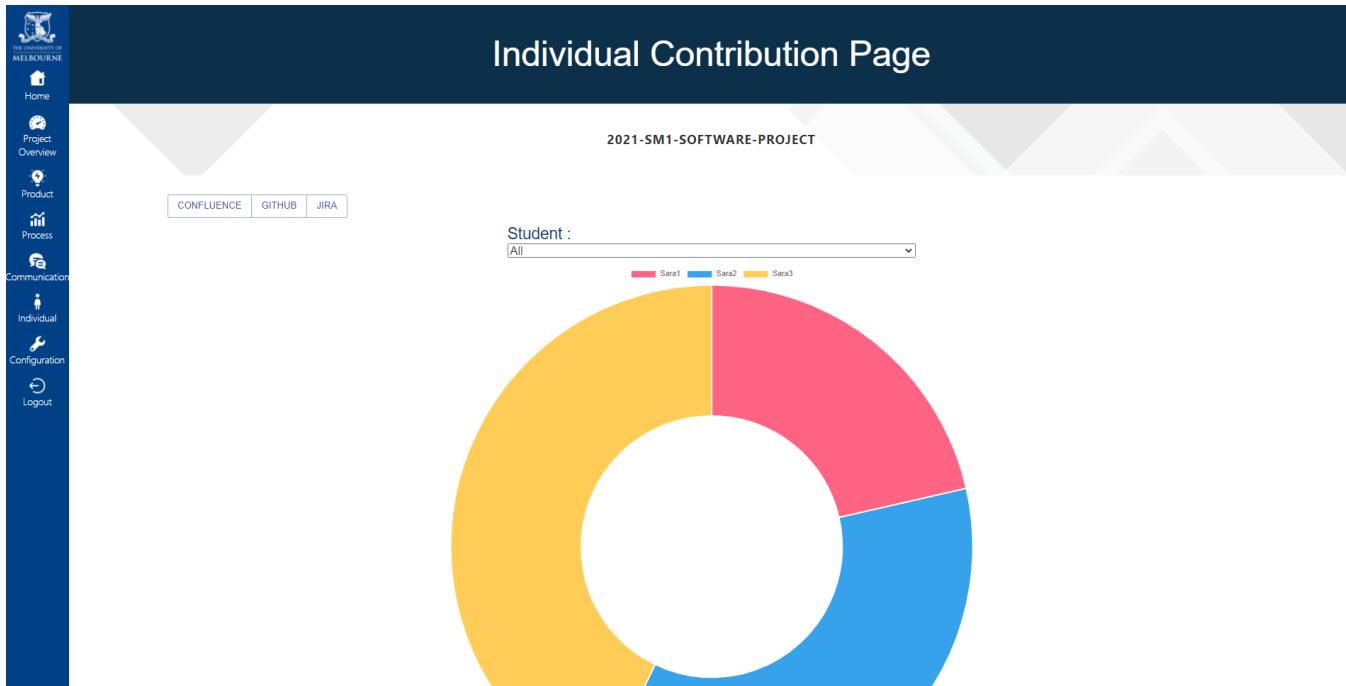
Communication

COMP90082-2021-SM1-SP

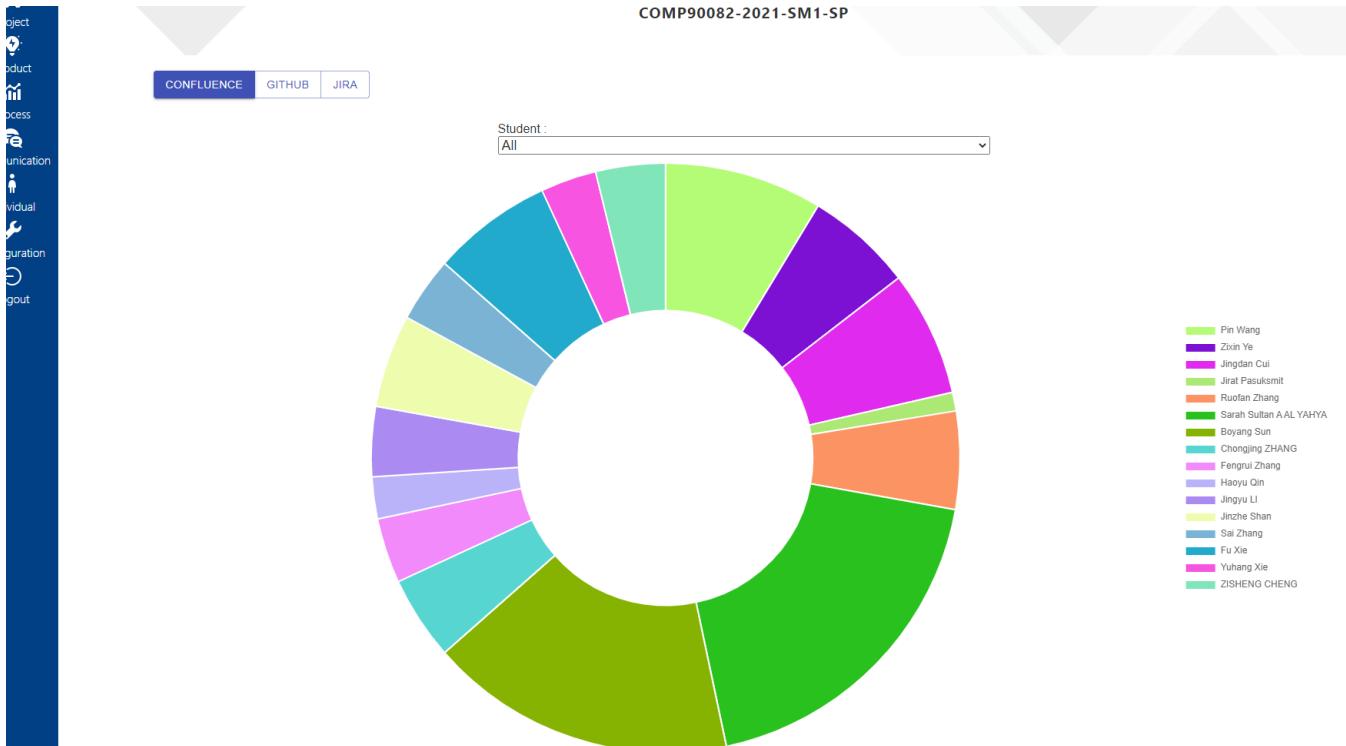
Meeting Name	Meeting Minutes
Meeting with client #1	https://confluence.cis.unimelb.edu.au:8443/pages/viewpage.action?pageId=74781618
Meeting Notes for 23rd March	https://confluence.cis.unimelb.edu.au:8443/display/COMP900822021SM1SP/Meeting+Notes+for+23rd+March
10 Meetings	https://confluence.cis.unimelb.edu.au:8443/display/COMP900822021SM1SP/10+Meetings
Meeting Notes for 28th, March	https://confluence.cis.unimelb.edu.au:8443/display/COMP900822021SM1SP/Meeting+Notes+for+28th%2C+March
Front-end Meeting Notes on 28th March -	https://confluence.cis.unimelb.edu.au:8443/pages/viewpage.action?pageId=74787344
Stand-up Meeting 18 March 2021	https://confluence.cis.unimelb.edu.au:8443/display/COMP900822021SM1SP/Stand-up+Meeting+18+March+2021
Stand-up Meeting 25 March 2021	https://confluence.cis.unimelb.edu.au:8443/display/COMP900822021SM1SP/Stand-up+Meeting+25+March+2021
Meeting Notes for 30th, March	https://confluence.cis.unimelb.edu.au:8443/display/COMP900822021SM1SP/Meeting+Notes+for+30th%2C+March
Meeting with Client #2	https://confluence.cis.unimelb.edu.au:8443/pages/viewpage.action?pageId=74788829
Meeting Notes for 1st, April(Workshop Stand-up Meeting)	https://confluence.cis.unimelb.edu.au:8443/pages/viewpage.action?pageId=74791109

7. Individual Page

Previous:



Current:



8. Configuration Page

Previous:

The screenshot shows a dark-themed web application interface. On the left is a vertical sidebar with icons and labels: Home, Project Overview, Product, Process, Communication, Individual, and Configuration. The Configuration icon is highlighted with a blue border. The main content area has a dark header with the title "Project Configuration". Below the header is a section titled "2021-SM1-SOFTWARE-PROJECT-DATABASE". It contains two input fields labeled "Git:" and "Jira:", each with a corresponding empty text input box. A blue "SUBMIT" button is located at the bottom right of the form.

Current:

