1. RS team 2 2
1.1 Background 3
1.2 Team roles 5
1.3 Communication Plan 6
1.4 Project requirements 8
1.4.1 Motivational Model 9
1.4.2 Personas 11
1.4.3 User Stories 15
1.4.4 Acceptance criteria 18
1.4.5 Interview questions with swen90013 group users 21
1.5 Design 22
1.5.1 Prototype 23
1.6 Architecture 33
1.6.1 Architecture diagram 34
1.6.2 Component Diagram 36
1.7 Quality 40
1.7.1 Code Review Policy 41
1.7.2 Git Policy 42

RS team 2

Project name

Software Resources for SWEN90013

Project Description

This project aims to provide a social learning platform as a learning community. Compared with traditional discussion boards, we will focus on the review and provision of resources. At the same time, this platform has the ability to measure popular trends, and active participants will have hope to get some permission rewards.

Project repository

https://github.com/Taylbr/COMP90082-2021-RS-Group-2

Client

Name	E-mail
Professor Leon Sterling	leonss@unimelb.edu.au
Dr. Eduardo Oliveira	eduardo.oliveira@unimelb.edu.au

Supervisor

Name	E-mail
Colton Carner	colton.carner@unimelb.edu.au

Project Team

Student ID	Full Name	Preferred Name	E-mail	Role
1155892	Lixia Schwer	Lisa	lschwer@student.unimelb.edu.au	Scrum Master
900437	Runfeng Du	Aaron	runfengd@student.unimelb.edu.au	Tech Lead
1037690	Xu Han	Zoe	xhha3@student.unimelb.edu.au	Product Owner

Background

Project name

Software Resources for SWEN90013

Project Description

The project is a repository of learning resources for Melbourne University Engineering students to self-learn about important emerging technologies such as programming frameworks, testing tools, and deployment tools. The product aims to provide a convenient and easy-to-use social learning platform as a learning community by sharing, discussing, and ranking the resources provided either by teaching staff or a student related to CIS subjects. Students need to login to get access to the resources and need permission to share a new resource. However, active participants will get rewards and possibly escalate to a moderator. The platform encourages interaction and communication among students by actively contributing to the platform.

Project repository

https://github.com/ccarner/COMP90082-RS-2021

Jira

https://comp90082-rs-swen90013.atlassian.net/secure/RapidBoard.jspa?projectKey=RS&rapidView=1

Client

Name	E-mail
Professor Leon Sterling	leonss@unimelb.edu.au
Dr. Eduardo Oliveira	eduardo.oliveira@unimelb.edu.au

Supervisor

Name	E-mail
Colton Carner	colton.carner@unimelb.edu.au

Project Team

Student ID	Photo	Full Name	Preferred Name	E-mail	Role
1155892		Lixia Schwer	Lisa	lschwer@student.unimelb.edu.au	Scrum Master & Quality Lead
900437		Runfeng Du	Aaron	runfengd@student.unimelb.edu.au	Tech Lead

1037690		Xu Han	Zoe	xhha3@student.unimelb.edu.au	Product Owner

Team roles

Scrum Master & Quality Lead

Lixia Schwer

Major: Distributed computing in MC-IT. Preferred programming languages: Python and Java. Programming experience: Android development and a bit of experience of full-stack. Interested area: Security.

Product Owner

Han Xu

Doing MC-IT, AI in Unimelb. Graduated from Deakin University as Bachelor of Information Technology with Honors. Good knowledge of programming, and the ability to learn new skills and adapt to challenges. Unity project development experience. Also carry out website development. Blockchain and biological inspired scheme research background and hope to do further research about computer science on AI.

Tech Lead

Aaron Du

MC-IT Distributed computing, interested in system architecture designs and distributed system designs.

Fluent in Java, Python, Go.

Communication Plan

The communication plan is used to assist in communicating effectively and efficiently for describing the tasks and responsibilities of each team member, the frequency and methods used to communicate within the team and between team and client in order to deliver the project as expected.

Description	Objective	Frequency	Method	Nature	Audience	Owner	Records
Daily communication	For exchanging any ideas or asking questions about the project within the team	Daily	Slack	Informal.	Project team	Project team	Yes. Have message history on slack.
Stand-up meeting	Discuss what have done yesterday, what will do today and any problem encountered	2 times/week	Zoom	Informal.	Project team	Scrum Master	Yes. Have meeting minutes on Confluence: "Meetings" "Stand-up meetings"
Group meeting	Discuss project progress and any problem encountered	Once each week	Zoom	Formal.	Project team	Scrum Master	Yes. Have meeting minutes on Confluence: "Meetings" "Group meetings"
Sprint planning	Plan the sprint goal and clarify the sprint backlog, prioritize the work by the product owner, create release plan	Every 4 weeks	Zoom	Formal.	Project team	Scrum Master	Yes. Have meeting minutes on Confluence: "Meetings" "Group meetings"
Sprint retrospective	Summarise the sprint about what was working, what wasn't working, and what to improve.	Every 4 weeks	Zoom	Formal.	Project team	Scrum Master	Yes. Have meeting minutes on Confluence: "Meetings" "Group meetings"
Sprint review	Present the progress during the sprint and show a demo of new features and functions to the client and get feedback	Every 4 weeks	Zoom	Formal.	Project team	Scrum Master	Yes. Have meeting minutes on Confluence: "Meetings" "Group meetings"
Client meeting	Clarify project requirements and any unclear questions about the project.	When necessary	Email /Zoom	Formal.	Project team	Scrum Master	Yes. Have meeting minutes on Confluence: "Meetings" "Client meetings"
Emergency co mmunication	Any emergency communication needed.	When necessary	Slack, Ema il	Informal /Formal	Project team	Project team	No if it's individual communication. Yes if it's a group meeting, and meeting minutes can be found on Confluence: "Meetings" "Group meetings"

Meeting notes

For taking meeting notes, we will take turns. Here is the order of taking notes:

Order of taking notes
1. Lixia Schwer
2. Xu Han
3. Runfeng Du

Team member contact

Name	Slack	Email
Lixia Schwer	Lixia Schwer	lschwer@student.unimelb.edu.au
Runfeng Du	Aaron Du	runfengd@student.unimelb.edu.au
Xu Han	Xu Han	xhha3@student.unimelb.edu.au

Communication methods

Methods	Purpose	Owner	Possible Participants
Email	Use Email mainly for external communication and receive Zoom invitation • Contact with the client and potential users by either Product Owner or Scrum Master • Schedule a meeting by either Product Owner or Scrum Master • The sender will need to add possible participants to the email	Scrum Master /Product owner	• Clients • Potential users • Project team • Supervisor
Slack	Use Slack mainly for communication within the team, across teams, and with the supervisor • Discuss problems • Exchange ideas • Provide resources • General questions and reminders • Meeting invites and agenda • Group-related general topic post in the "group-2-swen90013" channel • Group-related resources topic post in the "group-2-resources" channel • Group-related development topic post in the "group-2-development" channel • Project-related topic post in "general" channel	Project team	• Project team • Supervisor
Zoom	Use Zoom mainly for meetings with client and group meetings • Scrum master will send the Zoom invitation via Email before the meeting • Product owner will need to record the meeting when it's a client meeting • Meeting notes will be taken by team members in turn and please refer to the Meeting Notes section above for who should be taking the notes for each meeting. The person who is taking the notes will also need to update the meeting minutes on Confluence and create the action items. • The recorded meeting link can be found in the meeting minutes.	Scrum Master /Product owner	• Clients • Potential users • Project team • Supervisor

Confluence	Use Confluence to document all the project work, resources, and progress
	• Every team members' responsibility to keep Confluence updated and maintained to reflect the current project progress
	  • Scrum Master needs to make sure the project progress is as expected and no blockers for the team and project
	  • Product Owner needs to review the project backlog, sprint backlog, and tasks for the current sprint to make sure the high priority tasks are implemented
	  • Quality Lead needs to review the Confluence documents and code quality as expected
	  • Tech Lead needs to make sure any tech issues are resolved on time, the website runs as expected, and the project deployment problems
	• Work should be documented on Confluence listed below, but not limited to:
	  • Project requirements, scope, and description
	  • Roles and responsibilities of each team member
	  • Project-related design, diagrams, models, and architecture
	  • Meeting minutes
	  • Project-related quality control and assurance including both documentation and coding eg: code testing cases, acceptance criteria, etc
	  • Any project-related resources
	  • Github link
	  • Jira progress
		Project team	• Project team
			• Supervisor

Jira	Use Jira for managing tasks and progress
	• Every team members' responsibility to keep the Jira board updated for
	  • the tasks have been assigned
	  • the tasks have been working on
	  • the tasks have been testing
	  • the tasks have been done
	• Product Owner needs to make sure tasks have been ranked priority and high priority tasks have been added to the sprint backlog
	• Product Owner needs to make sure the effort estimation of each task is completed by the team
	• Scrum Master should be aware of any blockers the team is facing at any stage
		Project team	• Project team
			• Supervisor

Github	Use Github for version control and work on the project simultaneously. The branches of the project git repository much at least have a Master branch and Dev branch before start working on the code. • Everyone should always create a new branch for the new task • Everyone should use the standard name conversion when creating a new branch and coding variables • Everyone should give appropriate commit messages which detail what have been changed in the commit • Everyone should follow standard commenting principles • Before each merges to the Dev branch or Master branch, the team member should create a code review. The code then can be merged after reviewed by the team and receives approval from the team. • Tech Lead should create the release tag at the end of each sprint	Project team	• Project team • Supervisor

Project requirements

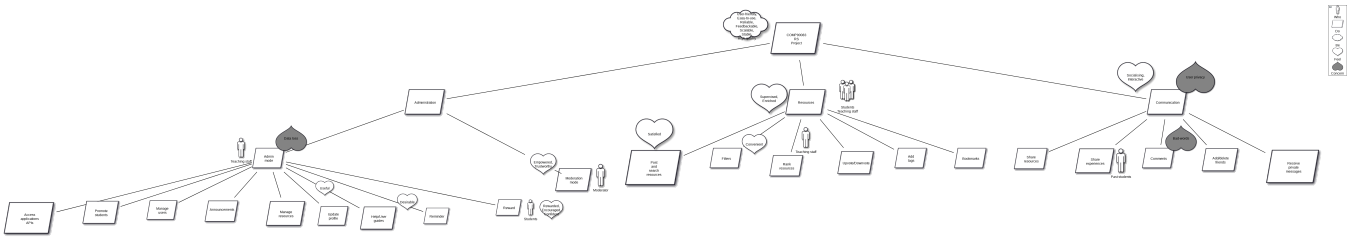
In this section, the Motivational model, Personas, User Stories, Acceptance Criteria, and interview questions with the swen90013 group users can be found.

- Motivational Model
- Personas
- User Stories
- Acceptance criteria
- Interview questions with swen90013 group users

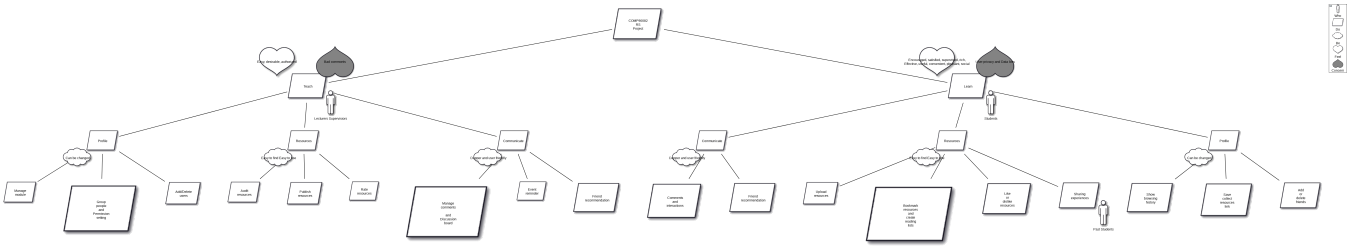
Motivational Model

Page Description:	The Motivational Model shows the high level of this project. There are 2 versions included on this page.
Page Status:	COMPLETE

Version 2



Version 1



Do/Be/Feel table (V1&V2)

Who	Do	Be	Feel	Concern
Teaching staff & moderator	Moderation function/mode eg for teaching staff or student	Easy-to-use	encouraged	bad words
Current students	Share resources	High-quality	Convenient	user privacy
Past students	Add resources from teaching staff	stable	satisfied	loss data
	Like a resources	Feedbackable	rewarded	
	Comment on resources	reliable	supervised	
	View resources	scalable	empowered	
	Rank resources	User friendly	enriched	
	Filters eg readings, materials for a specific subject		useful	
	Rewards for active participation eg points or ranking		socialising	
	Add tags		desirable	
	Promotions for active participation eg escalate to Moderator		confident	
	Bookmark		Interactive	
	Discussion board		Trustworthy	
	Submit resources from students			
	Show comments or discussion from previous semesters			
	Search eg resources, answers, etc			
	Categories eg programming frameworks, testing tools, and deployment tools			
	Update resources			
	Login			

