1. Home . . . 3
1.1 Product Scope . . . 7
1.2 Team Roles . . . 8
1.3 Design . . . 9
1.3.1 Design Ideas . . . 10
1.3.2 Digital Prototype . . . 13
1.3.3 Mood Board . . . 14
1.3.4 User Journey . . . 16
1.3.5 WireFrame . . . 17
1.4 Requirements . . . 18
1.4.1 Goal Model . . . 19
1.4.2 User Stories . . . 21
1.4.3 Personas . . . 23
1.4.4 Acceptance Criteria . . . 28
1.5 Architecture . . . 38
1.5.1 Component Diagram . . . 39
1.5.2 Activity Diagram . . . 43
1.5.3 Front-end . . . 44
1.5.3.1 Front-end Class Design . . . 45
1.5.3.2 Research Front-end Patterns . . . 46
1.5.4 Back-end . . . 48
1.5.4.1 Back-end Design . . . 49
1.5.4.2 Database Design . . . 53
1.5.4.3 Sequence Diagram . . . 54
1.5.4.4 Communication Diagrams . . . 57
1.5.5 Deployment . . . 59
1.5.5.1 Deployment Plan . . . 60
1.5.6 High-level Architecture Diagram . . . 61
1.6 Quality Assurance . . . 63
1.6.1 Code Review Checklist . . . 64
1.6.2 Code Review Policy . . . 66
1.6.3 Git Policy . . . 67
1.7 Resources . . . 68
1.7.1 API authentication . . . 69
1.7.2 API for article . . . 70
1.7.3 API for Comment . . . 76
1.7.4 API for homepage . . . 78
1.7.5 Api for images . . . 79
1.7.6 API for Section . . . 80
1.7.7 API for Subject . . . 82
1.7.8 API for User Profile . . . 85
1.7.9 Use Cases . . . 88
1.8 Retrospectives . . . 94
1.8.1 2020-06-06 Retrospective . . . 95
1.8.2 2020-05-11 Retrospective . . . 96
1.8.3 2020-08-31 Retrospective . . . 97
1.8.4 2020-10-05 Retrospective . . . 98
1.9 Meetings . . . 99
1.9.1 Client meeting . . . 100
1.9.1.1 2020-03-13 Meeting notes . . . 101
1.9.1.2 2020-03-30 Meeting notes . . . 104
1.9.1.3 2020-06-01 Meeting notes . . . 106
1.9.1.4 2020-09-01 Meeting notes . . . 107
1.9.2 Supervisor meeting . . . 108
1.9.2.1 2020-03-10 Meeting notes . . . 109
1.9.2.2 2020-03-17 Meeting notes . . . 110
1.9.2.3 2020-03-24 Meeting notes . . . 112
1.9.2.4 2020-03-31 Meeting notes . . . 114
1.9.2.5 2020-04-07 Meeting notes . . . 115
1.9.2.6 2020-04-28 Meeting notes . . . 117
1.9.2.7 2020-08-07 Meeting notes . . . 118
1.9.2.8 2020-08-14 Meeting notes . . . 119
1.9.2.9 2020-09-11 Meeting notes . . . 120
1.9.2.10 2020-09-25 Meeting notes . . . 121
1.9.3 Team meeting . . . 122
1.9.3.1 2020-03-21 Meeting notes . . . 123
1.9.3.2 2020-03-28 Meeting notes . . . 126
1.9.3.3 2020-04-04 Meeting notes . . . 128
1.9.3.4 2020-04-11 Meeting notes . . . 130
1.9.3.5 2020-05-02 Meeting notes . . . 131
1.9.3.6 2020-05-18 Meeting notes . . . 133
1.9.3.7 2020-06-02 Meeting notes . . . 134
1.9.3.8 2020-08-09 Meeting notes . . . 135
1.9.3.9 2020-08-15 Meeting notes . . . 136
1.9.3.10 2020-08-31 Meeting notes . . . 137
1.9.3.11 2020-09-09 Meeting notes . . . 138
1.9.3.12 2020-09-16 Meeting notes . . . 139
1.9.3.13 2020-10-05 Meeting notes . . . 140
1.9.3.14 2020-10-12 Meeting notes . . . 141

1.10 Testing 142
1.10.1 API Testing 143

Home

Project Name: Software Training Repository

Project Description

The objective of this project is to build 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. Students need access to videos and tutorials that explain new tools and languages clearly. An example is a tutorial on React or Docker.

To help you get started, here are some important links for this page:

- Team roles can be found here
- To read an overview of this project, please see the Product Scope
- To view the project design ideas, mood board and wire frame, please see the Design
- For the Goal Model, User Stories, Personas and Acceptance Criteria, visit the Requirements
- Our architecture, research and design patterns are found here

Project Repository

Link to: https://github.com/matt01671/SWEN90013_RS

Client

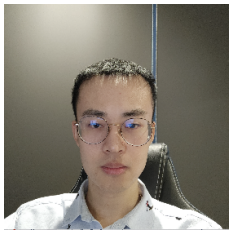
Name	E-mail
Professor Leon Sterling	leonss@unimelb.edu.au

Supervisor

Name	E-mail
Eduardo Oliveira	eduardo.oliveira@unimelb.edu.au

Team Detail

Student ID	Photo	Full Name	Preferred Name	E-mail	Contact Number	Role
942953		Patawat Watakul		pwatakul@student.unimelb.edu.au	0406 247 727	

966255		Yiyang JIN	Joyce	yiyangj@student.unimelb.edu.au	0456734335	
702322		Matthew O'Halloran	Matt	mohalloran@student.unimelb.edu.au	0467399385	
795989		Grant Boon Hwa Lim	Grant	limg2@student.unimelb.edu.au	0402489331	
833092		Gabriel Cheung	Gabriel	cheungg@student.unimelb.edu.au	0433724768	
1011946		Xujia Zhang	Xujia	xujzhang@student.unimelb.edu.au	0411357589	
1066830		Kaung Khant Thu	Kaung Khant	kthu@student.unimelb.edu.au	0414084419	

860072		Yi Wang	Yi	ywang20@student.unimelb.edu.au	0452613526	
962111		Jinxin TIAN	TIAN	jinxint@student.unimelb.edu.au	0411964395	
1035899		Congran LI	Ryan	congranl@student.unimelb.edu.au	0468699906	

Team Roles

Name	Sprint 0 Role
Patawat Watakul	Frontend, Quality lead
Yiyang JIN	Backend
Matthew O'Halloran	Scrum Masterm,Devops
Grant Boon Hwa Lim	Front end
Gabriel Cheung	Backend
Xujia Zhang	Backend
Kaung Khant Thu	Stand in Scrum master , Devops
Yi Wang	Backend
Jinxin TIAN	Frontend
Congran LI	Frontend

Recent space activity

Kaung Khant Thu

Acceptance Criteria updated 04 Nov, 2020 • view change

Yiyang JIN

Acceptance Criteria updated 04 Nov, 2020 • view change

Space contributors

- Kaung Khant Thu (4 days ago)
- Yiyang JIN (4 days ago)
- Yi Wang (14 days ago)
- Xujia Zhang (28 days ago)
- Patawat Watakul (51 days ago)
- ...