The screenshot shows a dark-themed web application interface. On the left is a vertical sidebar with icons and labels: Home, Project, Product, Process, Communication, Individual, Configuration, and Logout. The Configuration icon is highlighted with a blue border. The main content area has a dark header with the title "Project Configuration". Below the header is a section titled "COMP90082-2021-SM1-SP". A blue information icon is followed by the text "Informational Notes" and "You may need to wait for a few minutes before you can view the data after finishing the configuration". The configuration form below includes fields for "Jira Url:" (with value "https://jira.cis.unimelb.edu.au:8444/projects/EEQU14/summary"), "Git Url:" (with value "https://github.com/ruofzhang/COMP90082-SM1-2021-SP-Frontend"), "Git Username:" (with value "Saiskye"), and "Git Password:" (with value "....."). A blue "SUBMIT" button is located at the bottom right of the form.

Handover report - Front-end team S2

Final deliverable

[COMP90082_2021_RLSE_SP_FINAL](#)

The app has been deployed on the client's server and it can be accessed by visiting <http://sp.unimelb.site>

Detailed deployment instructions can be found here: [Deployment report - Front-end team S2](#)

Tasks achieved - Front-end team S2

In Sprint 2, our team mainly achieved these tasks:

1. implement the login pagep0 [Fengrui Zhang](#)
2. complete the redux part of individual contribution pagep0 [Zixin Ye](#)
3. deploy the website on a serverp0 [Ruofan Zhang](#)
4. popup dialog to confirm the operation when deleting an imported projectp0 [Sai Zhang](#)
5. github username and password inputs in the project setting pagep0 [Jingyu Li](#)
6. add "view" button to the "operation" columnp1 [Sai Zhang](#)
7. different colors for line chart legendsp1 [Sarah Sultan A AL YAHYA](#)
8. prompt to indicate the space visibility is not set correctly when the length of the team student list passes a certain thresholdp1 [Yuhang Xie](#)
9. visualization of the product quality pagep2 [Sarah Sultan A AL YAHYA](#)

Note p0 is the top priority.

The code update summary is in [Coding summary - Front-end team S2](#)

The UI improvements can be perceived in the [Design update - Front-end team S2](#).