	Update profile eg password			
	Design improvement			
	Reminder			
	Vote the best answer			
	Sample codes eg for API			
	Admin mode for other operations eg CRUD resources or students			
	reflection - ideas from previous students			
	Help/User guide			
	Add/Delete friends			
	Send/Receive private messages			

Personas

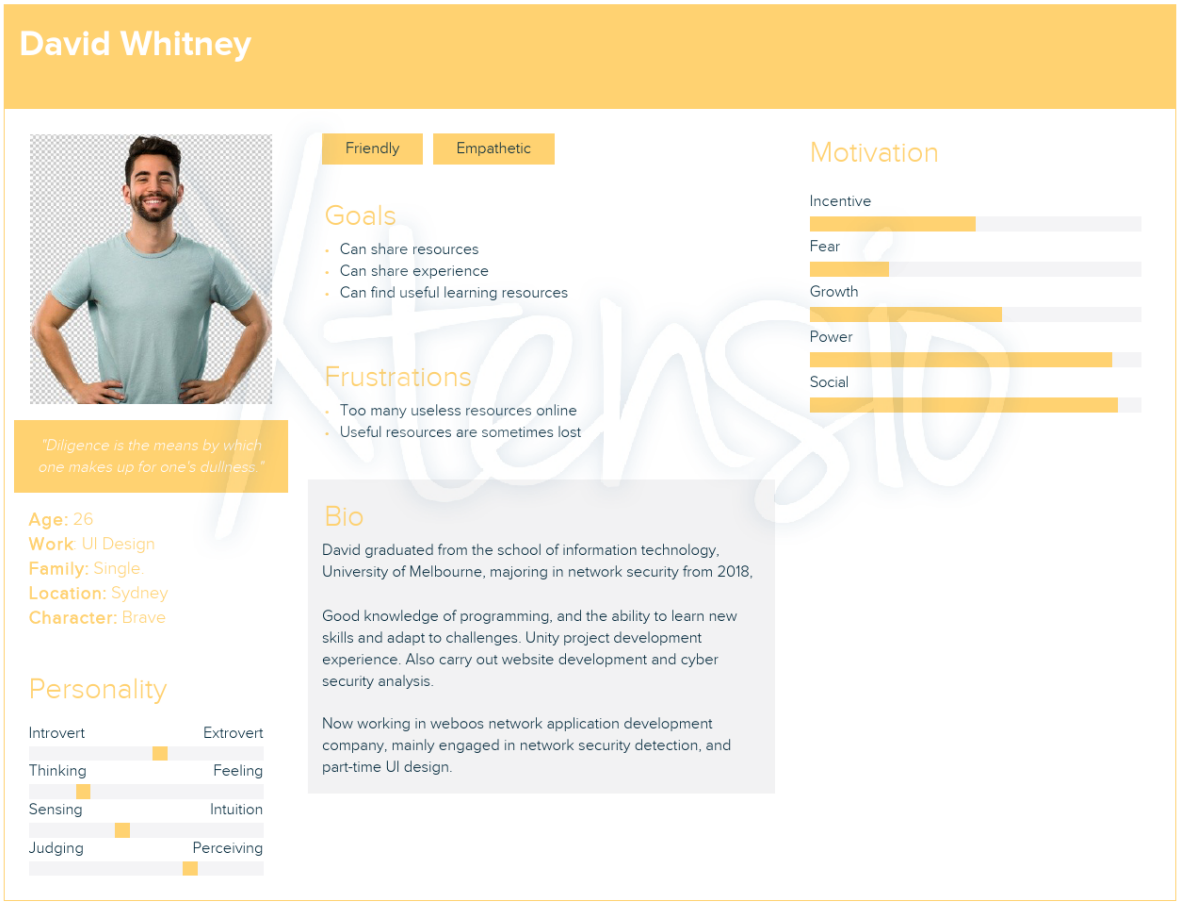
Page Description:	User profiles that define different users for the product, their goals, motivations and characteristics.
Page Status:	COMPLETE

We mainly choose 4 typical users. They are mainly 3 types of students, a lecturer, and a tutor. The details are as follows:

- Student (Current)
- Student (Experienced)
- Teaching staff
- Moderator

Student (Current):

Student (Experienced):



Teaching Staff:

Made With Xtensio The new way to create, manage and share visual documents and presentations.

Lonnie Klein

"Professors known as outstanding lecturers do two things; they use a simple plan and many examples."

Age: 45
Work: Lecturer
Family: Married
Location: Melbourne, VIC
Character: Eloquent

Personality

Introvert — Extrovert
Thinking — Feeling
Sensing — Intuition
Judging — Perceiving

Technological **Curious** **Cooperative** **Clever**

Goals

- To teach a student about the latest technology
- To bring useful knowledge and resources to students for self-learning

Frustrations

- Have some ideas to improve it but too busy to do it alone
- There are too many resources over the internet but It is very hard to check the quality

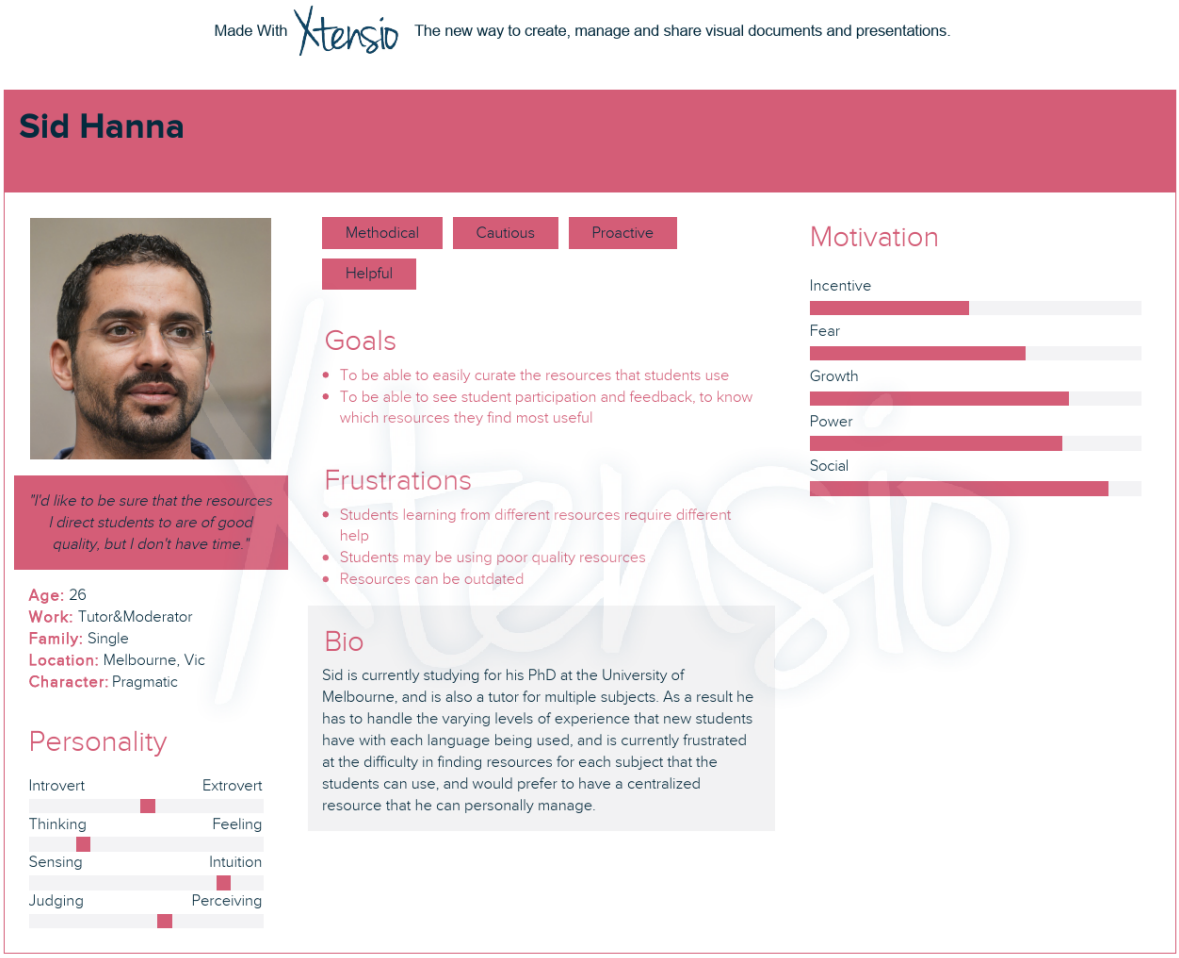
Bio

Lonnie Klein received his PhD in Computer Science in 2005 from MIT(*Massachusetts Institute of Technology*). In 2006 he joined the CIS Department at the University of Melbourne, where he served as a lecturer. He has taught in various subject in computer science and software engineer. He has experience in teaching students for more than 15 years

Motivation

Incentive
Fear
Growth
Power
Social

Moderator:



reference:

- https://confluence.cis.unimelb.edu.au:8443/display/SWEN900132020RS/Personas

User Stories

Page Description:	User stories specify the tasks to be completed through the development with the story points and priority
Page Status:	DONE

Legends

- Priorities
  1. HIGH — Must have
  2. MEDIUM — Great to have
  3. LOW — Minor / Not essential
- Story Point
  1. 5 PT — Major focus required
  2. 3 PT — Some effort required
  3. 1 PT — Simple and easy

3PT story is **Thrice** as long as a **1PT** story

User Stories (Version 1&2 - 27 Mar 2021)

<All Students> = <Past students + Current students + Moderators>.

*** Priorities are provisional and pending confirmation by Product Owner.**

Category	ID	As a	I can	so that	Story Point	Priority *
Site Admin	1.1	Teaching staff	Access **standalone admin mode** to manage user accounts, change settings, activate maintenance mode, etc.	Admin functionalities are centralised, easy to access, and not dispersed in hundreds of different pages.	5 PT	HIGH
	1.2	Teaching staff	Set site-wide or subject-wide **announcements** or notifications.	I can announce stuff like distinguished contributors and new resources.	1 PT	LOW
	1.3	Teaching staff	Access **applications APIs** for automation, CRUD, etc.	I can have more control of the application and do more fun stuff.	1 PT	LOW
	1.4	Teaching staff Moderator	Access **standalone moderation mode** to approve, post, modify, and/or delete resources.	Admin functionalities are centralised, easy to access, and not dispersed in hundreds of different pages.	3 PT	HIGH
Resources	2.1	All Students	Post resources in the forms of **Markdown, Rich-Text, Source Codes, PDF documents, Images, file attachment, etc.,** which requires approval from moderators or teaching staff to be publicly visible.	Resources are diverse and rich, and manageable by moderators and teaching stuff.	3 PT	HIGH
	2.2	Teaching staff	Directly add high-priority resources, which are marked with the **Teaching Stuff** badge.	Students have higher chances to notice these resources.	1 PT	MEDIUM
	2.3	Teaching staff All Students	**Upvote/downvote** resources or comments.	Resources with high-quality standards are identified and rewarded.	1 PT	MEDIUM

	2.4	Teaching staff All Students	Comment on resources or **comments** in forms of **Markdown, Rich-Text, Source Codes, PDF documents, Images, file attachment**, etc.	Resource sharing is not one-way only, but an interactive process.	3 PT	HIGH
	2.5	Teaching staff All Students	Apply **filters** such as subject, keywords, tags, etc.	I can easily search for the resources I required.	5 PT	HIGH
	2.6	Teaching staff Moderator	Directly **rank** resources and/or assign priorities.	Resources with high-quality standards are directly identified and rewarded.	1 PT	HIGH
	2.7	Teaching staff All Students	Add **tags** to my resources.	Resources are categorised and easily located.	1 PT	HIGH
	2.8	Teaching staff All Students	**Search** for resources, comments by keywords or RegEX.	I can easily search for the resources I required.	5 PT	HIGH
	2.9	Teaching staff All Students	**Bookmark** resources and create reading lists, such as favourites, read-later, etc.	I can easily manage/find resources I came across.	3 PT	MEDIUM
Rewards/ Promotion	3.1	Teaching staff	**Promote active participants to moderators**.	Active participants are identified and rewarded.	3 PT	HIGH
	3.2	All Students	Receive rewards for **active participation**, in the forms of **EXP/Level or badges.**	Active participants are identified and rewarded.	3 PT	MEDIUM
Profile	4.1	Teaching staff All Students	**Update my profile**, such as nicknames.	I can customise my account.	3 PT	LOW
	4.2	Past Students	Receive a badge, identifying me as **Past Students** which highlights the resources I submitted.	Past students' resources are easily identified.	1 PT	MEDIUM
Others	5.1	Teaching staff All Students	Login via **University SSO** or other **OAuth** services, and my login can be remembered.	I won't be frustrated by typing my passwords for hundreds of times in one day.	5 PT	MEDIUM
	5.2	Teaching staff All Students	Access **user guides**, **tips**, and **search** for complicated interfaces such as editors and admin panel.	I will feel more guided and acquire help when necessary.	3 PT	MEDIUM