Kaung Khant Thu

API Testing updated 04 Nov, 2020 • view change

Yiyang JIN

2020-04-11 Meeting notes updated 02 Nov, 2020 • view change

2020-04-04 Meeting notes updated 02 Nov, 2020 • view change

Product Scope

Training resources for software engineering projects

Product Abstract

With the rapid evolution of software technology, including frameworks, testing and deployment tools, there is a need for a repository of high quality software engineering resources. Students need access to videos and tutorials that they can use to self-learn. The resources are be lightly controlled to ensure inappropriate material does not appear, be easy to search, and allow for comments on the effectiveness of the learning materials.

Product Description

The product is a wiki of learning resources for Melbourne University Engineering students to self-learn about important emerging technologies such as programming frameworks, testing tools and deployment tools. Students need access to videos and tutorials that explain new tools and languages clearly. An example is a tutorial on React or Docker. The wiki encourages two-way communication through active contribution to a subjects page.

Product License

The product is run using the GPL FLOSS license.

Team Roles

Scrum Master

- Kaung Khant

Front-end

- Congran Li
- Grant Lim
- Jinxin Tian
- Patawat Watakul (team lead)
- Gabriel Cheung

Back-end

- Yiyang Jin
- Xujia Zhang
- Yi Wang (team lead)
- Matthew O'Halloran

DevOps

- Kaung Khant (team lead)
- Matthew O'Halloran

Quality Lead

- Matthew O'Halloran

Design

Design Ideas

Page Description:	A list of websites used for inspiration, their associated features and comments
Page Status:	YELLOW

Digital 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

Mood Board

Page Description:	Mood board captures the theme to be used throughout the product design
Page Status:	COMPLETE

User Journey

https://uxpressia.com/w/C5Fre/p/xYOUf/m/DR9iX

WireFrame

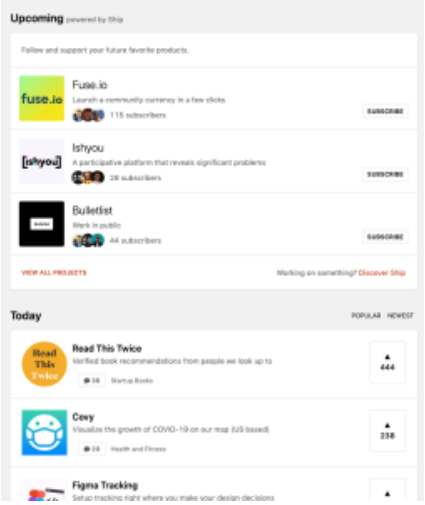
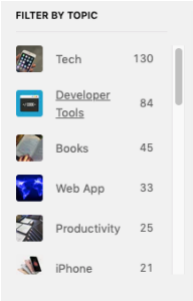
Page Description:	The wireframe is a digital prototype used to concept the design and specification of the product, without full development
Page Status:	COMPLETE

Design Ideas

Page Description:	A list of websites used for inspiration, their associated features and comments
Page Status:	YELLOW

Product Hunt

Website: https://www.producthunt.com/

Features	Photo	Description
Upvote		Upvote allows the platform to crowd source the 'best' resource
Main Page		Simple landing page leads the user directly to the most popular and up and coming apps of the day
Topic Filter		Ability to filter search results by topic

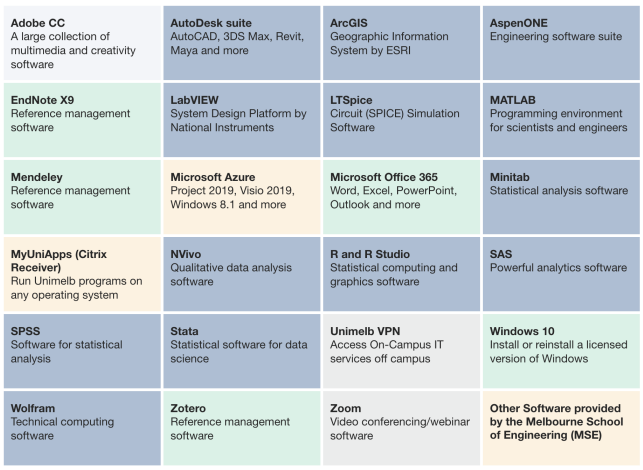
StackShare

Website: https://stackshare.io/

Features	Photo	Description
Current role of contributor	Robert Zuber CTO at CircleCI · Jul 24, 2019	Shows the current of the user that posted so that you can have confidence in their knowledge of the resource
Simple post with relevant tags for each tool	Shared insights on Datadog PagerDuty Honeycomb Rollbar Segment Amplitude PostgreSQL Looker at CircleCI Our primary source of monitoring and alerting is Datadog. We've got prebuilt dashboards for every scenario and integration with PagerDuty to manage routing any alerts. We've definitely scaled past the point where managing dashboards is easy, but we haven't had time to invest in using features like Anomaly Detection. We've started using Honeycomb for some targeted debugging of complex production issues and we are liking what we've seen. We capture any unhandled exceptions with Rollbar and, if we realize one will keep happening, we quickly convert the metrics to point back to Datadog, to keep Rollbar as clean as possible. We use Segment to consolidate all of our trackers, the most important of which goes to Amplitude to analyze user patterns. However, if we need a more consolidated view, we push all of our data to our own data warehouse running PostgreSQL; this is available for analytics and dashboard creation through Looker.	A short post consists of `tags` of each mentioned tool so that it is easy to refer to the what is being mentioned

Software for Students

Website: https://studentit.unimelb.edu.au/software

Features	Photo	Description
Tiled display of software		Simple tile display of applications
Coloured by software type		Software is coloured to reflect the type of software
Unimelb sponsored tools	N/A	Only displays tools that have a student subscription through UniMelb