User stories checklist

Epic	ID	As a	I want	So that	Priority	Story Points	Completed (Front-end)
Home Page	U S _1	Coodinator	To import student projects from confluence	I can assess students' work	Must have	2	
	U S _2	Coodinator	To see those imported projects	I can view each project	Must have	2	
	U S _3	Coodinator	to delete a project that does not belong to my subject.	I can only keep the projects from my subject(s).	Could have	1	
Product Quality	U S _4	Coodinator	to see some metrics from a third-party code analysis tool	I can assess the product quality of each team based on the summary	Must have	3	
	U S _5	Coodinator	to view up to date statistical data	I can view the progress they made each week	Must have	4	
Process Quality	U S _6	Coodinator	to see the process quality summary of my project teams	I can assess the process quality of each team based on the summary	Must have	12 (break down)	
	U S _7	Coodinator	to view graphical data on a project team's Confluence activities	I can measure and assess a project team's process quality in terms of task management	Must have	4	
	U S _8	Coodinator	to view graphical visualisations on a project team's JIRA task activities (i.e., total vs. completed vs. remaining tasks)	I can measure and assess a project team's process quality in terms of task management	Must have	4	
	U S _9	Coodinator	to view graphical visualizations on the number of code commits with proper code reviews and Continuous Integrations tests in GitHub /Bitbucket/GitLab	I can measure and assess a project team's process quality in terms of codebase management	Must have	4	
Communication Quality	U S _10	Coodinator	to see the communication summary of my project teams	I can assess the communication of each team based on the summary	Must have	10 (break down)	
	U S _11	Coodinator	the system to filter out irrelevant messages (emoji only messages) when gathering assessable communication data for each project team	I can measure and assess a project team's communication quality in Slack accurately by excluding irrelevant messages.	Should have	2	
	U S _12	Coodinator	to view statistical data on the total number of messages sent over assessable Slack channels for each project team	I can measure and assess a project team's communication quality in terms of chat frequency in Slack	Should have	2	
	U S _13	Coodinator	to view each meeting information including meeting minutes for each project team	I can measure and assess a project team's meeting frequency and meeting quality	Must have	3	
	U S _14	Coodinator	to view statistical data on comments on GitHub	I can measure and assess a project team's communication quality in GitHub	Must have	3	
Individual Contribution	U S _15	Coodinator	to see the individual contribution summary of each team member in my project teams	I can assess the individual contribution of each team member based on the summary	Must have	13 (break down)	

	U S - 16	Coo rdin ator	to view statistical data on the number of pull requests involving peer reviews made per team member in GitHub/Bitbucket/GitLab	I can assess the individual contribution of each team member based on their pull requests	Mus t have	3	
	U S - 17	Coo rdin ator	to view statistical data on the number of lines of executable code made per team member	I can assess the individual contribution of each team member based on the number of lines of code they have written.	Can have	5	
	U S - 18	Coo rdin ator	to view statistical data on the number of completed tasks/tickets on JIRA by each team member	I can assess the individual contribution of each team member based on their task management /completion	Mus t have	3	
	U S - 19	Coo rdin ator	to view statistical data on the number of complete tasks from JIRA by each team member	I can assess the individual contribution of each team member based on their individual contribution	Mus t have	3	
	U S - 20	Coo rdin ator	to view statistical data on publishing and editing activity made by each team member on Confluence	I can assess the individual contribution of each team member based on their documentation work	Mus t have	8	
	U S - 24	Coo rdin ator	to view statistical data on meeting attendance of each team member	I can assess the individual contribution of each team member based on their attendance in meetings	Can have	5	
Configura tion	U S - 22	Coo rdin ator	to change the URLs of JIRA, Confluence and Git.	I can make any changes to the links of JIRA, Confluence and Git in case some project teams might want to change their JIRA/Confluence /Git destination.	Mus t have	3	

8.3.3 Sprint 2: Ceremonies

Stand-up Meeting 6th May 2021

Date:

2021-5-6

Time:

14:15 - 15:30

Attendees:

- Jirat Pasuksmit
- Zhang Sai
- Jingyu Li
- Boyang Sun
- Ruofan Zhang
- 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	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• Try to do the unified test before 13rd May 2021

Tasks completion in the Sprint 2 phase:

Who	What have you done	What you plan to do next
Zhang Sai	Continually developed Coordinator Home page's functions according to client's requirements and feedback	Try to finish all Coordinator home UI design and complete Readme and change log file
Ruofan Zhang	Led the team to complete 80% of the UI and logic of the front-end project	Implement the unfinished features and deploy the app so that it can be accessed by a domain
Jingyu Li	Add input git username and password function	Test the configuration page with the backend team member
Sarah Sultan		
Yuhang Xie	<ul style="list-style-type: none">▪ Get total number of team members from backend Confluence API response.▪ Set the threshold of the member list.	<ul style="list-style-type: none">▪ Display warning of Confluence visibility if the list exceeds the threshold.