	5.3	Teaching staff All Students	Add **reminders** to remind me of future events.	I can take notes, remind myself to read documents, etc.	3 PT	MEDIUM
	5.4	All Students	Add or delete **friends.**	I can make friends.	3 PT	MEDIUM
	5.5	All Students	Send or receive **private messages**	I can communicate with other students privately.	3 PT	MEDIUM

Acceptance criteria

Page Description:	List the acceptance criteria for each corresponding user story
Page Status:	DONE

Acceptance criteria (Version 1&2)

ID	User Story	Criteria
1.1	Access **standalone Admin Panel** to modify resources, change settings, activate maintenance mode, etc.	1. Given I am teaching stuff/moderator. 2. I click on the **Admin Panel** link. 3. I enter the **Admin Panel** which shows available options/settings based on my privilege. 4. I click on any of these options, a prompt appears to confirm my action. 5. I click Yes, a prompt appears to acknowledge the results of my action.
1.2	Set site-wide or subject-wide **announcements** or notifications.	1. Given I am teaching stuff, and I am in **Admin Panel.** 2. I click on **Post an Announcement**, a form appears asking for contents of the announcement, and the target of my Announcement, which is a drop-list containing Site-wide, and all the subjects I manage. 3. I fill in the form and click Continue, a prompt appears to confirm my action. 4. I click Yes, a prompt appears to acknowledge the results of my action.
1.3	Access **applications APIs** for automation, CRUD, etc.	1. Given I am teaching stuff. 2. A link named **API reference documents** appears in the **Admin Panel.** 3. I can access these APIs with my user account.
2.1	Share resources in forms of **Markdown, Rich-Text, Source Codes, PDF documents, Images, file attachment**, etc.	1. Given I am a student or teaching staff. 2. I click on the **Post** button on top of any page. 3. A form appears, asking for the subject I am posting to, the title, and the content of my post, where I can insert **Markdown, Rich-Text, Source Codes, PDF documents, Images, file attachment**, and they are parsed instantly and previews appear. 4. I fill in the form and click Continue, a prompt appears to confirm my action. 5. I click Yes, a prompt appears to acknowledge results of my action. 6. When I view the post, all contents of my resources should be displayed correctly.
2.2	Directly add high-priority resources, which are marked with **Teaching Stuff** badge.	1. Given I am teaching stuff. 2. I click on the **Post** button on top of any page. 3. A form appears, asking for the subject I am posting to, the content of my post. 4. I fill in the form and click Continue, a prompt appears to confirm my action. 5. I click Yes, a prompt appears to acknowledge results of my action. 6. When I view the post list or the post, a **Teaching Stuff** badge should appear on the left side of the post title.
2.3	**Upvote/downvote** resources or comments.	1. Given I am a student or teaching staff. 2. I am viewing a post published by myself, two buttons indicating **Upvote/downvote** app ears in grey, and not-clickable. Below the buttons shows a **vote count.** 3. I am viewing a post published by others, two buttons indicating **Upvote/downvote** appe ars in blue/red, and clickable. Below the buttons shows a **vote count.** 4. I click on any of these buttons, a simple animation appears informing my action. The **vo te count** increases when upvoted and decreases when downvoted.
2.4	Comment on resources or **comments** in forms of **Markdown, Rich-Text, Source Codes, PDF documents, Images, file attachment**, etc.	1. Given I am a student or teaching stuff. 2. I click on the **Reply** button on bottom of a post/comment. 3. A form appears, asking for my input, where I can insert **Markdown, Rich-Text, Source Codes, PDF documents, Images, file attachment**, and they are parsed instantly and previews appear. 4. I fill in the form and click Continue, a prompt appears to confirm my action. 5. I click Yes, a prompt appears to acknowledge results of my action. 6. When I view the comment, all contents of my resources should be displayed correctly.

2.5	Apply **filters** such as subject, keywords, tags, etc.	1. Given I am a student or teaching stuff. 2. I click on the **Filter** button on the left of the post list. 3. A form appears, asking for the nature and parameters of the filter. 4. I fill in the parameters and click **Apply**, the post list is instantly refreshed, and filtered correctly. 5. I client on **Remove Filter** button, the post list is instantly refreshed, and not filtered.
2.6	Directly **highlight** resources.	1. Given I am a moderator or teaching stuff. 2. I click on the **highlight** button on top of any post, a prompt appears to confirm my action. 3. I click Yes, a prompt appears to acknowledge results of my action. 4. When I view the post list or the post, a **Highlighted** badge should appear on the left side of the title of the post. 5. I click on **Remove Highlight** button, A prompt appears to confirm my action. 6. I click Yes, a prompt appears to acknowledge results of my action. 7. When I view the post list or the post, the **Highlighted** badge should not appear.
2.7	Add **tags** to my resources.	1. Given I am a student or teaching stuff. 2. I can view all my tags on the bottom of any post. 3. I click on the **Modify** button on the bottom of my own post (student), any post (moderator/teaching stuff). 4. A prompt appears, displaying all the tags I have already set, and a form asking for new tags. 5. I fill in the new-tag-form and click **Add**, prompt does not disappear, and the tag is added to the tag list. 6. I click on **X** button on any tag, tag is instantly removed.
2.8	**Search** for resources, comments by keywords or RegEX.	1. Given I am a student or teaching stuff. 2. A search form appears on the top of the post list. 3. I fill in the form with keywords or RegEX, and I click on the **Search** button on the left side of the form. 4. Only the post list refreshes, showing the search results. 5. I changed my keywords or RegEX, and I click on the **Search** button on the left side of the form. 6. Only the post list refreshes, showing the search results. 7. I click on the **X** button **inside and on right side** the search form 8. Only the post list refreshes, showing the original list.
2.9	**Bookmark** resources and create reading lists.	1. Given I am a student or teaching stuff. 2. I am viewing a post published by anyone, a button indicating **Bookmark** appears on bottom of a post. 3. I click the button, a simple animation appears informing my action, and the **drop list** ap pears showing my reading lists. 4. The post is automatically saved to the **default reading-list**, when I click on an another reading list, the post is moved to that list.
3.1	**Promote active participants to moderators**.	1. Given I am teaching stuff, and I am in **Admin Panel.** 2. I click on **User Management**, a list shows all the active users in the subjects I manages. 3. I click on **sort by EXP/Level,** a new lists shows details of top users. 4. I click Promote, a drop list appears asking for the subject I am giving privilege, which are sorted by the subject-wise score. 5. I fill in the form and click Apply, a prompt appears to confirm my action. 6. I click Yes, a prompt appears to acknowledge the results of my action.
3.2	Receive rewards for **active participation**, in forms of **EXP/Level or badges.**	1. Given I am a student. 2. I can view anyone's **EXP/Level or badges** in their profile. 3. I can view anyone's **EXP/Level or badges** on the right side of their post 4. The EXP increments when I post resources, receive upvotes, post comments, get highlighted by moderators, etc.

4.1	Receive a badge, identifying me as **Past Student** which highlights the resources I submitted.	1. Given I am a past student. 2. A **Past Student** badge is granted to me after I passed the subject. 3. When I post resources or comments, a **Past Student** badge should appear on the left side of the post title.
4.2	**Update my profile**, such as nicknames.	1. Given I am a student or teaching staff, and I am at my profile page. 2. I click on the **Edit** button at the bottom of my profile 3. A form appears, displaying all my details, and is editable. 4. I fill in the form and click Continue, a prompt appears to confirm my action. 5. I click Yes, a prompt appears to acknowledge the results of my action. 6. When I view my profile, all details should display correctly.
5.1	Login via **University SSO** or other **OAuth** services, and my login can be remembered.	1. Given I am a student or teaching staff, and I am at the login page. 2. I click on the **MelbUni** button or other providers button. 3. The page redirects to the OAuth provider, where I finished my login process. 4. The page redirects back to the application, and I am already logged in. a. I close the browser and reopened the application, I am still logged in.
5.2	Access **user guides**, **tooltips**, and **hint popovers** for complicated interfaces such as editors and admin panel.	1. Given I am a student or a teaching staff, I could be on any page and needed help with some functions or pages. 2. I click on the **Help** button on the top right of a page. 3. It shows me 3 options which is **user guides**, **tooltips**, and **hint popovers.** 4. It should give me related information when I click on one of them.
5.3	Add **reminders** to remind me of future events.	1. Given I am a student or a teaching staff, and I am at the **Reminders** page. 2. I can schedule a reminder by choosing a date, time, and writing a topic/comment. 3. I can also see the reminders I have already scheduled. 4. When the shceduled time comes, it should alert me with the content.
5.4	Add or delete **friends.**	1. Given I am a student. 2. I can add or delete a friend by clicking an Add or Delete button.
5.5	Send or receive **private messages**	1. Given I am a student. 2. I can send or receive private messages with other students by choosing a recipient, typing my message in the text area and clicking the Send button.

Interview questions with swen90013 group users

Questions:

1. What functions/features would you like to have?
2. What do you think how the system looks like? (about design)
3. If let you design the system, how would you design it? (could be both functions & design)
4. Would you be happy if the resources you provide need to get approval from an admin? how do you feel about that, and why?
5. Would you normally share and discuss the resources on the website with others? If not, in what situation you will be sharing and discussing the resources on the website?
6. What should we do to make you feel that you are welcomed and encouraged to participate on the resource website?
7. what can we do to help you share and discuss the resources on the website in order to learn more?
8. Can you imagine yourself using this product? Why/Why not?
9. Have you used/tried similar products, what's good and bad?
10. Do you want to be a student moderator? What privileges would you like to have?
11. What kind of resources would you expect/want? (Video, Image, Paper, Articles, etc)
12. Which language do you prefer
13. What do you want to show on your profile

Design

Prototype

Page Description:	A prototype is a digital simulation or demo of a product or service that enables you to test assumptions and virtually explore a product before it's built.
Page Status:	COMPLETE

Since this project is built on top of the previous SWEN90013 project, some functions are the same and have already been implemented such as login as different roles, create subjects and topics for teaching staff, view and react to the topics for students. So the prototype of these functions will be re-used here for this project. However, the improvement of design for some pages like the login page and home page of students and teaching staff, together with new functions are demonstrated on this page.

The previous prototype and the additional prototype can be found under the "Previous digital Prototype Link" and the "Additional prototype" sections below.

Previous digital Prototype Link

There are two versions of the prototype which are student and moderator versions. Those versions have slightly different.

	Axure Link	Information
Student Version	https://zv8nzm.axshare.com	**Student Account** • Username: miranda@student.unimelb.edu.au • Password: 12345678
Moderator Version	https://bwh6ha.axshare.com	**Moderator Account** • Username: admin@unimelb.edu.au • Password: 12345678

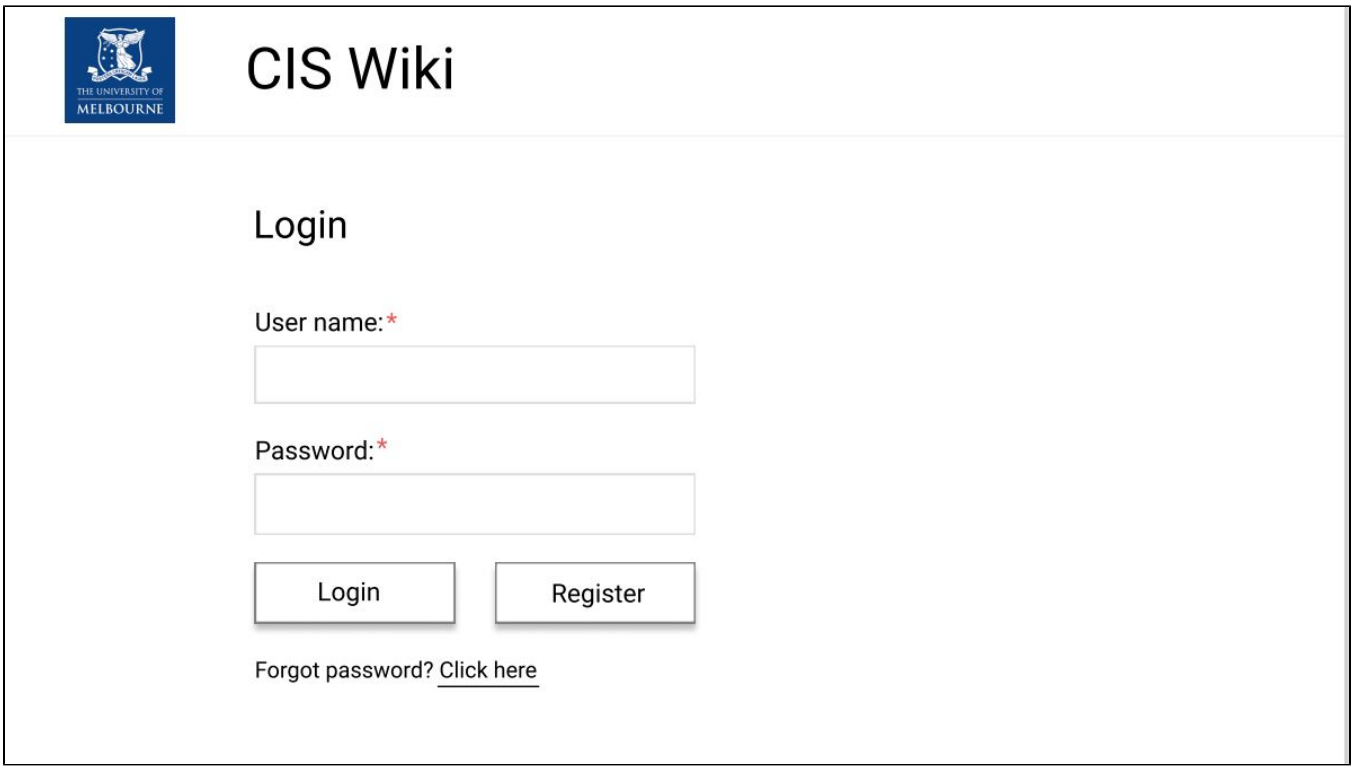
Reference: Digital Prototype

Additional Prototype

The improvement of design for some pages and the new function pages are presented in this section.