Digital 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

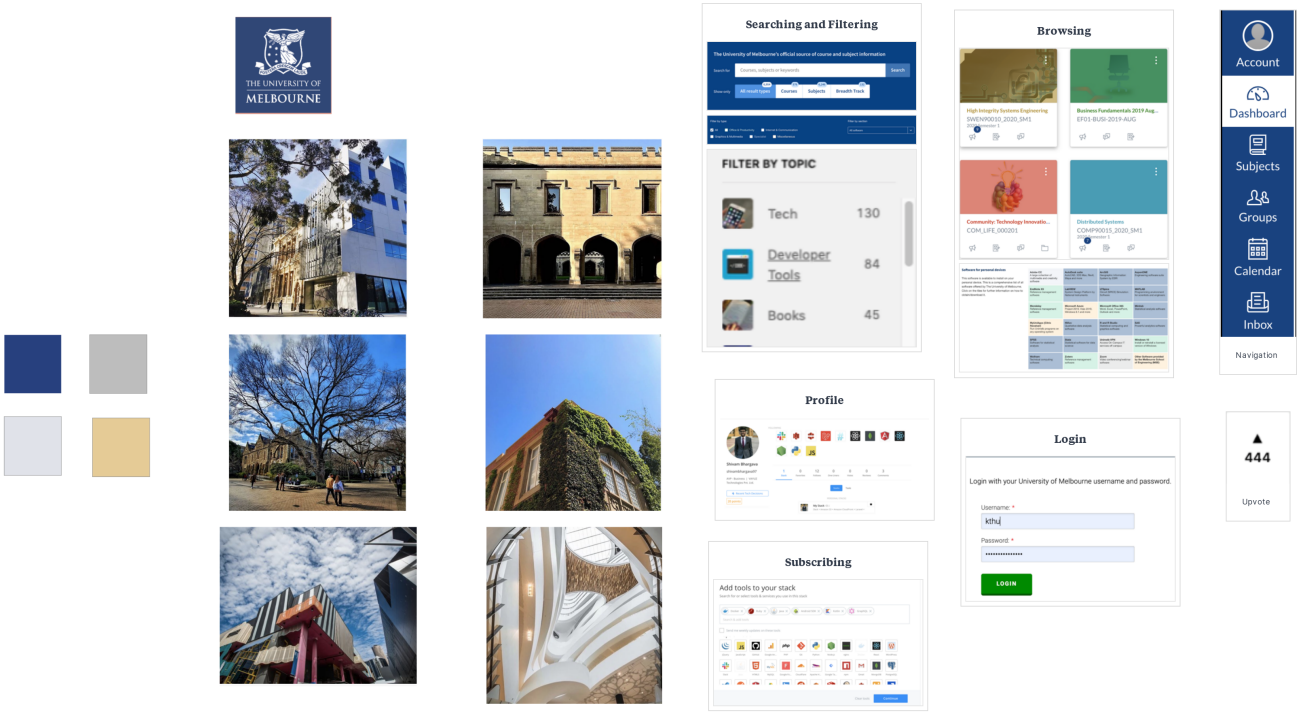
Digital Prototype Link

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

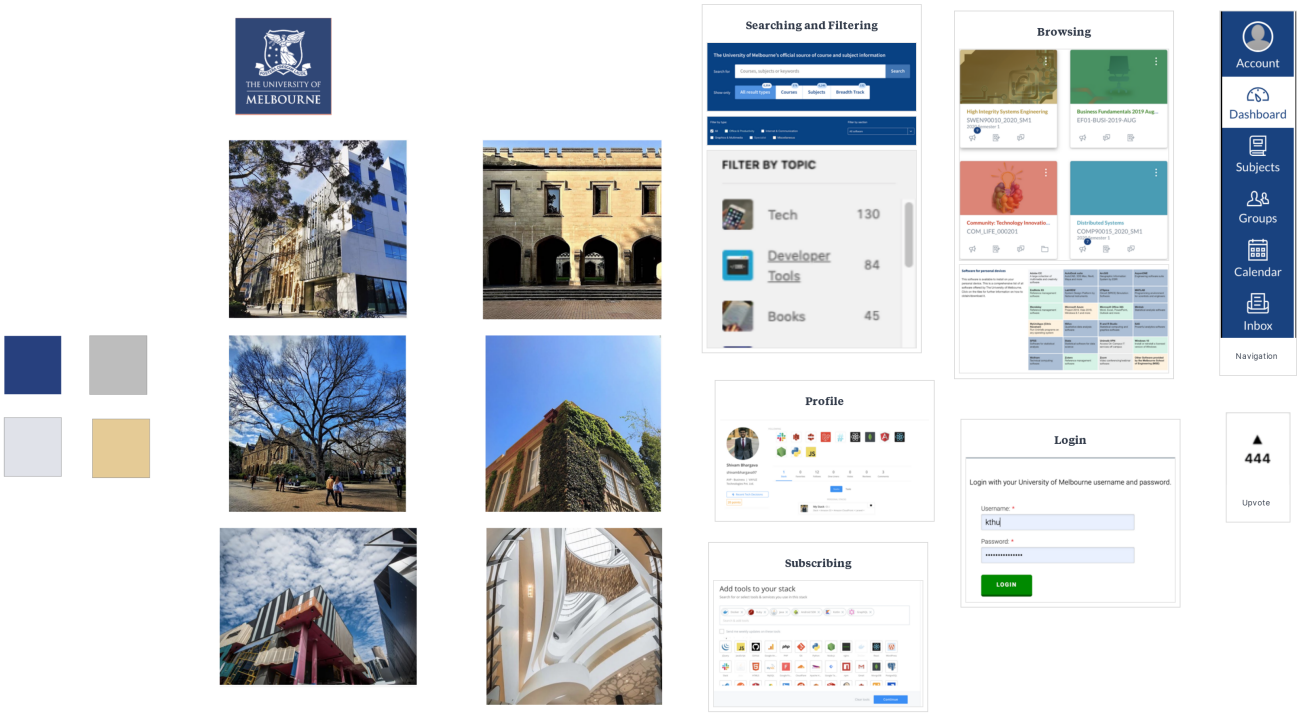
	Axure Link	Information
Student Version	https://zv8nzm.axshare.com	**Student Account** • Username: miranda@student.unimelb.edu.au • Password: 12345678
Moderator Version	https://bwh6ha.axshare.com	**Moderator Account** • Username: admin@unimelb.edu.au • Password: 12345678

Mood Board

Page Description:	Mood board captures the theme to be used throughout the product design
Page Status:	COMPLETE



Direct link to Mood Board https://app.milanote.com/1Jl1Wv10Z64RcM



Direct link to Mood Board https://app.milanote.com/1Jl1Wv10Z64RcM

User Journey

https://uxpressia.com/w/C5Fre/p/xYOUf/m/DR9iX

WireFrame

Page Description:	The wireframe is a digital prototype used to concept the design and specification of the product, without full development
Page Status:	COMPLETE

Link to our current WireFrame
https://marvelapp.com/achj5gg

Requirements

Goal Model

Page Description:	The Goal Model follows the Do/Be/Feel approach to specifying the scope and design of the project
Page Status:	PROGRESS

User Stories

Page Description:	User stories specify the low level tasks to be completed through development
Page Status:	PROGRESS