Zixin Ye		
Fengrui Zhang	Implemented the login page	Test the login page and product quality page with the back-end, make a new version of code matrix visualization.
Boyang Sun	Get imported project bugs	Unified test for other features
Zisheng Cheng	Api exception handling Modify the data return format	test and presentation
Jingdan Cui	Insert meeting minutes function	Review the assigned functions done by myself and ensure codes are correct
Haoyu Qin		
Jinzhe Shan		
Pin Wang		
Fu Xie	Store Github and Jira URLs in the same table. Bug fix for Jira APIs. Auto update records for everyday	Fix bugs for Jira APIs.
Chongjing Zhang	Git authentication Daily update Git commit function	prepare final presentation

Stand-up Meeting 20th May 2021

Date:

2021-5-20

Time:

14:15 - 15:30

Attendees:

- Jirat Pasuksmit
- All team members

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	<ul style="list-style-type: none">• 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 (Presentation)
Give feedback and Suggestions	Jirat Pasuksmit	<ul style="list-style-type: none">■ Team list warning for exceeding the limit = need to present it in presentation■ Product quality page = check if the data display■ Process quality page= confluence: need to notify the user of the beginning time and end time e.g. (3 Month)■ Logout = need a notification the logout is successful■ Presentation:<ul style="list-style-type: none">■ show what we have done■ details backend design■ prepare some problems■ what we have changed from the previous team, what is the original project, explain what can be improved?■ 20 minutes is fine for the presentation■ system demo and presentation = play it together it will be great!■ Don't forget to present what we've done during the inception, sprint1, and sprint2. Please also include the Agile processes we adopt as well.

Tasks completion in the Sprint 2 phase:

Who	What have you done	What you plan to do next
Zhang Sai	Completed Coordinator Home UI design and Readme file	Test functions for preparing project demonstration
Ruofan Zhang	<ul style="list-style-type: none">• Integrated all modules of the front-end project• Deployed the app	Test the app and fix bugs
Jingyu Li	<ul style="list-style-type: none">• Finish testing with backend team• Finish configuration page	<ul style="list-style-type: none">• Work on some relevant confluence documents

Sarah Sultan	<ul style="list-style-type: none"> • Readme file in Github • Update confluence document 	<ul style="list-style-type: none"> • Update confluence document
Yuhang Xie	<ul style="list-style-type: none"> ▪ Set warning on the top of the project to reminds the visibility of the Confluence page. 	<ul style="list-style-type: none"> ▪ Display a part of the members list even the response of the member number exceeds the threshold.
Zixin Ye		
Fengrui Zhang	Made a new version of code matrix visualization, added notice bar and loading symbol on the login page and tested the login page.	Test product quality page with the back-end.
Boyang Sun		
Zisheng Cheng	<p>prepare presentation</p> <p>Finish testing with front-end team</p>	Write related git related documents
Jingdan Cui	Prepare for the final presentation	Write related Confluence documents
Haoyu Qin		
Jinzhe Shan		
Pin Wang		
Fu Xie	<ul style="list-style-type: none"> • Work with Haoyu Qin to implement jira-agile-metrics to retrieve data • Bug fix for Jira APIs 	<ul style="list-style-type: none"> • Update confluence document
Chongjing Zhang	Final presentation	Sprint2 document

[Like](#)

8.3.4 Sprint 2: Documentation

Title		Creator	Modified
1 Project Requirement		Boyang Sun	Zixin Ye
	1.1 Project Overview		Jingyu Li
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	
6 API documents		Boyang Sun	Pin Wang
	6.3 Sprint 2: API	Pin Wang	
	6.3.1 Unified Test	Boyang Sun	
7 Deployment		Boyang Sun	Jinzhe Shan Pin Wang
	7.1 Cloud Server Deployment	Pin Wang	Boyang Sun Haoyu Qin
	7.2 Scitools Understand Deployment	Jinzhe Shan	
8 Sprint Summaries		Boyang Sun	Sarah Sultan A AL YAHYA
8.3 Sprint 2		Sarah Sultan A AL YAHYA	
	8.3.5 Sprint 2: Individual Contribution	Sarah Sultan A AL YAHYA	
8.3.1 Review of Sprint 2 - Back-end Team		Sarah Sultan A AL YAHYA	
	Coding Summary - Back-end team S2	Boyang Sun	
	Deployment Report - Back-end team S2	Boyang Sun	
	Design Update - Back-end team S2	Boyang Sun	
	Handover Report - Back-end team S2	Boyang Sun	
	Requirement Update - Back-end team S2	Boyang Sun	
	Tasks achieved - Back-end team S2	Boyang Sun	
	Testing Report - Back-end team S2	Boyang Sun	
8.3.2 Review of Sprint 2 - Front-end team		Sarah Sultan A AL YAHYA	
	Coding summary - Front-end team S2	Ruofan Zhang	
	Deployment report - Front-end team S2	Ruofan Zhang	
	Design update - Front-end team S2	Ruofan Zhang	Sarah Sultan A AL YAHYA Sam Ross
	Handover report - Front-end team S2	Ruofan Zhang	
	Tasks achieved - Front-end team S2	Ruofan Zhang	
	8.3.3 Sprint 2: Ceremonies	Sarah Sultan A AL YAHYA	
	Stand-up Meeting 13th May 2021	Sarah Sultan A AL YAHYA	
	8.3.4 Sprint 2: Documentation	Sarah Sultan A AL YAHYA	
9 Testing		Boyang Sun	Jinzhe Shan
	9.4 Testing Progress	Ruofan Zhang	Ruofan Zhang ZISHENG CHENG Fu Xie Jingyu Li Boyang Sun Pin Wang

	9.5 Super User Account	Boyang Sun	
10 Meeting		Sarah Sultan A AL YAHYA	
10.1 Meeting Notes for Project Team		Boyang Sun	
10.2 Meeting with Clients		Boyang Sun	
10.3 Meeting Note for Front-end Team		Sarah Sultan A AL YAHYA	
10.4 Meeting Note for Back-end Team		Sarah Sultan A AL YAHYA	
11 Team and role		Boyang Sun	
12 Risk Management		Boyang Sun	Jingyu LI ZISHENG CHENG Jingdan Cui