Login

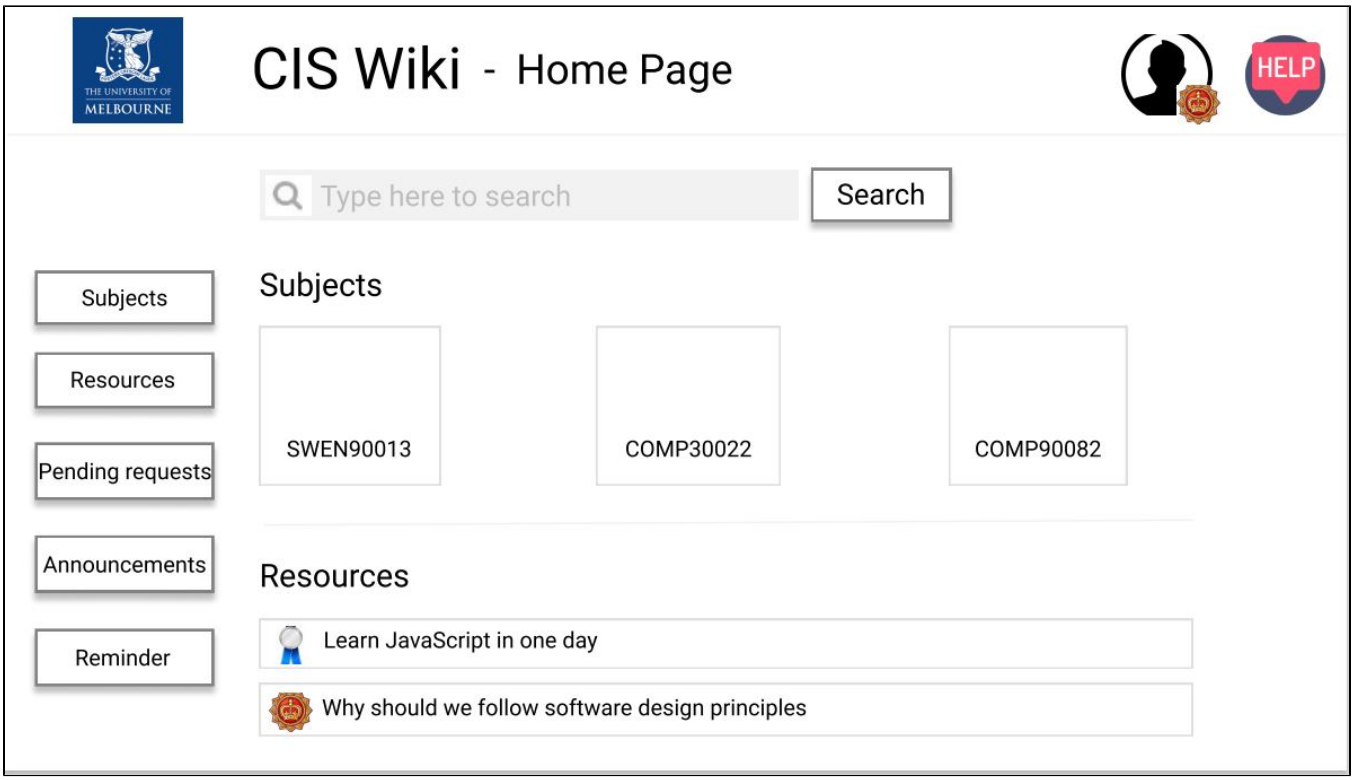
This page is re-designed to have a better user experience and added a new function for reset passwords if the password is forgotten.



Teaching staff home page

This page is re-designed to improve the UI and show the new functionalities for teaching staff. For example, the exclusive badge for teaching staff, the help function, the announcements function, and the reminder.

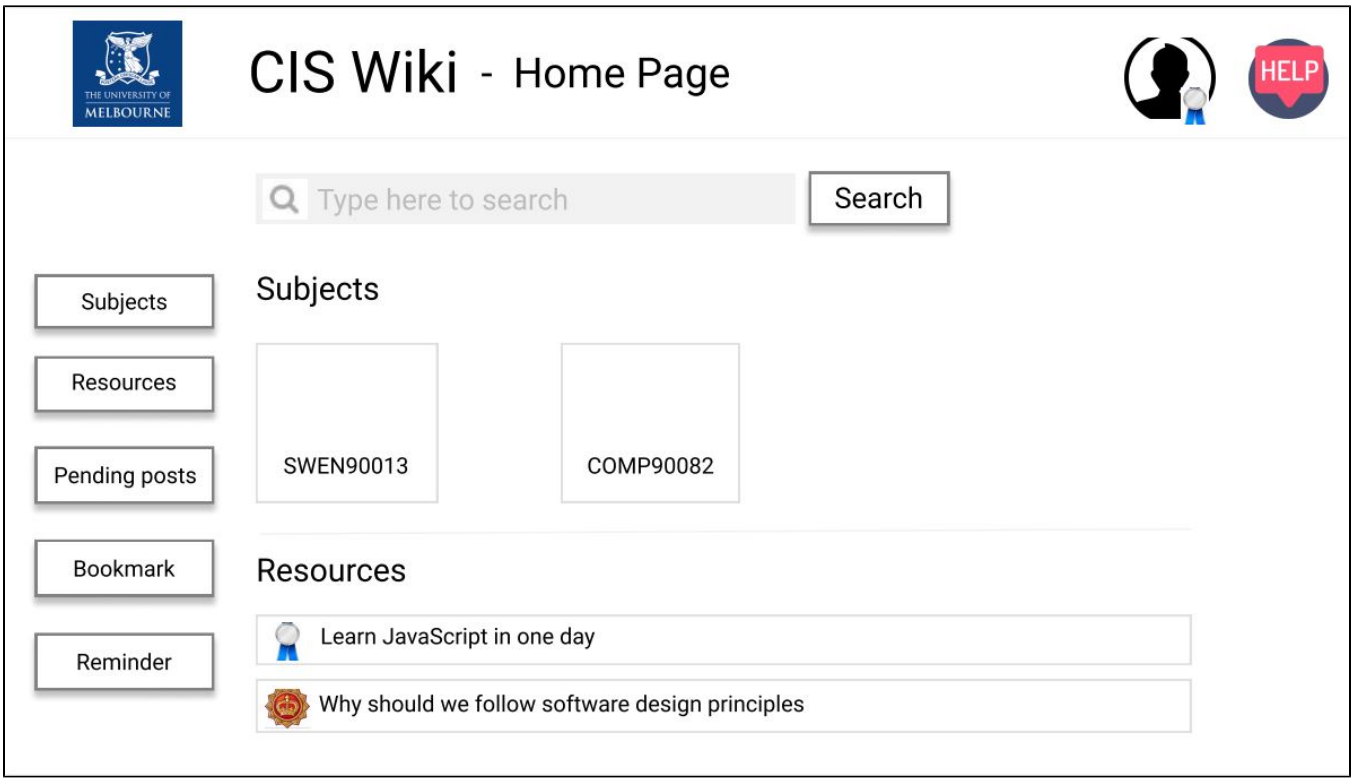
- the exclusive badge: will be shown when the teaching staff have some actions eg announcements and post new resources for students. It is very convenient for students to identify which resources were posted by the teaching staff.
- the help function: to provide a user-friendly user experience. The help function offers 3 different functions: User guides, Tips, and Search. The help function will be explained at the end of this section.
- the announcements function: to propagate useful information to related groups eg students, teaching staff, or moderators.
- the reminder: provides convenient service for teaching staff to prevent forgetting important things. This function will be introduced later.



Student home page

This page is re-designed to improve the UI and show the new functionalities for students. For example, the different badges showing the different levels of a student, the help function, the bookmark function, and the reminder.

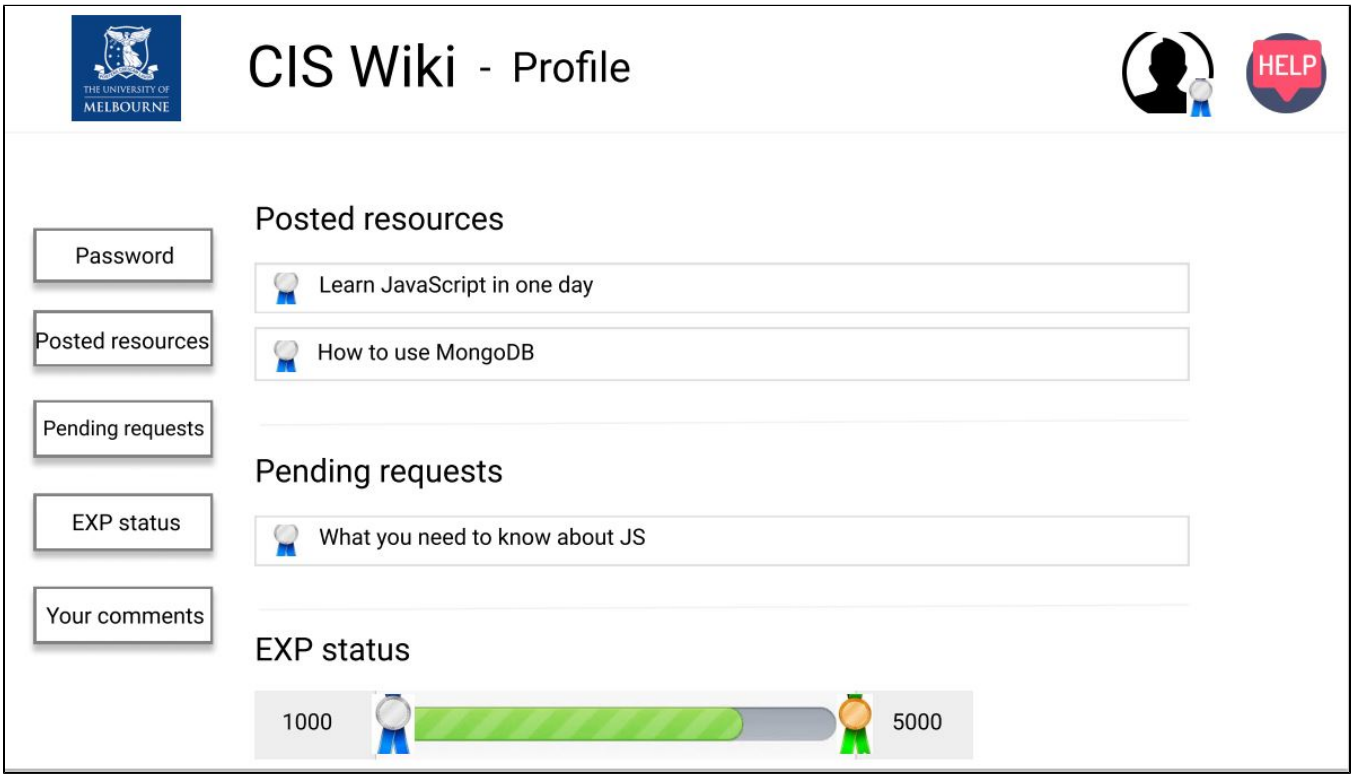
- the badge: is differentiated by a gold medal, silver medal, and bronze medal for showing 3 different levels to encourage students' participation.
- the help function: to provide a user-friendly user experience. The help function offers 3 different functions: User guides, Tips, and Search. The help function will be explained at the end of this section.
- the bookmark function: can create many different lists for different purposes. For instance, a list for bookmark favourite resources and a list for "Read later".
- the reminder: provides convenient service for students to prevent forgetting important things. This function will be introduced later.