Personas

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

Acceptance Criteria

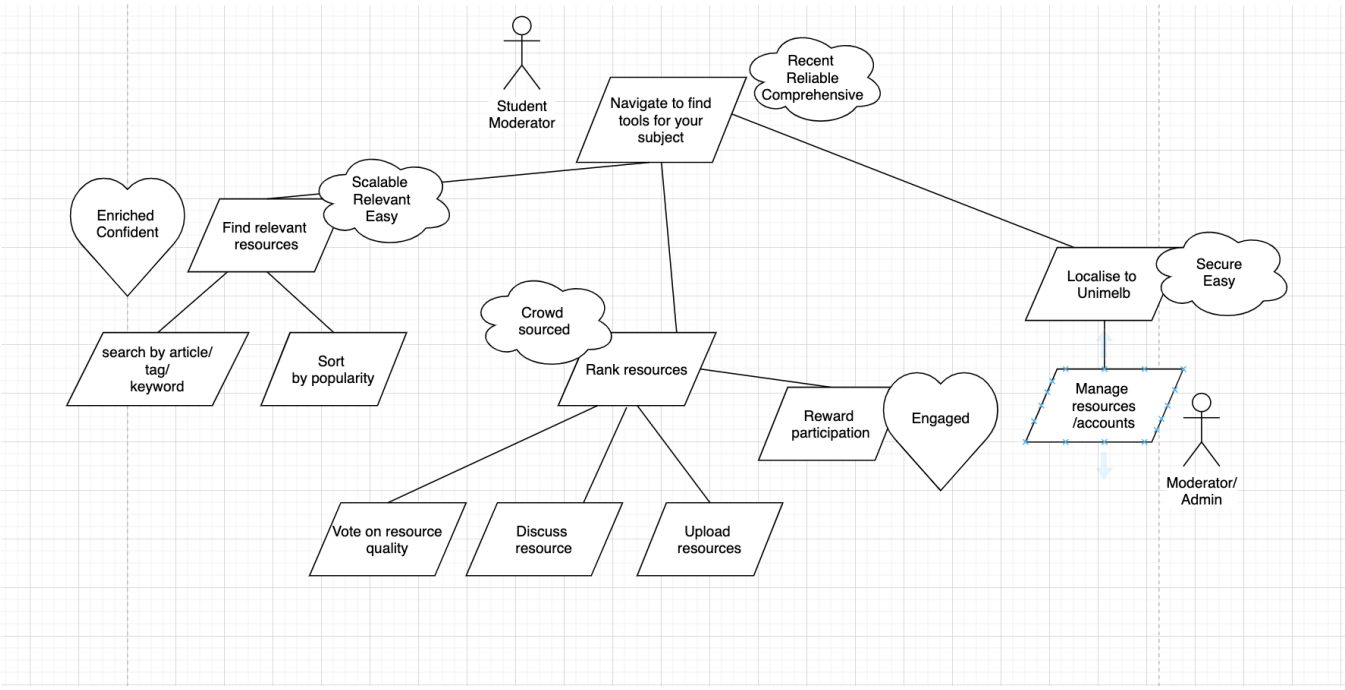
Page Description:	List the acceptance criteria for each corresponding user story
Page Status:	DONE

Goal Model

Page Description:	The Goal Model follows the Do/Be/Feel approach to specifying the scope and design of the project
Page Status:	PROGRESS

Goal Model

Link: https://drive.google.com/file/d/108Y5PuJ2tcIf1SDxgYo2qmuR8iQRIupk/view?usp=sharing



Goal List

Who	Do	Be	Feel
• Students • Moderator • Admin	• Discover new ways to learn • Find relevant resources • Search articles by tags and keywords • Sort by popularity • Rank resources • Vote on resource quality • Discuss resource • Upload resources • Reward participation • Localise to Unimelb • Manage resources/accounts	• Scalable • Relevant • Easy • Crowd sourced • Secure • Easy	• Enriched • Confident • Engaged

Ideas:

ideas about different roles:

- moderator: manage the content of the website: e.g. delete resources, ban users, etc.
- administrator: have full control of the entire website. assign roles

ideas about the leaderboard:

- a rank system for users (according to number of resource posted, rankings of their resources etc.)
    - other users can reach the user directly from the leaderboard
- a ranking system of good resources(e.g. by ranking in different topics)
    - a link of each resource so that we can click into the resource directly

ideas about the website:

- profile
  - show uploaded resources
  - show pending resources
- Main feed and subscription?
  - subscribing tutors ,lecturer, subjects , topics
    - Users mainly(lecturer) can limit who can view which content(because some contents might be related to leacture material)
- follow a user

Uploading function

- embedding resources (referencing)
- direct upload
- resources can be tag with specific subject. or tools when uploading(for example when upload react tutorial video the resource can be tag with react,web development and so on)

User Stories

Page Description:	User stories specify the low level tasks to be completed through development
Page Status:	PROGRESS

Our user stories follow the format: as <user role> I want to <goal> so that <benefit>. Each user story has corresponding acceptance criteria, and an acceptance test

They follow the INVEST criteria:

- **I**ndependent
- **N**egotiable
- **V**aluable
- **S**ized appropriately
- **T**estable

Story points are estimated on the Fibonacci scale: 1,2,3,5,8,13.17

MEDIUM = User story and respective criteria/tests should be satisfied upon development unless there is a clearly identified reason to extend to a later iteration

HIGH = User story and respective criteria/tests should be satisfied upon development.

LOW = User story and respective criteria/tests do not necessarily need to be satisfied upon development.

Epic	ID	As a	I want to	So that	Priority	Story Points	Goal Model
1. Login	1.0	student, moderator	I want to login using my unimelb account so that I can synchronize my enrolled subjects to the website (as we cannot get the unimelb credentials, we instead implemented the user story ID 1.1 - 1.3)		HIGH	13	Localise to Unimelb
	1.1	student, moderator	register an account on the website	I can log in to the website and use it.	HIGH	8	
	1.2	student, moderator, administrator	login to my account	I can use this website to learn more knowledge / manage my subjects/ manage the website.	HIGH	8	
	1.3	administrator	manage user account	I can help with the account issue when it is necessary.	HIGH	5	Manage resources /accounts
2. Subject	2.1	moderator	create a new subject	I can teach students with related learning resources.	HIGH	8	Find relevant resources
	2.2	moderator	add a new section into a subject	I can organize tools together with the subject.	MEDIUM	13	Find relevant resources
	2.3	moderator	delete a section	I can remove the unwanted section that I created.	MEDIUM	5	Find relevant resources
	2.4	moderator	review all the article requests related to a subject suggested by the students	I can check the validity of the resource.	HIGH	8	Find relevant resources
	2.5	moderator, student	view a subject	I can check what are the articles related to the subject	HIGH	13	Find relevant resources
	2.6	moderator	add a new tool into the section	I can suggest tools that are related to the subject.	MEDIUM	5	Find relevant resources
3. Article	3.1	student, moderator	create a new article	I can share my understanding with other people.	HIGH	13	Upload resources
	3.2	student, moderator	edit an existing article	I can correct the errors in the existing articles and add an extension to them.	HIGH	13	Upload resources
	3.3	student, moderator	add a tag to an article	I can link an article to a particular topic.	MEDIUM	5	Upload resources
	3.4	moderator	make sure that all articles created/edited by students become pending articles, and will be reviewed by moderators	the quality of articles on the website can be guaranteed.	HIGH	13	Upload resources