8.3.5 Sprint 2: Individual Contribution

Back-end Team

S2 Contribution - Boyang Sun

S2 Contribution - Chongjing Zhang

S2 Contribution - Pin Wang

S2 Contribution - Jinzhe Shan

S2 Contribution - Fu Xie

S2 Contribution - Zisheng Cheng

S2 Contribution - Haoyu Qin

S2 Contribution - Jingdan Cui

Front-end Team

S2 Contribution - Ruofan Zhang

S2 Contribution - Zixin Ye

S2 Contribution - Fengrui Zhang

S2 Contribution - Yuhang Xie

S2 Contribution - Sarah Sultan Alyahya

S2 Contribution - Sai Zhang

S2 Contribution - Jingyu Li

S2 Contribution - Boyang Sun

- Client VM Deployment
- Clearing obstacles for team members
- Design and control the final presentation
- Unified test of Sprint 2 features
- Workload allocation

S2 Contribution - Chongjing Zhang

- Optimize git_commit_counts function
- Rewrite the logic of git commit in gitutil
- Add git authentication function to gitutil
- Implement function of daily update
- Modify some previous logic of git_commit
- Fix previous bugs
- Write some document about my tasks in Sprint1

S2 Contribution - Fengrui Zhang

- **Update for ProductQualityPage**
 - redesigned code matrix form to make it more readable
 - change the API to be consistent with the back-end
- **Implement Login Page**
 - Implemented the login page
 - Added more visual aids on the login page
- **Test fetching data from back-end**
 - show data from the back-end successfully
 - login successfully
- **Update Design Concept document**
 - Update all the pictures

S2 Contribution - Fu Xie

- Store Jira and Github URLs in the same table
- Make the Jira APIs auto update when users setup Jira URLs
- Auto update records everyday
- Fix bugs for the previous way of fetching data for Jira API
- Work with my teammate to implement jira-agile-metrics to fetch data for Jira API
- Testing, integrating the new implemented method
- Fix bugs related to database
- Presentation and documents

S2 Contribution - Haoyu Qin

- Mitigation for SSL issues in unimelb Atlassian
- Tweaks in Jira fetching API
- Revised fetched data schemes in accordance with new requirements by Django and the front-end
- Optimization: Switched to jira-agile-metrics as ticket history fetching backbone
- Tests: Unified tests on deployment server
- Bug fixes in Django
- Changes to SSL certificate installation script for server compatibility
- Documentation for SSL mitigation and other updates

S2 Contribution - Jingdan Cui

1. Continue to update and modify the 2 APIs assigned to me in Sprint 1
 - a. Get the Already Imported Project: now the function will get the coordinator id from the session.
 - b. Get the Meeting Minutes: the meeting minutes table will be updated every 24 hours. And all the meeting minutes related will be imported when the project is imported.
2. Use Postman to test the APIs above
3. Modify the meeting minutes table
4. Write the functions-related documentations.

S2 Contribution - Jingyu Li

Modify the Configuration Page

1. Update the UI (add input Git username and password) for Project Configuration Page.
2. Update API for Project Configuration Page.
3. Test API for Project Configuration Page when backend server is deployed.

Documentation

1. Complete [Tasks achieved - Front-end team S2](#).

S2 Contribution - Jinzhe Shan

There is the contribution of Jinzhe Shan in Sprint 2

1. Researched SciTools Understand software
2. Communicate and get an educational license from the SciTools Support Team
3. Implement Git Metrics Module from 0 to 1
4. Deploy the environment of VM and EC2 with Boyang Sun and Pin Wang
5. Fix some bugs of the Git module such as Git Commit and Git Utils
6. Problem solver of the conflict between Understand and Django
When I get the code metrics using python and Understand, It works on a single python but meets a QT thread exception when it runs in our Django Server which is really hard to solve because it is the conflict between Django Structure and Understand Tool
7. Write the Deployment Document about SciTools Understand: [7.2 Scitools Understand Deployment](#)
8. Joint testing with Front-end Ruifeng Zhang

S2 Contribution - Pin Wang

S2 Contribution - Ruofan Zhang

1. deploy the app on the server using nginx
2. register a domain for the app
3. re-config the routing
4. implement logout
5. implement preview team member list
6. implement import projects
7. implement delete imported projects
8. implement the redirect logic
9. implement the 404 page for invalid url
10. review pull requests
11. assign tasks to teammates
12. help teammates solve problems
13. integrate all modules in the front-end project

S2 Contribution -Sai Zhang

Continually developed Coordinator Home page according to Client's requirements and feedback from supervisor.

- Implemented the View button for previewing student name and student email information of all students who took the current project.
- Implemented the Alert Dialog for responding the Delete button click action. It aims to notice the user if he/she wants to delete the specific project.
- Implemented the scrollable search option list for showing search results, for avoiding showing infinity long list on the webpage.
- Updated the table UI for better design purpose.
- Learned to deploy API and connect with remote server that team deployed

Test each function of whole system for helping team to generate a good product.

- Did unit testing for each functionality to make sure each feature work smoothly.
- Reported problems I found to the team

Non-code related work:

- Participated in completing ReadMe file in Github Repository of Frontend Team
- Completed Change Log file in Github repository of Frontend Team

Arranged regular client meeting with whole team for reporting work progress and getting feedback from client.