Student profile page

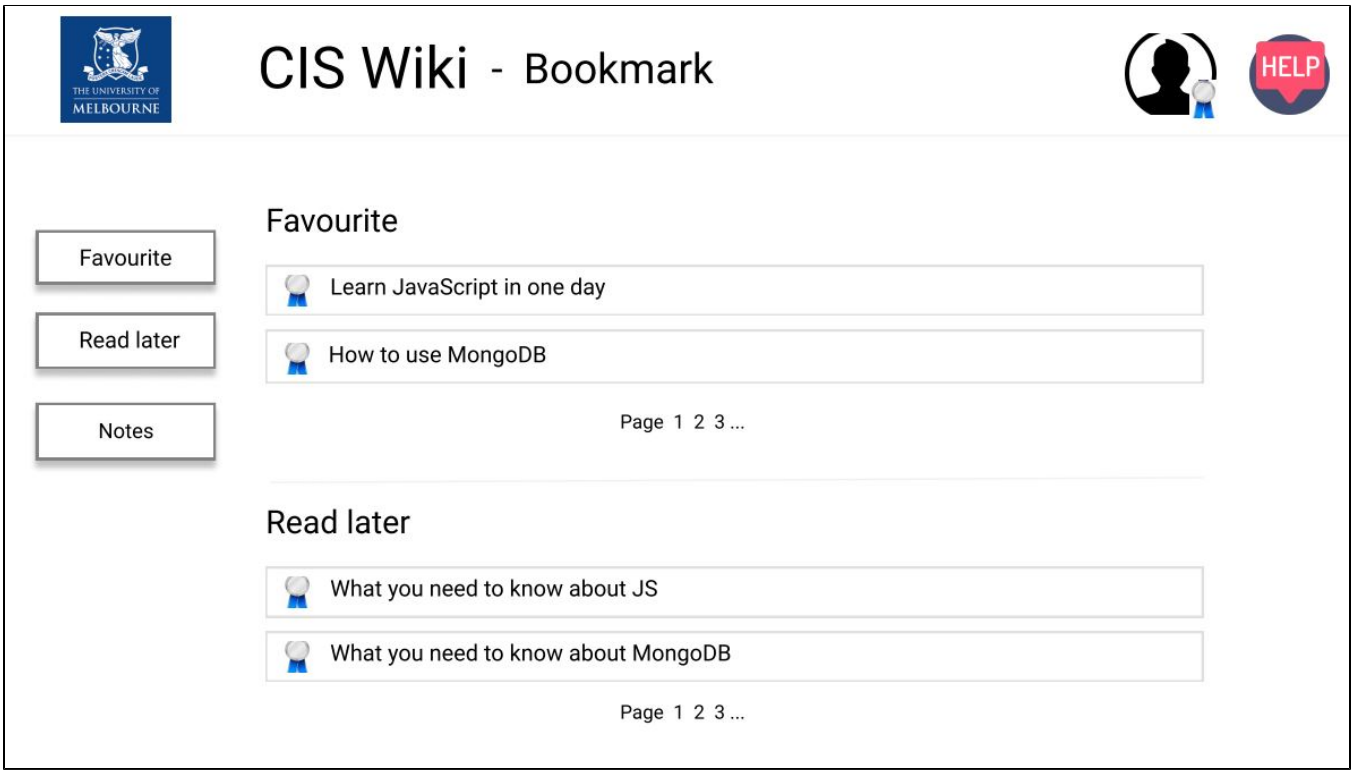
This page is re-designed to improve the UI and show the new functionalities for students. For example, the EXP status and the help function.

- the EXP status: shows the progress of upgrading to the next level.
- the help function: to provide a user-friendly user experience. The help function offers 3 different functions: User guides, Tips, and Search. The help function will be explained at the end of this section.



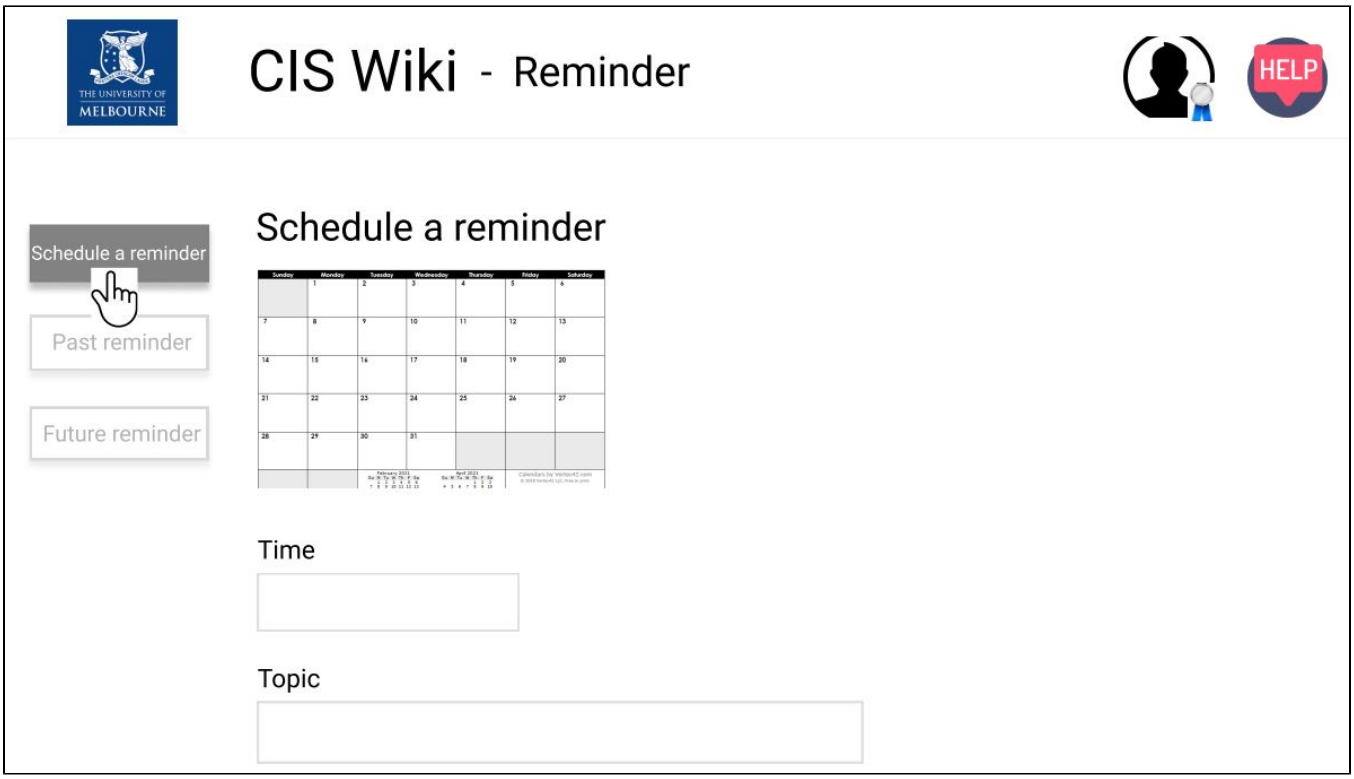
Student bookmark page

This page is to help students to manage the bookmarked resources. The lists can be created for different purposes such as Favourite, Read later, and notes, etc.



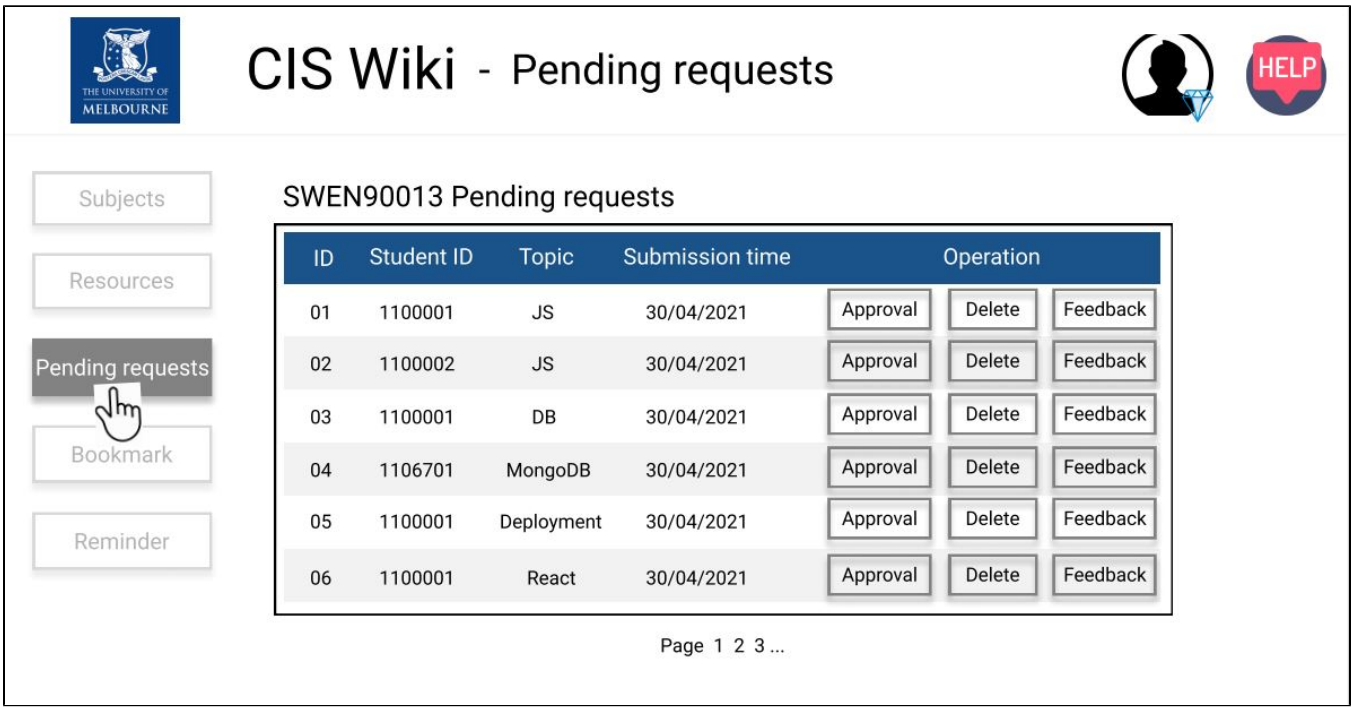
The reminder page for both teaching staff and students

This new function is to help the teaching staff and students organising events in an efficient manner. On the "Schedule a reminder" page, the user can choose a date, time, and topic or comments for the event and the system will remind them at the chosen time.



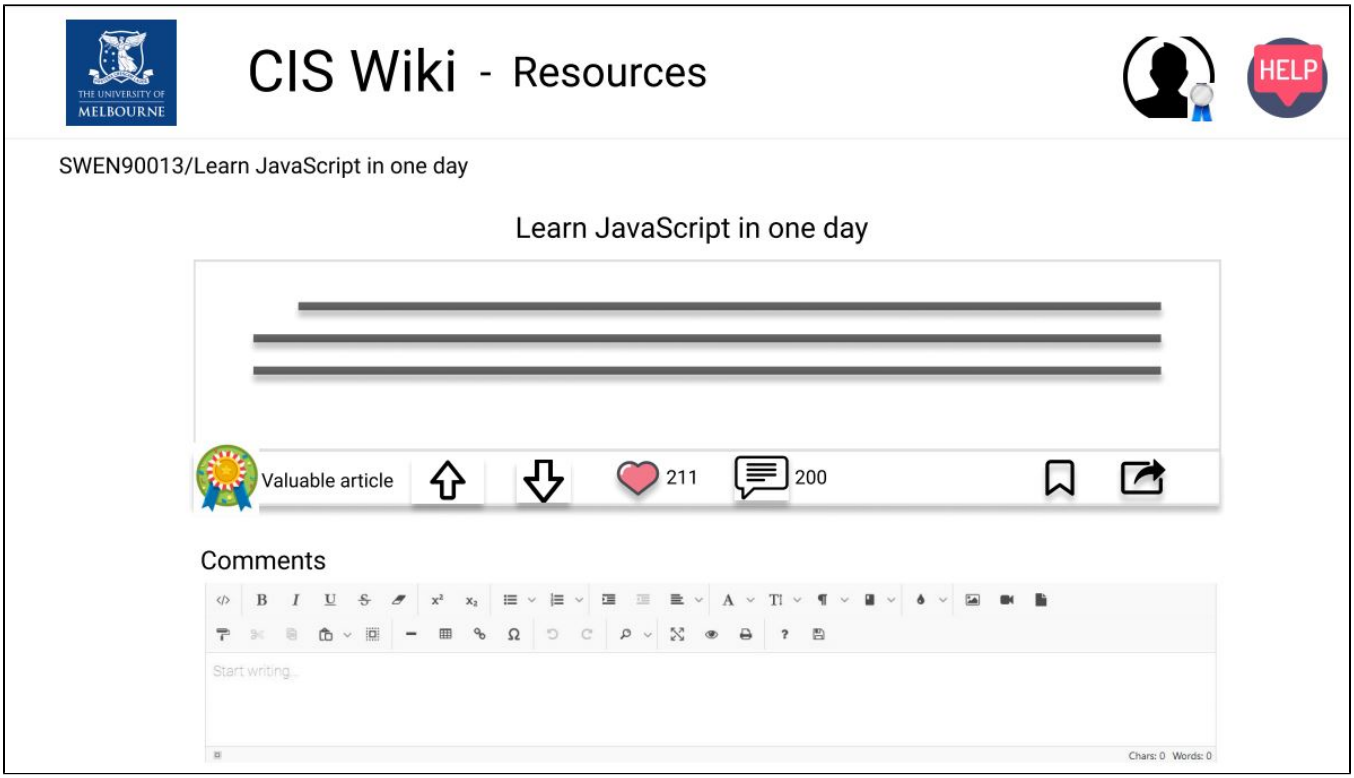
The Pending requests page for the moderator

This page is to show the pending requests page for the promoted student as a moderator and to show the special badge for the promoted student which is a diamond to differ the role since there is a different badge for teaching staff and students, respectively. The pending requests page is very similar to the previous project. However, the Feedback function is added for the operation section in case if any message needed for the specific request.