	3.5	moderator	approve a pending article created/edited by a student	a high-quality article created/edited by a student can be published on the website.	HIGH	5	Upload resources
	3.6	moderator	reject a pending article created /edited by a student	a poor-quality article created/edited by a student cannot be published on the website.	HIGH	5	Upload resources
	3.7	student, moderator	search an article by keyword	I can find articles related to the topic I am interested in.	HIGH	8	Search articles by tags and keywords
	3.8	student, moderator	search an article by its tag	I can find articles based on their current tags.	MEDIUM	8	Search articles by tags and keywords
4. Comment	4.1	student, moderator	give comment on an article	I can give my opinion or ask a question if I am not clear about the content.	HIGH	8	Discuss resource
	4.2	student, moderator	delete my own comment	I can remove my mistakes that are related to the content.	HIGH	5	Discuss resource
	4.3	student, moderator	reply to other people comment	I can discuss my opinion with other people.	MEDIUM	13	Discuss resource
5.Homepage	5.1	student, moderator	have all the subjects enrolled /moderated by me listed on my homepage	I can easily access those subjects from my homepage.	HIGH	13	Discuss resource
6.Ranking	6.1	student	be listed and ranked in the leaderboard according to my contribution	I can be encouraged to post more valuable articles on the website.	LOW	13	Reward participation,
	6.2	student, moderator	add a like to an article	I can add popularity to high-quality articles.	LOW	8	Reward participation, Rank resources
	6.3	student, moderator	remove a like that I have given	I can remove my mistake of misclicking.	LOW	5	Reward participation
	6.4	student, moderator	sort the articles by the number of likes	I can find the most recognized article under a topic.	LOW	8	Reward participation, Sort by popularity
7.Tools	7.1	moderator	I would like to create a new tool	I can organize articles that are related to the tool in one place	MEDIUM	8	Discuss resource
	7.2	moderator	I would like to view the tool	I can see all the tutorials that are related to the tool	MEDIUM	8	Discuss resource, Manage resources /accounts
8. Profile	8.1	student, moderator	I would like to view my profile	I can check all the published and pending articles that I have submitted.	HIGH	8	Discuss resource, Manage resources /accounts

Personas

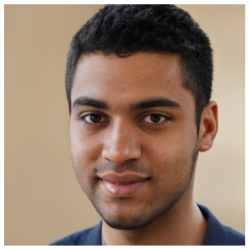
Page Description:	User profiles that define different users for the product, their goals, motivations and characteristics
Page Status:	DONE

- Student (Inexperienced/International)
- Student (Intermediate)
- Student (Experienced)
- Teaching Staff
- Moderator

Student (Inexperienced/International)

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

Aakash Dheer



"Study abroad is not easy and I need some tools to help"

Age: 25
Work: Master Student
Family: Single
Location: Melbourne, VIC
Character: Unconfident

Personality

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

Afraid | Curious | Lonely | Nice

Goals

- He wants to find a programming tutorial which uses simple English to explain a complicated tool.
- He wants to know the discussion on chosen tools from the previous Project.
- Need to learn a new programming framework which may use in the Software Project.

Frustrations

- Some tutorials on the internet are not written for international students who are not super strong in English.
- It is hard to find a resource which explains the tool for a particular project.

Bio

Aakash Dheer studies Master of Software Engineering in India and he does exchange program at the University of Melbourne for 1 year. When he begins to study at the university, he gets the various issues to study here. Firstly, even though, he can speak and write in English very well but he has to spend a lot of time to read some complex tutorial. Moreover, he doesn't have any friend in the university to give advice to him.

Motivation

Incentive
Fear
Growth
Power
Social

Student (Intermediate)

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

Joanna Patterson

"I need to learn new programming languages and frameworks for my assignment but I don't know where I can find the reliable resources"

Age: 24
Work: Student
Family: Single
Location: Melbourne, VIC
Character: Ambitious

Personality

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

Peacefulness | Confidence | Proactive | Self-control

Goals

- To find some relevant learning resources for self-practicing
- To learn python for using to complete AI's assignments
- To write a new tutorial for next year students.

Frustrations

- Lack of knowledge in python
- Have the motivation to learn but unsure where to begin
- Want to give a suggestion for next year students but don't know where to put it.

Bio

Joanna is a Master student in Software Engineering at the University of Melbourne. She has basic skills in some programming languages, such as Java, JavaScript and C. She is interested in Artificial Intelligence; therefore, in this semester, she has decided to enroll in the Artificial Intelligence subject which has an assignment that must be developed in Python. However, she has not learned python before, and would like to find the best resources to quickly learn what is necessary for the assignment.

Motivation

Incentive
Fear
Growth
Power
Social

Student (Experienced)

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

Miranda Shaffer

"I remember how hard it was to find good resources when i was starting out."

Age: 26
Work: Student
Family: Single.
Location: Melbourne, VIC
Character: Calm

Personality

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

Friendly | Helpful | Empathetic

Goals

- Wants to be able to share resources she feels were most useful
- Check their understanding of whatever is required for the subject and refresh their memory

Frustrations

- No single method of easily sharing resources with other students
- Other students cannot ask for help easily

Bio

Miranda is studying a Masters of Software Engineering at the University of Melbourne. She has significant experience in multiple programming languages and technologies. She had an interest in Artificial Intelligence; therefore, in this semester, she has decided to enroll in the Artificial Intelligence subject which has an assignment that must be developed in Python. She already has experience developing in Python, however she would like to be able to revisit the resources she used to learn them as it has been a while since she last used Python, as well as be able to share them with her classmates and friends who are struggling.

Motivation

Incentive
Fear
Growth
Power
Social

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

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

Sid Hanna

"I'd like to be sure that the resources I direct students to are of good quality, but I don't have time."

Age: 26
Work: Tutor&Moderator
Family: Single
Location: Melbourne, Vic
Character: Pragmatic

Personality

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

Methodical | Cautious | Proactive | Helpful

Goals

- To be able to easily curate the resources that students use
- To be able to see student participation and feedback, to know which resources they find most useful

Frustrations

- Students learning from different resources require different help
- Students may be using poor quality resources
- Resources can be outdated

Bio

Sid is currently studying for his PhD at the University of Melbourne, and is also a tutor for multiple subjects. As a result he has to handle the varying levels of experience that new students have with each language being used, and is currently frustrated at the difficulty in finding resources for each subject that the students can use, and would prefer to have a centralized resource that he can personally manage.

Motivation

Incentive
Fear
Growth
Power
Social

Acceptance Criteria

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

Goal Model		User Stories	Acc. Criteria ID	Acceptance Criteria
	1.1	As a student, I want to register an account on the website so that I can log in to the website and use it.	.1	Given that student wants to register an account, When the student fill in the userid , username and password. a new student account should be created and the account should be able to use for login
			.2	Given that student wants to register an account, When the student fill in the userid or username which already existed in the system. the system should display an error.
	1.2	As a student/moderator, I want to login to my account so that I can use this website to learn more knowledge/manage my subjects/ manage the website.	.1	Given that the student /moderator wants to log in to the account, When the student /moderator fill in the correct account and password on the Login page, The student /moderator's homepage will be shown on the website.