Send Final Presentation Invitation to every examiner.

Prepared Product Demonstration for final presentation.

S2 Contribution - Sarah Sultan Alyahya

- **Preparing work:**

- Learned how to modify the LineChart.js, so that it marks one line or more lines with different color

- **Developing:**

- Different colours for line chart legends
- Visualization of the product quality page
- Editing the README in the repo.

- **Presentation:**

- Prepare slides for the presentation (Front-end previous/current project)

- **Confluence page:**

- Worked on [8 Sprint Summaries](#)
- Create [8.3.4 Sprint 2: Documentation](#) and update daily

S2 Contribution - Yuhang Xie

Modify the project overview page

1. Add the total number of member to get the response from Confluence API.
2. Set member list threshold to 30
3. Show 30 members in the table if the list exceeds 30
4. Show the warning of Confluence visibility if the list exceeds 30

Documentation

1. Update design concept.

S2 Contribution - Zisheng Cheng

- Return different responses according to the abnormal execution of the api function
- Cooperate with front-end individual contribution teammates to make small changes to the api.
- Fix previous bugs
- Write some document about my tasks in Sprint1

S2 Contribution - Zixin Ye

1. Fix some problems on UI of individual contribution page.
2. Implement some API functions for individual contribution page.
3. Test the API functions on Communication Page and Individual Contribution Page when back-end server is deployed.

Handover Report

Strengths of our Project

Good UI design (Screenshot and description)

- Coordinator Home:

Search projects by entering key words

Project Name	Confluence Link	Operation	
ActiveBoard	https://confluence.cis.unimelb.edu.au:8443/display/SWEN900162019S1ATeam/Home	DELETE	VIEW
COMP90082-2021-SM1-SP	https://confluence.cis.unimelb.edu.au:8443/display/COMP900822021SM1SP/Home	DELETE	VIEW
Aphasia Mail Client Green	https://confluence.cis.unimelb.edu.au:8443/display/AphasiaGreen/Home	DELETE	VIEW
Queue	https://confluence.cis.unimelb.edu.au:8443/display/QUEUE/Home	DELETE	VIEW

- COMP90082-2021-SM1-SP

Student Name	Student Email
Pin Wang	pinwang@student.unimelb.edu.au
Zixin Ye	yezy2@student.unimelb.edu.au
Jingdan Cui	jingdanc1@student.unimelb.edu.au
Jirat Pasuksmit	pasuksmitj@student.unimelb.edu.au
Ruofan Zhang	ruofzhang@student.unimelb.edu.au
Sarah Sultan A.AL YAHYA	salyahya@student.unimelb.edu.au
Boyang Sun	boyangs@student.unimelb.edu.au
Chongjing ZHANG	chongjingz@student.unimelb.edu.au
Fengrui Zhang	fengrzhang@student.unimelb.edu.au

- Project Overview
- Product Quality
- Process Quality
- Communication
- Individual Contribution
- Configuration

Good Back-end design (Code logic, performance data, and architecture diagram if needed)

- Confluence
 - Store and retrieve projects from the database instead of fetching data from Confluence
 - the page loading time from previously >10s to currently <1s
 - Update three tables, which are user list, individual confluence contribution and meeting minutes, every 24 hours
- Git
 - Add git authentication to restrict invalid access and be able to visit a private repository.
 - Use database cache to speed up visit time from more than 30s to less than 1s.
 - Git commits data in git commit database updated logic are designed in different situations: 1. No space_key in db, crawl, 2. Has space_key in db, execute logic 3. If updated data with the space_key exists, do nothing, else update.
 - Data with timestamps are sorted by the last second of the day.
- Jira
 - Switched to jira-agile-metrics for ticket history fetching. Time required to do cold fetches is down from ~70min to ~5s .
 - Caching query results with Django. Long query times (~10s) only occur on cold fetches (initial caching), hot fetches are within 1s.
 - Daily syncs with Jira for better data availability

Implemented Optional Features of the client (Show how well we finish the optional feature of our requirement)

Code Metrics Optional Function Jinzhe Shan

Code Metrics is an optional function in the client's requirement. It is no any implement in the previous team because it is really a difficult module. Base on the suggestion of Jirat and our client, I researched SciTools Understand software, which is a kind of professional code analysis tool and it needs to purchase a license.

I applied for the educational license for the whole SP Group with the help of Jirat and Eduardo and deploy the Understand into the Linux server.

Next, I got software source code using git methods.

Then, I used Understand Python API & Und Command Lines to generate indexes for describing the product quality of a software project and store them in JSON files.

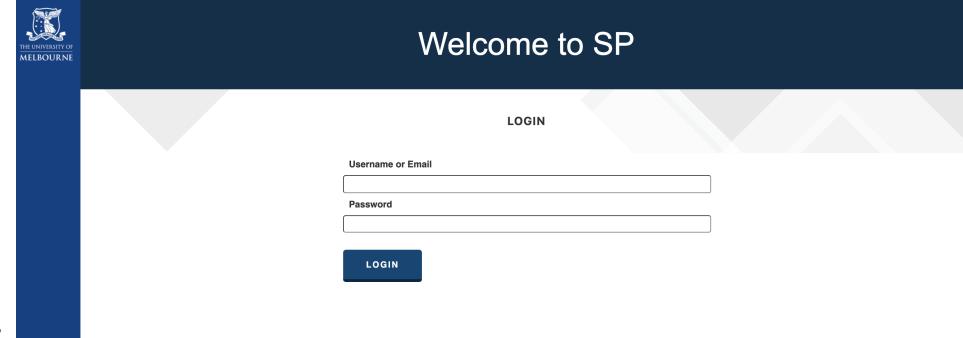
Finally, I read the corresponding JSON file in Django and package them to Git Metrics API.