The resources page

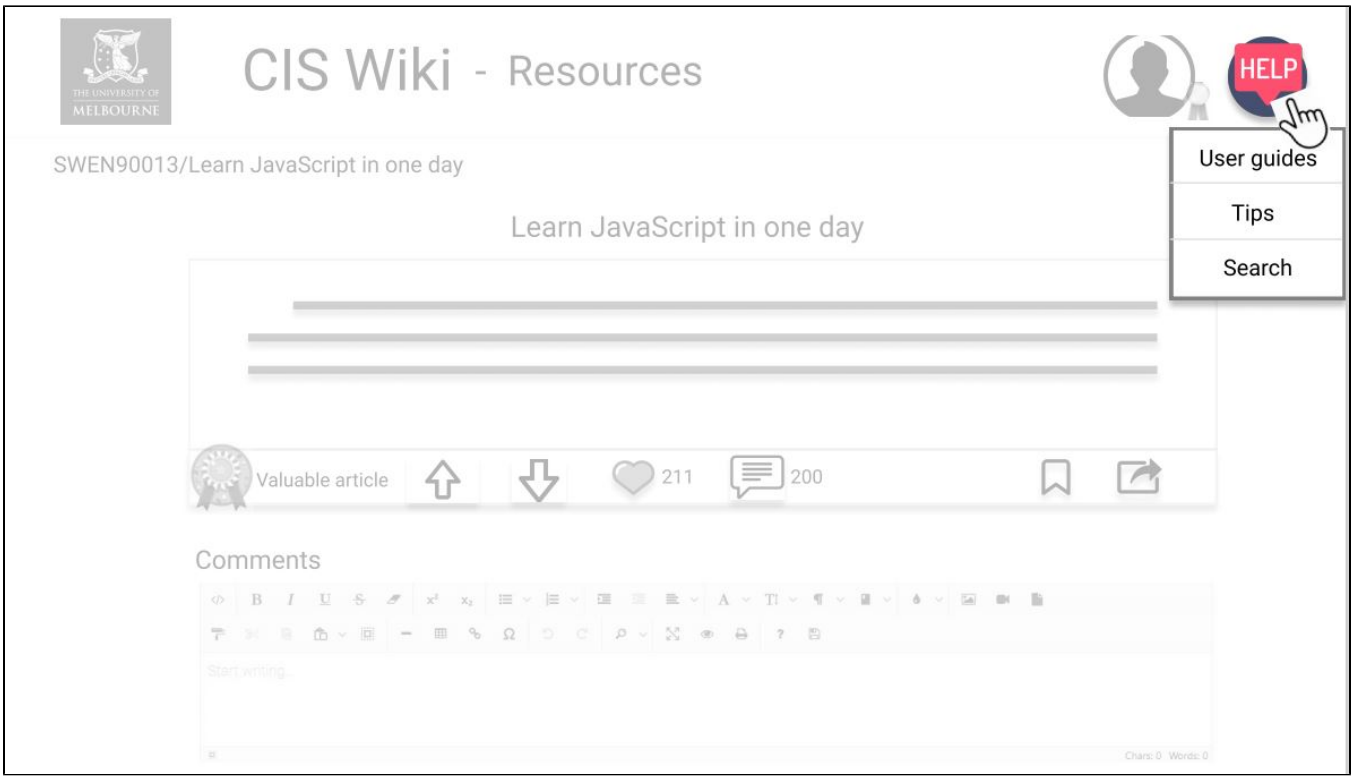
This page is re-designed to improve the UI and added additional functions like valuable article label, upvoting, and downvoting. Students can view resources, also can upvote, downvote, like, comment, bookmark, and share resources. The resources also can be labeled as valuable articles by the teaching staff.



The help page

This new function is added for both students and teaching staff to provide a user-friendly function when needed. It offers User guides, Tips, and Search functions to the user.

- the user guides: will give a brief introduction to the user to show some important functions on the page or the website. An example has been provided below.
- the tips: will provide some tips for using this website.
- the search: offers a special search function that is not used to search for the resources but the operations. For example, if you want to set a reminder and can't find where it is, you can certainly use this search function to help you locate the Reminder.

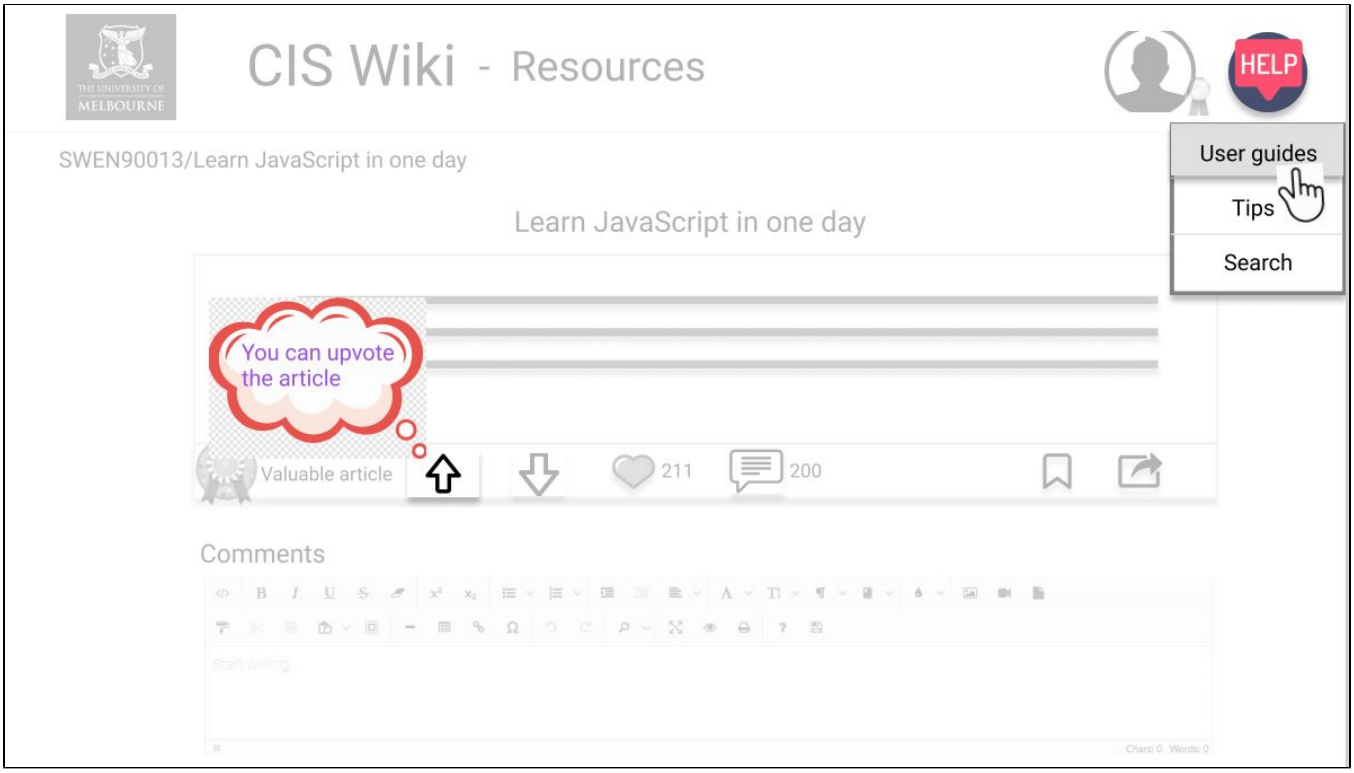


An example of using User guides from Help

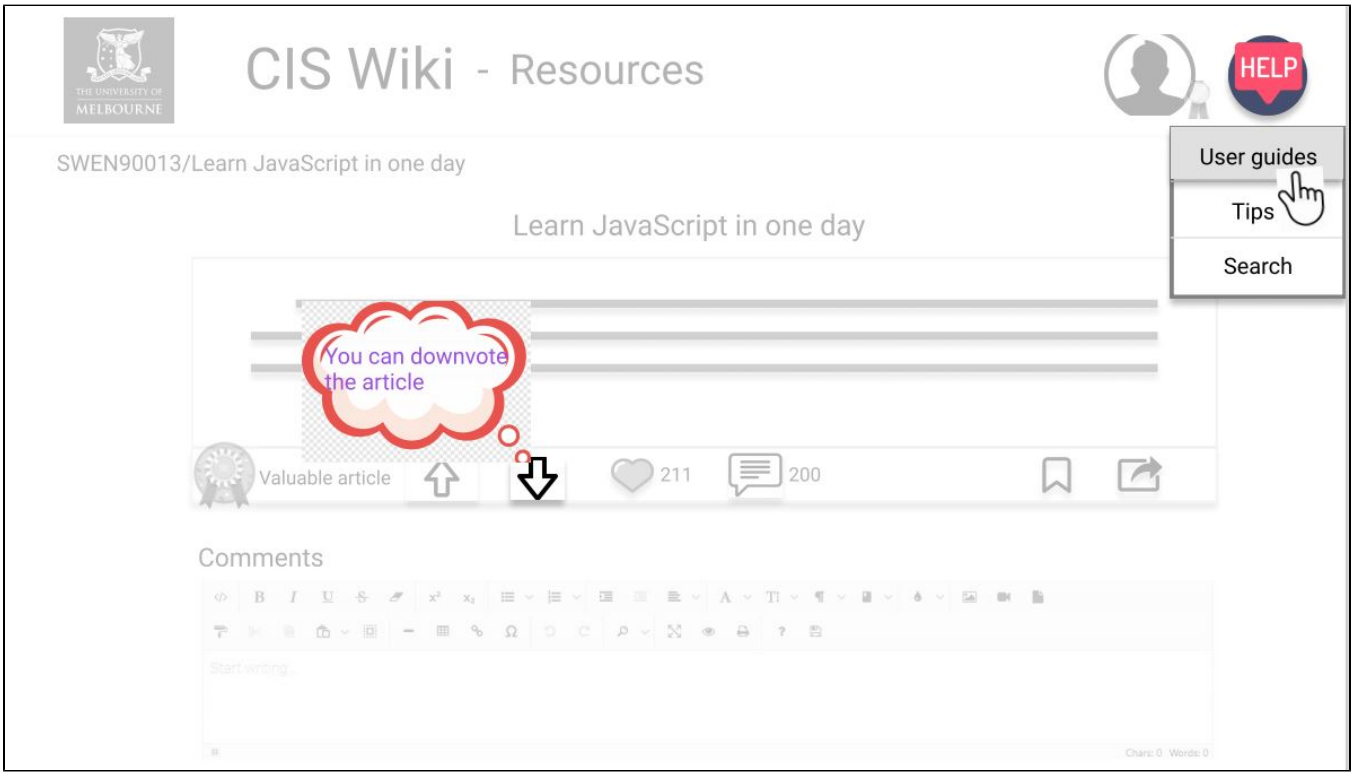
After the user clicks User guides from the Help drop-down menu, the system will show and explain to the user about some functions. The user can click through to see all the guidance or press ESC to exit this function. For example,