			.2	Given that the student /moderator wants to log in to the account, When the student /moderator fill in the wrong account and password on the Login page, The student /moderator will get an error message telling him/her the provided account information is wrong.
Manage resources /accounts	1.3	As an administrator I want to manage user account so that I can help with the account issue when it is necessary.	.1	Given that the administrator has already login to the system,(using username admin2 and password admin2) When the administrator click the "User Management" button on the home page, The administrator should be able to add new users into the system.
Find relevant resources	2.1	As a moderator, I want to create a new subject so that I can teach students with related learning resources.	.1	Given that moderator wants to create a new subject, When the moderator click "Add Subject" on his/her Homepage and provide a unique subject code and other relevant information, The subject will be successfully created.

			.2	Given that moderator wants to create a new subject, When the moderator click "Add Subject" on his/her Homepage and provide a duplicate subject code and other relevant information, The subject will not be successfully created.
Find relevant resources	2.2	As a moderator, I want to add a new section to a subject so that I can organize tools together with the subject	.1	Given that moderator has already login to the system and already selected the subject When the moderator click edit page layout in the subject page and continue by clicking "Add new section" moderator add a new section detail and publish A new section should be created for the subject.
			.2	Given that moderator has already login to the system and already selected the subject When the moderator click edit page layout in the subject page and continue by clicking "Add new section" moderator add a new section detail and cancel A new section should be created for the subject

Find relevant resources	2.3	As a moderator, I want to delete a section so that I can remove the unwanted section that I created.	.1	Given that the moderator wants to delete a section, When the moderator click "Delete" in the Edit Subject Page Layout page, and confirm the confirmation message in the pop-up window, The section will be successfully deleted.
			.2	Given that the moderator wants to delete a section, When the moderator click "Delete" in the Edit Subject Page Layout page, and cancel the confirmation message in the pop-up window, The section will not be successfully deleted.
Find relevant resources	2.4	As a moderator, I want to review all the article requests related to a subject suggested by the students so that I can check the validity of the resource.	.1	Given that the moderator wants to review an article for a subject, When the moderator click "Pending Submissions" on the subject page, The moderator should be able to view pending submissions submitted by students on the page
			.2	Given that the moderator wants to approve an article for the subject, When the moderator click approve for one of the pending submission, The moderator should be able to see the approved article on the subject page

			.3	Given that the moderator wants to delete a pending article for the subject, When the moderator click delete for one of the pending submission, The pending article should be deleted from the page.
Find relevant resources	2.5	As a student/moderator, I want to view a subject so that I can check what are the articles related to the subject.	.1	Given the student wants to view a subject, When the student clicks the subject on his/her Homepage, The student will be directed to the subject page and see all the related materials on the Subject page.
			.2	Given the moderator wants to view a subject, When the moderator clicks the subject on his /her Homepage, The moderator will be directed to the subject page and see all the related materials on the Subject page as well as other moderator-exclusive option.
Upload resources	3.1	As a student /moderator,I want to create a new article so that I can share my understanding with other people.	.1	Given that moderator want to create an article, When the moderator /student click "Add new article" and click "Publish" The article should be created in the subject page.

			.2	Given that student want to create an article, When the moderator /student click "Add new article" and click "Publish" The pending article should be created and moderator should be ready to review.
Upload resources	3.2	As a student/moderator, I want to edit an existing article so that I can share my understanding with other people.	.1	Given the student wants to edit an existing article, When the student clicks the "Edit Article" button on that article's page, makes a modification, and click Save, The article will be created as a pending article.
			.2	Given the moderator wants to edit an existing article, When the moderator clicks the "Edit Article" button on that article's page, makes a modification, and click Save, The article will be created and published directly.
			.3	Given the student /moderator wants to edit an existing article When the moderator clicks the "Edit Article" button on that article's page, makes a modification, but clicks Cancel, The article will be not created.

Upload resources	3.3	As a student/moderator,I want to add a tag when creating an article so that I can link an article to a particular topic	.1	Given that the moderator want to add a tag, When the moderator clicks the "Edit Article" button on that article's page, add or delete a new tag. the tag should be added to the article.
			.2	Given that student want to add a tag, When the student clicks the "Edit Article" button on that article's page, add or delete a new tag. The article should be added as a pending article for the moderator to review.
Upload resources	3.4	As a moderator, I want to make sure that all articles created/edited by students become pending articles, and will be reviewed by moderators so that the quality of articles on the website can be guaranteed.	.1	Given the moderator wants to make sure that all articles created /edited by students become pending articles, and will be reviewed by moderators. When the moderator clicks the "Pending Article Submission" button on the Subject page. The moderator will see a list of all the current pending articles on that subject.

Upload resources	3.5	As a moderator, I want to approve a pending article created/edited by a student so that a high-quality article created/edited by a student can be published on the website.	.1	Given that the moderator wants to approve an article for a subject, When the moderator click approve for one of the pending submission, The moderator should be able to see the approved article on the subject page
			.2	Given that the moderator wants to delete a pending article for te subject, When the moderator click delete for one of the pending submission, The pending article should be deleted from the page.
Upload resources	3.6	As a moderator, I want to reject a pending article created/edited by a student so that a poor-quality article created/edited by a student cannot be published on the website.	.1	Given the moderator wants to reject a pending article created /edited by a student. When the moderator clicks the "Reject" button of a pending article on the Pending Articles Submission page. The pending article will be rejected and will not be published.
Upload resources	3.7	As a moderator/student,I want to search an article by keyword so that I can find articles related to the topic I am interested in.	.1	Given that the moderator /student want to search for an article, When the moderator click search and type some content in the search bar, The related articles should be shown on the search list.

			.2	Given that the moderator /student want to search for an article, When the moderator click search and type some content in the search bar and choose the type of the content The related articles should be shown on the search list.
Upload resources	3.8	As a student/moderator, search an article by its tag so that I can find articles based on their current tags.	.1	GIven the student /moderator wants to search an article by its tag. When the student /moderator searches the tags in the search bar. The student /moderator can find all articles tagged with the target tag.
Discuss resource	4.1	As a student/moderator,I want to give comment on an article so that I can give my opinion or ask a question if I am not clear about the content.	.1	Given that the student /moderator want to give a comment on an article, When the student /moderator type comment and post, The comment should be shown on the article page
Discuss resource	4.2	As a student/moderator, I want to delete my own comment so that I can remove my mistakes that are related to the content.	.1	Given the student /moderator wants to delete his/her own comment, When the stude nt/moderator clicks the "Delete" button on his/her own comment. The comment will be successfully deleted.

Discuss resource	4.3	As a student/moderator, I want to reply to other people's comment so that I can discuss my opinion with other people.	.1	Given that the student /moderator want to reply to a comment on an article, When the student /moderator click "reply" and type content and click "Post" The comment should be shown on the article page
Discuss resource	5.1	As a student/moderator, I want to have all the subjects enrolled/moderated by me listed on my homepage so that I can easily access those subjects from my homepage.	.1	Given the student /moderator wants to have all the subjects enrolled /moderated by me listed on my homepage. When the student /moderator goes to his/her Hompage. All the subjects enrolled /moderated by the student /moderator will be listed there.
Discuss resource, Manage resources /accounts	8.1	As a student/moderator,I want to view my profile so that I can check all the published and pending articles that I have submitted.	.1	Given that the student /moderator want to view the profile, When the student /moderator click a profile on the right side of the navigation bar, The student /moderator should be able to view the published and pending articles that they did.