The Whole WorkFlow of Get Code Metrics is shown as below:

What we have done(you can use our final presentation ppt as core materials)

Front-end

- **Login page:** Allows university staff to login via using their university account



- **Coordinator Home:** Shows the list of team members and their corresponding projects. A coordinator can Viewing Project, Importing Project, Deleting Imported Project and Viewing Specific Imported Project.



Coordinator Home

PROJECT MANAGEMENT

Search projects by entering key words

Project Name	Confluence Link
ActiveBoard	https://confluence.cis.unimelb.edu.au:8443/display/SWEN900162019
COMP90082-2021-SM1-SP	https://confluence.cis.unimelb.edu.au:8443/display/COMP900822021

Student Name Student Email

Pin Wang	pinwang@student.unimelb.edu.au
Zixin Ye	yezy2@student.unimelb.edu.au
Jingdan Cui	jingdanc1@student.unimelb.edu.au
Jirat Pasuksmit	pasuksmitj@student.unimelb.edu.au
Ruofan Zhang	ruofzhang@student.unimelb.edu.au
Sarah Sultan A AL YAHYA	salyahya@student.unimelb.edu.au
Boyang Sun	boyangs@student.unimelb.edu.au
Chongjing ZHANG	chongjingz@student.unimelb.edu.au
Fengrui Zhang	fengrzhang@student.unimelb.edu.au
Haoyu Qin	haoyug@student.unimelb.edu.au
Jingyu Li	jingyul5@student.unimelb.edu.au



Coordinator Home

PROJECT MANAGEMENT

Search projects by entering key words

Project Name	Confluence Link	Operation
ActiveBoard	https://confluence.cis.unimelb.edu.au:8443/display/SWEN900162019S1ATeam/Home	<button>DELETE</button> <button>VIEW</button>
COMP90082-2021-SM1-SP	https://confluence.cis.unimelb.edu.au:8443/display/COMP900822021SM1SP/Home	<button>DELETE</button> <button>VIEW</button>

- Project Overview:** This page aims to allow the coordinator to view students' name, profile, student ID and student email address.



Project Overview

COMP90082-2021-SM1-SP

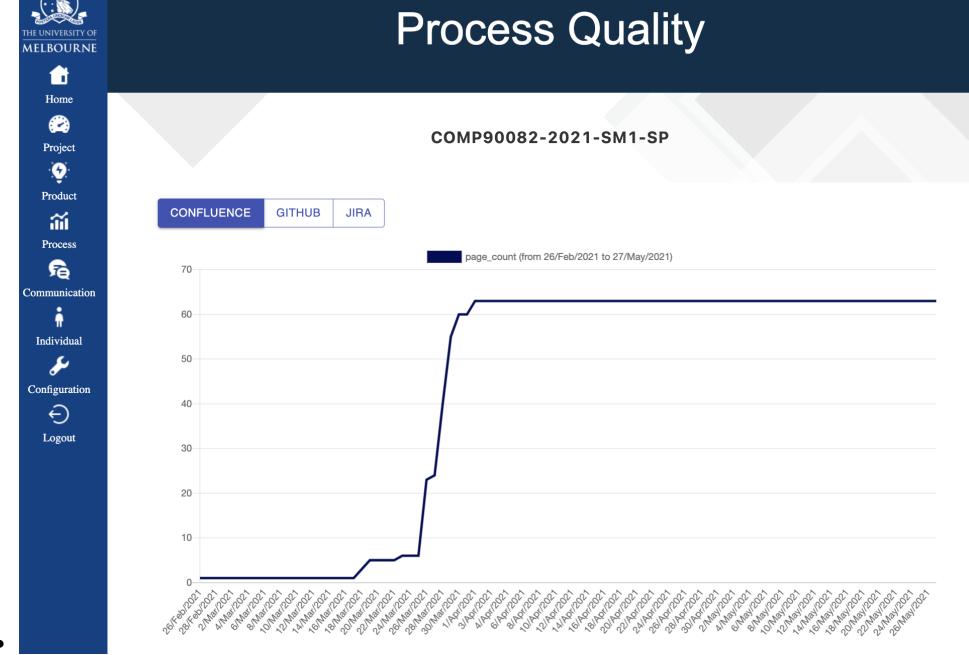
Name	Profile	Student ID	Email Address
Pin Wang		pinwang	pinwang@student.unimelb.edu.au
Zixin Ye		yezy2	yezy2@student.unimelb.edu.au
Jingdan Cui		jingdanc1	jingdanc1@student.unimelb.edu.au
Jirat Pasuksmit		pasuksmitj	pasuksmitj@student.unimelb.edu.au
Ruofan Zhang		ruofzhang	ruofzhang@student.unimelb.edu.au
Sarah Sultan A AL YAHYA		salyahya	salyahya@student.unimelb.edu.au
Boyang Sun		boyangs	boyangs@student.unimelb.edu.au
Chongjing ZHANG		chongjingz	chongjingz@student.unimelb.edu.au
Fengrui Zhang		fengrzhang	fengrzhang@student.unimelb.edu.au

- Product Quality:** In product quality page includes statistical and/or graphical summaries by code analysed tool (TBC) on: Code quality, Test code quality. For Software Engineering tools include Git (Team's choice of GitHub, GitLab or Bitbucket).