(1) Firstly, the system shows what the valuable article is

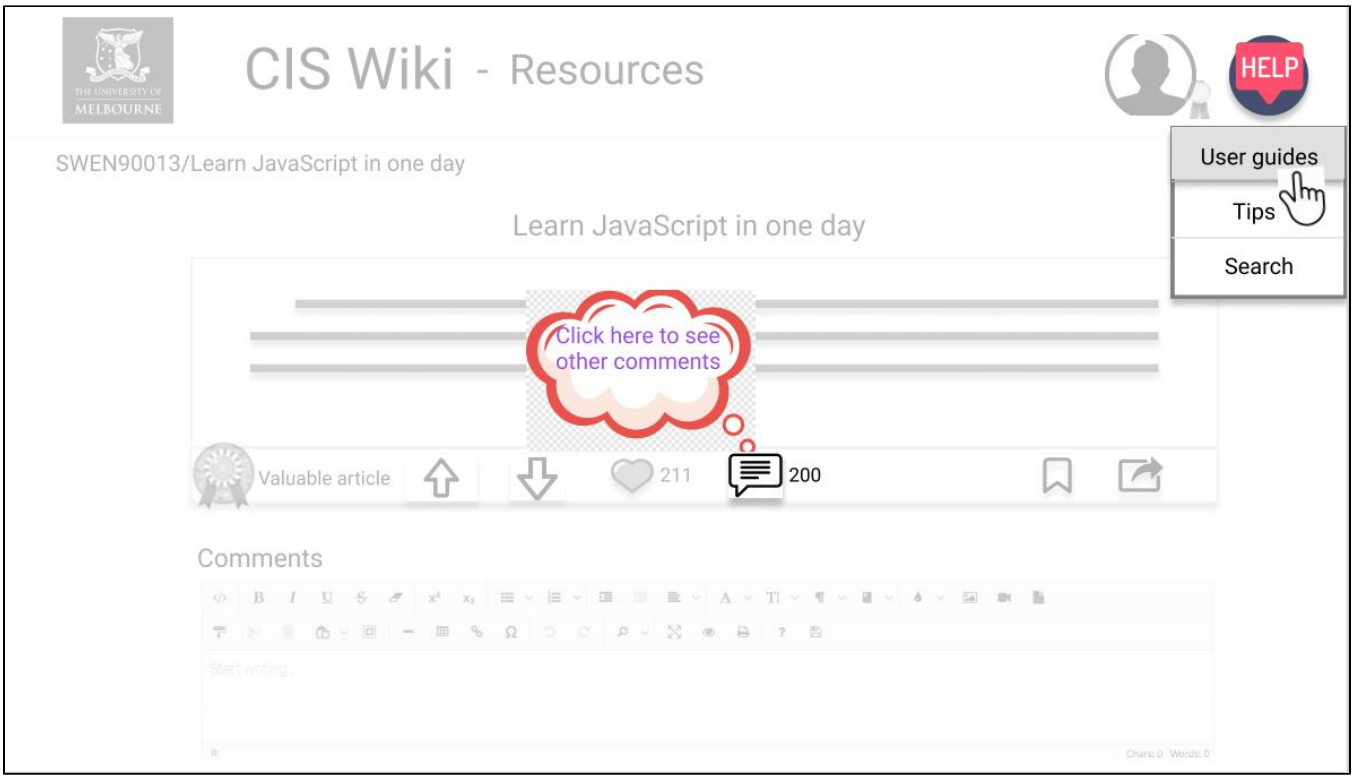
(2) Secondly, the system shows what the up pointed error is and its function



(3) Then, the system shows what the down pointed error is and its function



(4) After that, the system shows what the highlighted function is


CIS Wiki - Resources
HELP
User guides
Tips
Search
SWEN90013/Learn JavaScript in one day
Learn JavaScript in one day
Click here to see other comments
Valuable article
211
200
Comments
Start writing...

Architecture

Architecture diagram

This Architecture Diagram refers to the previous project since all the components are not changed and all the structures are the same. The original work can be found here: High-level Architecture Diagram.

Link to edit: https://drive.google.com/file/d/1YdoBjteQwrmcFpqA2MAGQJd9JjGjdvGj/view?usp=sharing

The 5-Layer Architecture generally works at a high level of abstraction. Each layer of the layered architecture pattern has a specific role and responsibility within the application. For example, a presentation layer would be responsible for handling all user interface and browser communication logic, whereas a business layer would be responsible for executing specific business rules associated with the request. Each layer in the architecture forms an abstraction around the work that needs to be done to satisfy a particular business request. For example, the presentation layer doesn't need to know or worry about *how* to get customer data; it only needs to display that information on a screen in a particular format. Similarly, the business layer doesn't need to be concerned about how to format customer data for display on a screen or even where the customer data is coming from; it only needs to get the data from the persistence layer, perform business logic against the data (e.g., calculate values or aggregate data), and pass that information up to the presentation layer.

One of the powerful features of the layered architecture pattern is the *separation of concerns* among components. Components within a specific layer deal only with the logic that pertains to that layer. For example, components in the presentation layer deal only with presentation logic, whereas components residing in the business layer deal only with business logic. This type of component classification makes it easy to build effective roles and responsibility models into your architecture, and also makes it easy to develop, test, govern, and maintain applications using this architecture pattern due to well-defined component interfaces and limited component scope.

UI Layer

Provides external access to the solution (specifically the Business Logic, and by extension the underlying data). This includes User Interfaces and can include System Interfaces such as Http Handlers, JSON Services, and so on.

Control Layer

Control the user interface layer and business logic layer for data interaction.

Service Layer

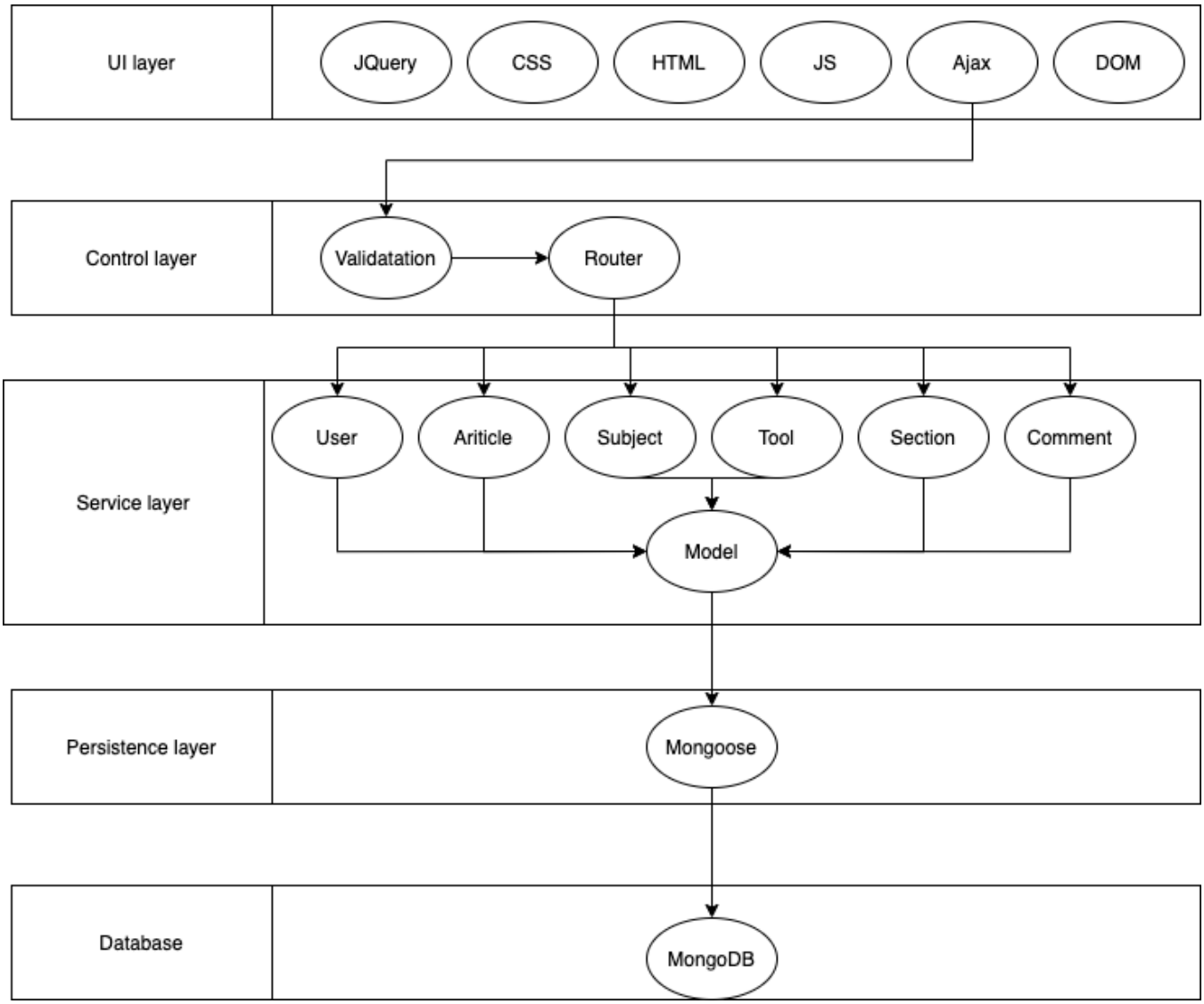
It contains all the core application logic, with a focus on implementing business rules and containing the business value.

Persistence Layer

Control of transactions, direct operation of the database, CRUD operation of the data (i.e. insert, delete, update, and select).

Database

An organized collection of structured information, or data, typically stored electronically in a computer system.



References

https://morphological.files.wordpress.com/2011/08/5-layer-architecture-draft.pdf

https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html

Component Diagram

The Component Diagram refers to the previous project since all the components are not changed and all the structures are the same. The original work can be found here: Component Diagram.

Internal Components

The Diagram below is the Component Diagram for our system. There are 6 major components:

1. The Front-end Components,
2. The Front-end Router,
3. The Back-end Server,
4. The Back-end Router,
5. The Back-end Controller,
6. The Back-end Middleware,
7. The Back-end Proxy,
8. The Database

The Front-end Components

In the current system, there are 11 sub-component in the front-end, and they are shown below in the diagram. Generally, each of the front-end components corresponds to the prototype and goal model in the system. The front-end components are responsible for displaying the static views of the pages in regard to their corresponding functionality. Moreover, the components will send out Http requests according to the user inputs with the help of the Ajax external framework, waiting for the response from the back-end and react to display the corresponding view to the users.

The Front-end Router

The front-end router is responsible to connect the different components. The router is also a data transformation layer. The component can use the router to pass data and state and contact each other.

The Back-end Server

The backend server works as a web server, it listens to the request sent by each connected client and be ready to respond to it.

The server also acts as a hub. Since we use different URL prefix in the front-end to distinctly identify different functionalities in the system, whenever there is an incoming HTTP request from the front-end, the server identifies its prefix of the URL address and directs it to the routers that are responsible for the certain URL prefix to process the request and respond to it.

The Back-end Router

The Back-end routers are actually lower-level servers. It takes all the Http requests of certain functionality, and call the corresponding controllers to handle it based on the suffix of the request.

The Back-end Controller

The Back-end controllers are sets of functions/methods to handle the request and create the corresponding HTTP responses in JSON as return values to send back to the front-end through the router and the server.