Architecture

Component Diagram

Internal Components

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

1. The Front-end Components,

Activity Diagram

The activity diagram is a model view used to describe system behavior. It can be used to describe the action and the result of the object state change caused by the action, without considering the event causing the state change. The following figure is the activity diagram of the RS system established based on the digital prototype and user story. Describes the work done during the execution of an operation.

Front-end

Front-end Class Design

As we use the react to develop our project, there are many components in the front-end.

Our project have 8 functions and each functions is shown by one main component. So the components can be simple as : Home, Project, Tool, Article, Search, Profile, Editing and Pending.

Back-end

Back-end Design

The backend design includes the backend domain design and the backend class design. In the back-end design, we treat data in an object-oriented way. And the relation between objects is showed in the back-end domain diagram. Also, we split our code into several layers: the routers, the controllers, the middleware, and the proxies. The details of their relations are shown in the back-end class diagram.

Deployment

Deployment Plan

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.

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

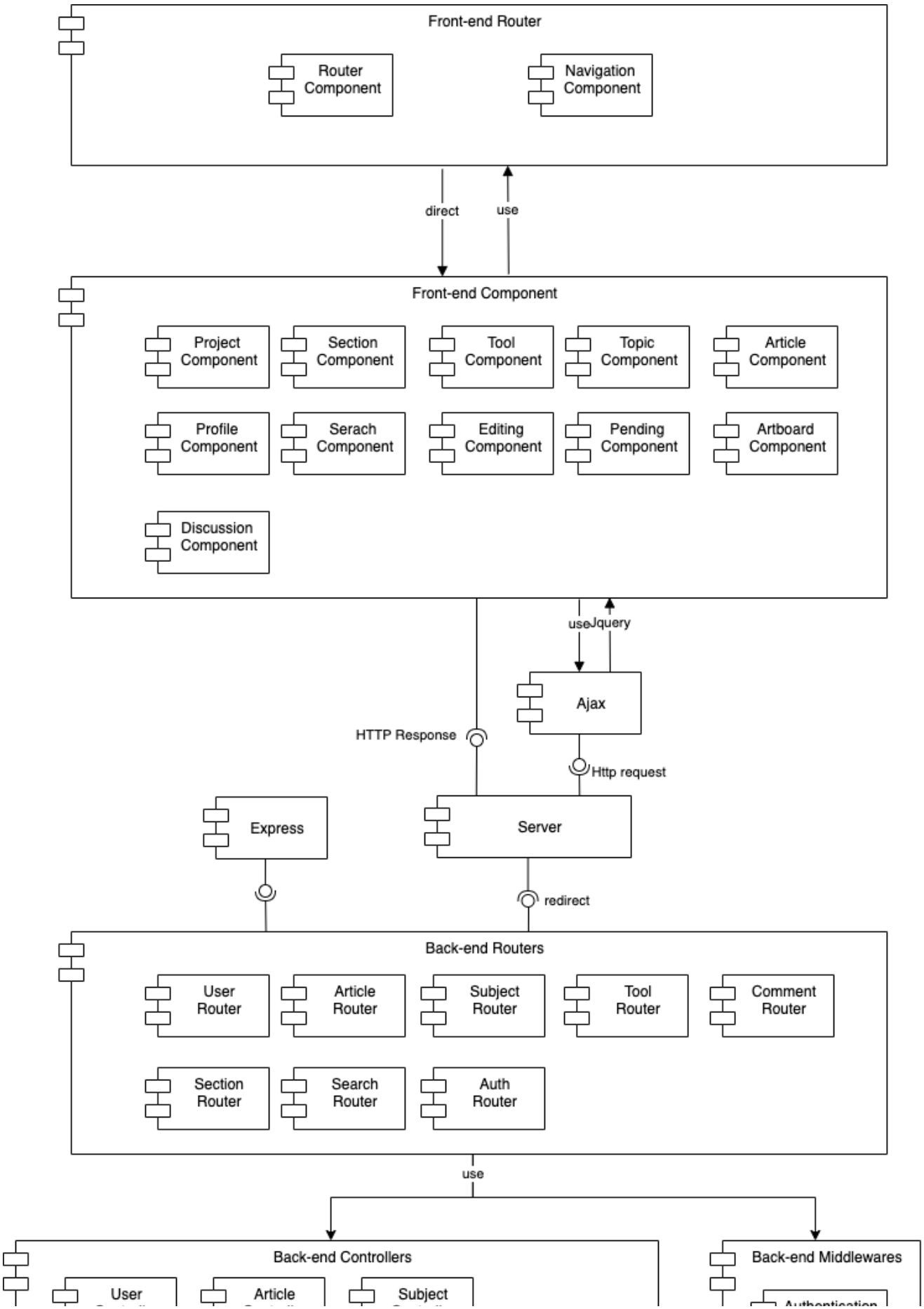
The Back-end middlewares are used specifically to provide user authentication and verification in the backend. Each incoming requests will be examined by the functions provided in the backend middleware to guarantee the website security.