The screenshot shows a dark-themed dashboard titled "Product Quality". On the left is a vertical sidebar with icons for Home, Project, Product, Process, Communication, Individual, Configuration, and Logout. The main content area has a title "COMP90082-2021-SM1-SP". Below it is a table with the following data:

Metric	Number
Lines	846
Classes	12
Files	12
Functions	53
Comment Lines	253
Comment Lines / Code Lines	0.3

- **Process Quality:** In process quality page includes statistical and/or graphical summaries on sprint velocity and burndown, documentation on Confluence, code review frequency and coverage and code commits. For software engineering tools include GitHub, JIRA and Confluence.



- **Communication:** In communication quality page includes statistical and/or graphical summaries on comments from engineering tools which are GitHub, Confluence.

The screenshot shows a table titled "Communication" with the identifier "COMP90082-2021-SM1-SP". The table has two columns: "Meeting Name" and "Meeting Minutes".

Meeting Name	Meeting Minutes
Meeting with client #1	https://confluence.cis.unimelb.edu.au/8443/pages/viewpage.action?pageId=74781618
Meeting Notes for 23rd March	https://confluence.cis.unimelb.edu.au/8443/display/COMP900822021SM1SP/Meeting+Notes+for+23rd+March
10 Meetings	https://confluence.cis.unimelb.edu.au/8443/display/COMP900822021SM1SP/10+Meetings
Meeting Notes for 28th, March	https://confluence.cis.unimelb.edu.au/8443/display/COMP900822021SM1SP/Meeting+Notes+for+28th+March
Front-end Meeting Notes on 28th March -	https://confluence.cis.unimelb.edu.au/8443/pages/viewpage.action?pageId=74787344
Stand-up Meeting 18 March 2021	https://confluence.cis.unimelb.edu.au/8443/display/COMP900822021SM1SP/Stand-up+Meeting+18+March+2021
Stand-up Meeting 25 March 2021	https://confluence.cis.unimelb.edu.au/8443/display/COMP900822021SM1SP/Stand-up+Meeting+25+March+2021
Meeting Notes for 30th, March	https://confluence.cis.unimelb.edu.au/8443/display/COMP900822021SM1SP/Meeting+Notes+for+30th+March
Meeting with Client #2	https://confluence.cis.unimelb.edu.au/8443/pages/viewpage.action?pageId=74788829

- **Individual Contribution:** In individuals' contribution page includes statistical and/or graphical summaries of every team member's activities which are: task completion on JIRA, Code reviews on Git and Documentation contribution on Confluence. For software engineering tools include Git (Team's choice of GitHub, GitLab or BitBucket), JIRA, and Confluence.

The screenshot shows a donut chart titled "Individual Contribution" with the identifier "COMP90082-2021-SM1-SP". The chart is divided into several segments of different colors, representing different team members. A legend on the right side lists the names of the team members corresponding to each color.

Team Member	Contribution Segment
Pin Wang	Large Green Segment
Zhen Ye	Small Purple Segment
Jingtao Cui	Small Red Segment
Jirat Puswamit	Small Yellow Segment
Ruotan Zhang	Small Blue Segment
Sarah Sultan A AL YAHYA	Small Green Segment
Boying Sun	Small Yellow Segment
Yingting ZHANG	Small Blue Segment
Fengyu Zhang	Small Red Segment
Heviq On	Small Purple Segment
Jingyu Li	Small Yellow Segment
Jinhe Shan	Small Green Segment
Sal Zhang	Small Blue Segment
Fu Xie	Small Purple Segment
...	...

- **Configuration:** Here the URLs for each of the team's tools should be set (currently not implemented). The account names/ emails for each team member's Slack and github accounts must be set in order to access their individual details in the 'Individual Contribution' section.

The screenshot shows a configuration page titled "Project Configuration" with the identifier "COMP90082-2021-SM1-SP". It includes fields for "Jira Url:", "Git Url:", "Git Username:", and "Git Password:". There is also a "SUBMIT" button at the bottom.

Informational Notes:
You may need to wait for a few minutes before you can view the data after finishing the configuration.

Jira Url:

Git Url:

Git Username:

Git Password:

SUBMIT

Back-end(Show related user stories here)

Confluence

Function	Previous	Current
User List	Use Confluence Group	Use Confluence Space; Daily update
Search Spaces by Keyword	Not Implemented	Search from our site
Total Number of Pages on Each Day	Not Implemented	Shows the progress Daily update
Import Project	Not implemented	Store the Confluence space key and the coordinator id into the database
Delete Project	Not implemented	Delete the project and clear the related data from the database
Get Project	Fetch project by calling Confluence API	Retrieve project from the database
Get Meeting Minutes	Not implemented	Show the titles and links of relevant meeting minutes

Github

- Get Git Individual Contribution
- Get Git Commit Count
- Get Git Code Metrics

Jira

Function	Previous	Current
View Individual Contribution	Similar function exists that returns queries in real time	Caches and returns relevant individual ticket count
View Project Process History	Not Implemented	Caches and returns relevant ticket history
Setup Github and Jira URL	Not Implemented	Stores URLs in database

Soft Skills Learned in the Whole Process(Add more here)

- Communication Skills
- Web service deployment skill
- Valuable experience about Agile process