The Back-end Middleware

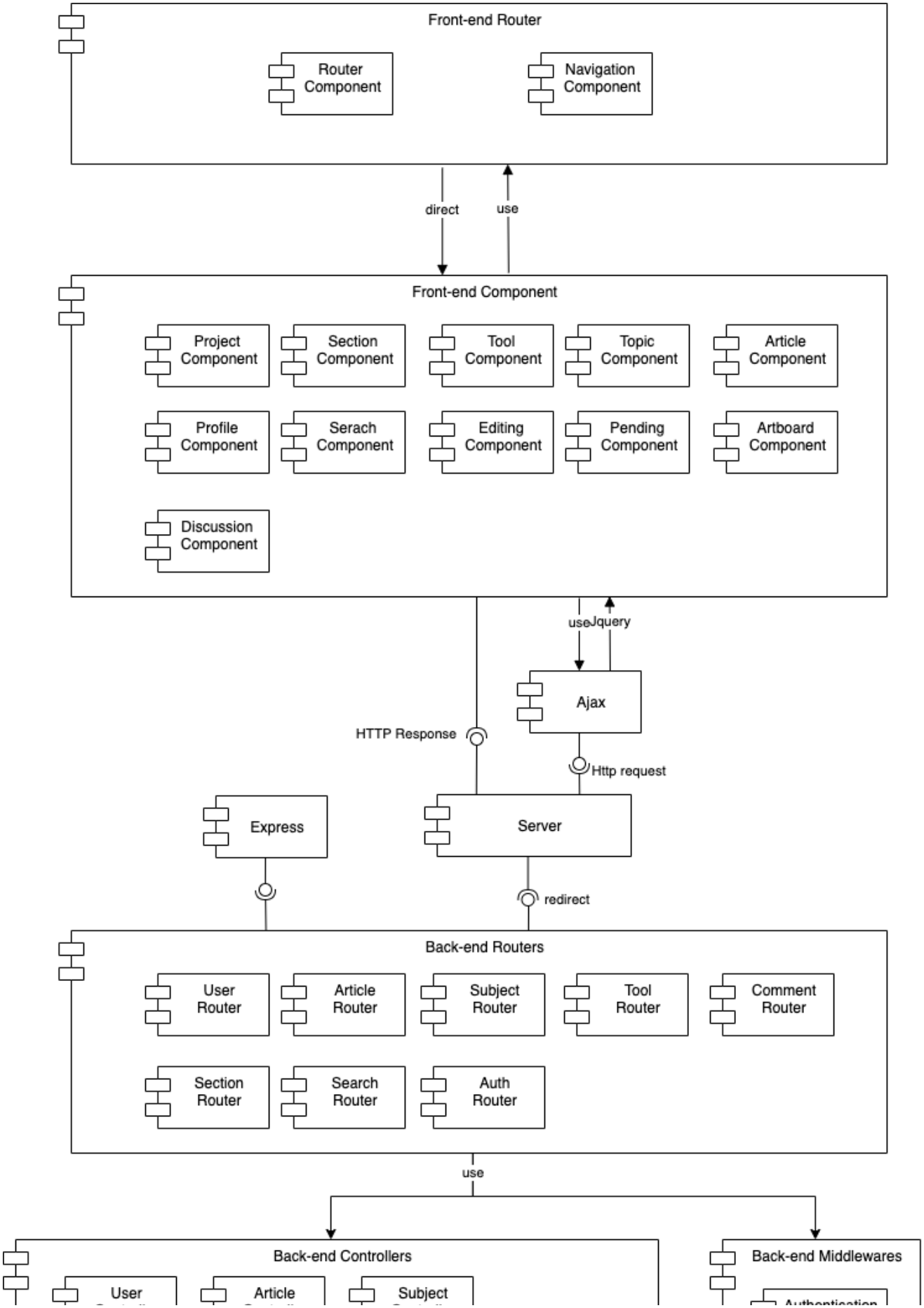
The Back-end middlewares are used specifically to provide user authentication and verification in the backend. Each incoming requests will be examined by the functions provided in the backend middleware to guarantee the website security.

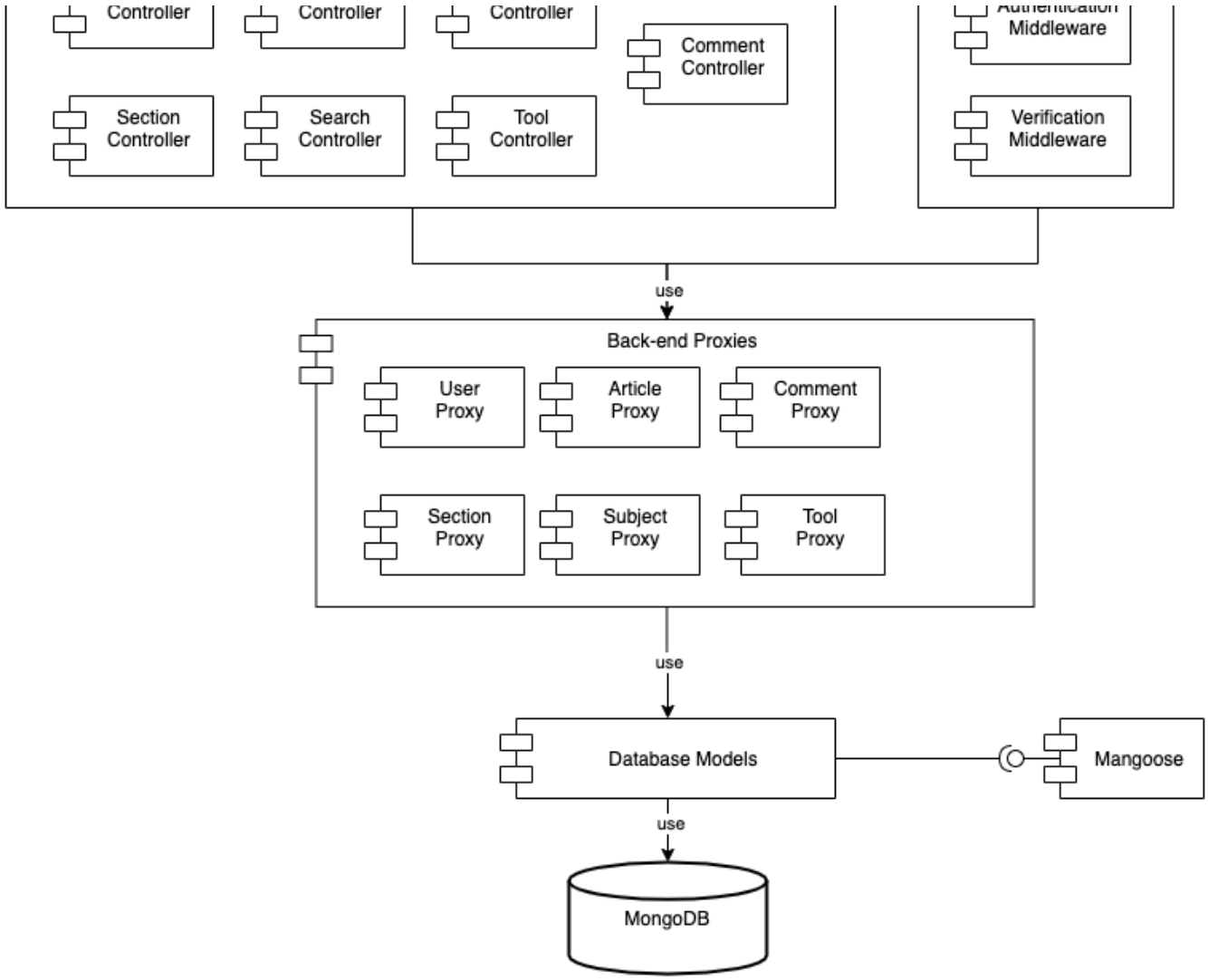
The Back-end Proxy

The Back-end proxies help the controllers to communicate with database models. They are sets of functions/methods to provide basic database IO service for the controllers to perform their functions.

External Components

Beyond these internal components, we mainly adopted Ajax and Express as an external framework to facilitate the front-end components sending HTTP requests, and help the server sending responses respectively. And we also used the Mongoose framework to in the system to help deal with database schemas as object-oriented models.


Front-end Router
Router Component
Navigation Component
direct
use
Front-end Component
Project Component
Section Component
Tool Component
Topic Component
Article Component
Profile Component
Serach Component
Editing Component
Pending Component
Artboard Component
Discussion Component
useJquery
Ajax
HTTP Response
Http request
Express
Server
redirect
Back-end Routers
User Router
Article Router
Subject Router
Tool Router
Comment Router
Section Router
Search Router
Auth Router
use
Back-end Controllers
User
Article
Subject
Back-end Middlewares



link:https://drive.google.com/file/d/1YdoBjteQwrmcFpqA2MAGQJd9JjGjdvGj/view?usp=sharing

Quality

For this section, Code Review policy and Git policy are constructed. Everyone should follow these 2 policies for quality assurance. These 2 policies can be found here:

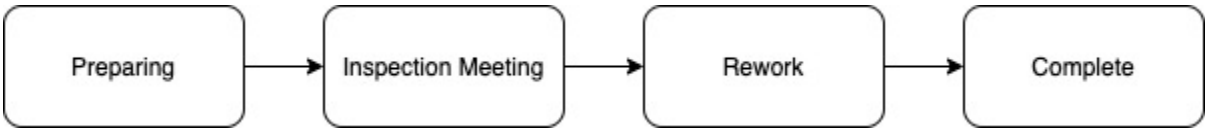
- Code Review Policy
- Git Policy

Code Review Policy

The Code Review policy refers to the previous project since it is the same. The original work can be found here: Code Review Policy.

How many reviewers per review?	A least 2 reviewers
Who should be a reviewer?	Can be members in the sub-team • Front-End code should be reviewed by Front-End member
How large of the code and how often?	• 400 LOC
How long should it take for a review?	• Less than one hour (60 min)
Use code review checklist	Code Review Checklist

Code Review Process



Preparing

- Developer(Auther) find the reviewers by SLACK
- Creating the pull request to the main sub-team branch
- After getting the reviewers then sending the invitation to them in a Pull request

Inspection Meeting

- Reviewer make an inspection on Github and comment it
- If there is any major issue, the reviewer must have a meeting with the code's owner.
- During the meeting should not take too long(<=1hour)

Rework

- After the reviewer gives feedback to the code's owner, they must fix it ASAP.

Complete

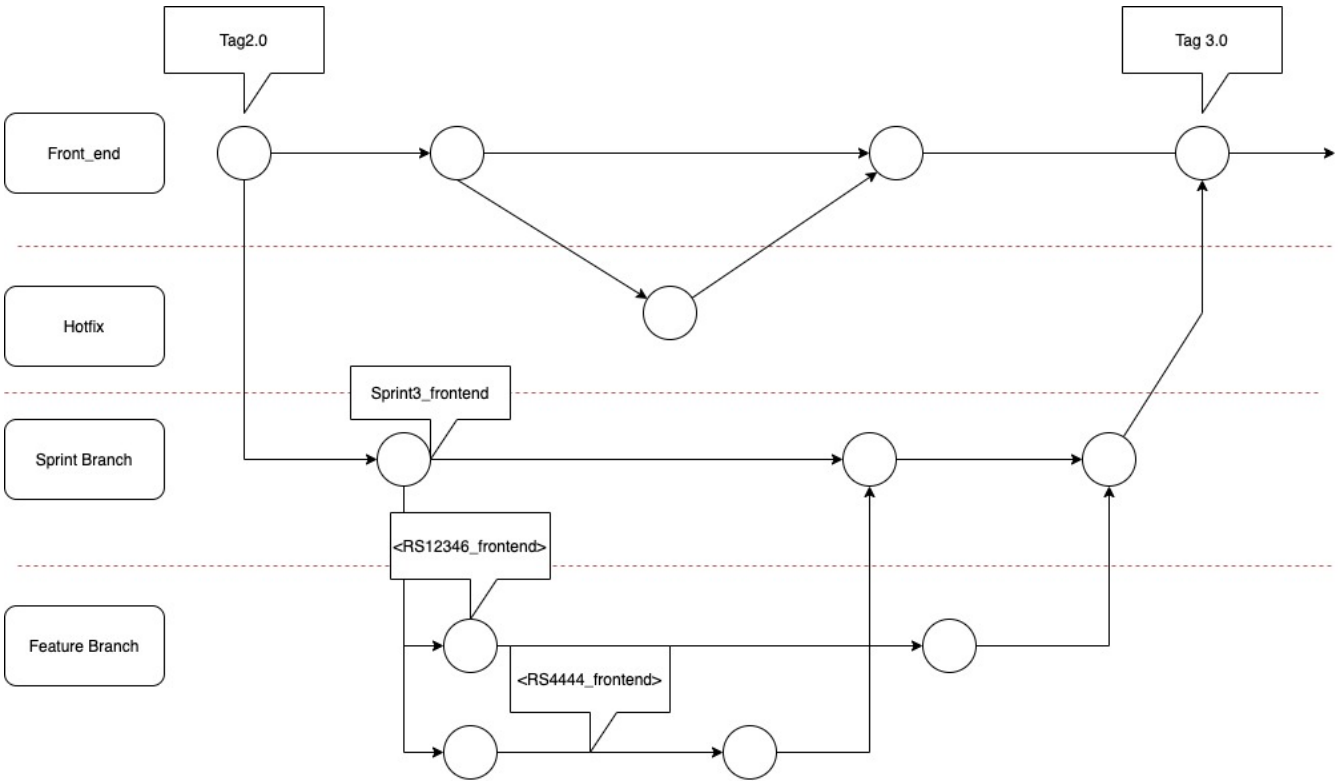
- Merge the code into the Main branch.

Git Policy

The Git policy refers to the previous project since it is the same. The original work can be found here: Git Policy.

Git Branch Strategy

Instance	Branch	Description, Instructions, Notes
Stable	master	
frontend	front_end	Branch for frontend
backend	back_end	Branch for backend
Sprint	sprint-*	This is a developing branch for each sprint. We can use sprint to do integration stuff and clean up before merge to (frontend or backend)
Features /Issues	topic-*	Always branch off HEAD of Working
hotfix	hotfix-*	Always branch off Stable



Feature Branches name

Name: <JiraTaskCode_(Backend/Frontend)>

For example

- Jira-Code: RS90013-104

- Backend RS90013-104_Backend
- Frontend RS90013-104_Frontend