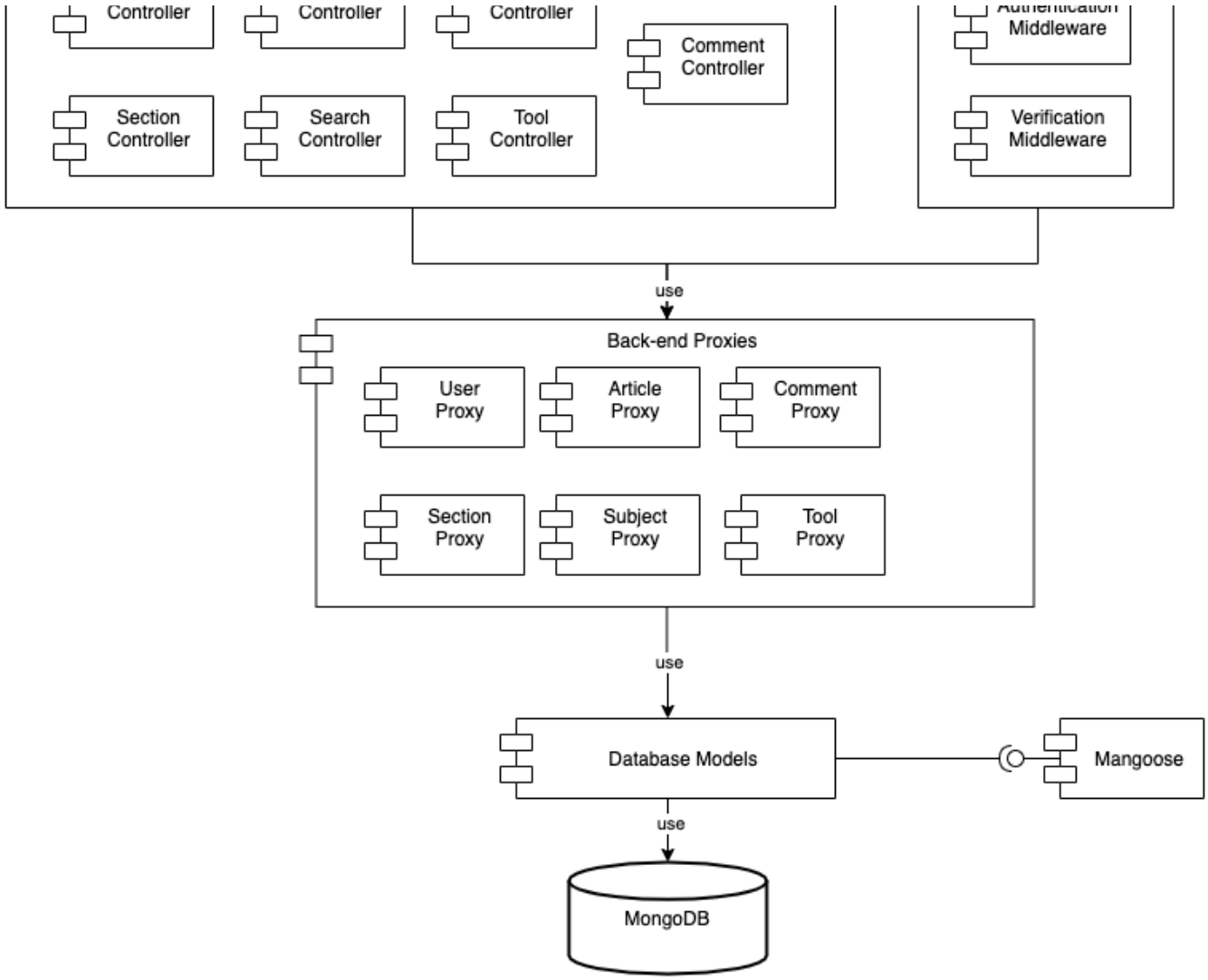
The Back-end Proxy

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.

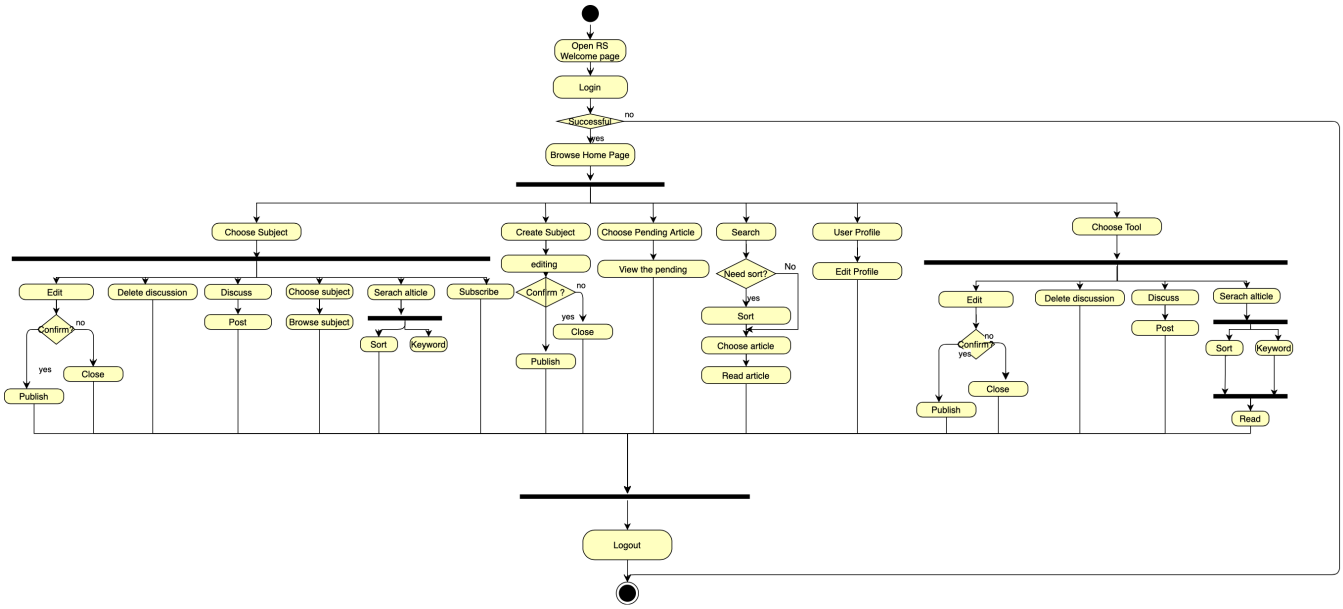

Front-end Router
Router Component
Navigation Component
direct
use
Front-end Component
Project Component
Section Component
Tool Component
Topic Component
Article Component
Profile Component
Serach Component
Editing Component
Pending Component
Artboard Component
Discussion Component
useJquery
Ajax
HTTP 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

Activity Diagram

The activity diagram is a model view used to describe system behavior. It can be used to describe the action and the result of the object state change caused by the action, without considering the event causing the state change. The following figure is the activity diagram of the RS system established based on the digital prototype and user story. Describes the work done during the execution of an operation.



Every user who wants to visit the RS website must open the welcome page firstly, then he must login with the correct username and password otherwise he cannot access the resource of the RS website. Once the user login successfully, he can choose what he wants to browse, including subjects and tools. The user can search what he wants to access by keyword in the search bar, and then he can sort the result according to some requirements. Besides, each user can edit his profile once he clicks the User Profile in the top-right of the webpage. The most important function of the RS website is to provide a useful resource to students and allow them to discuss freely. User with corresponding right can manage the discussion board once he accesses to the Subject Page or Tool Page, and he also can post a new discussion in this section. Additionally, a user can subscribe to a subject that he likes on the subject page. Each article will move to the pending section before they publish, and those articles also can be deleted before they published in the pending section. Moreover, a user with a legal right can create a new subject and edit it, then choose to publish or delete it.

Front-end

Front-end Class Design

As we use the react to develop our project, there are many components in the front-end.

Our project have 8 functions and each functions is shown by one main component. So the components can be simple as : Home, Project, Tool, Article, Search, Profile, Editing and Pending.

We use the route(index.jsx) to show our logical view and the Conceptual Class Diagram is as below:

Research Front-end Patterns

Compared with the MVC(Model View Controller)pattern and MVP(Model View Presentation)pattern, MVVM(Model View View-Model) can make the code loosely coupled, maintainable and easy to test.

In MVVM architecture patterns, DOM element is only interested in a certain part of the model (i.e. only part of the app state). By binding each element to its responsible part of the model (View-Model layer), it can directly listen to change of (read) / update (write) its part of the model. Using this pattern, we achieved separation of concerns — separating business logic (Model) from presentational layer (View). The following chart is the MVVM structure pattern.

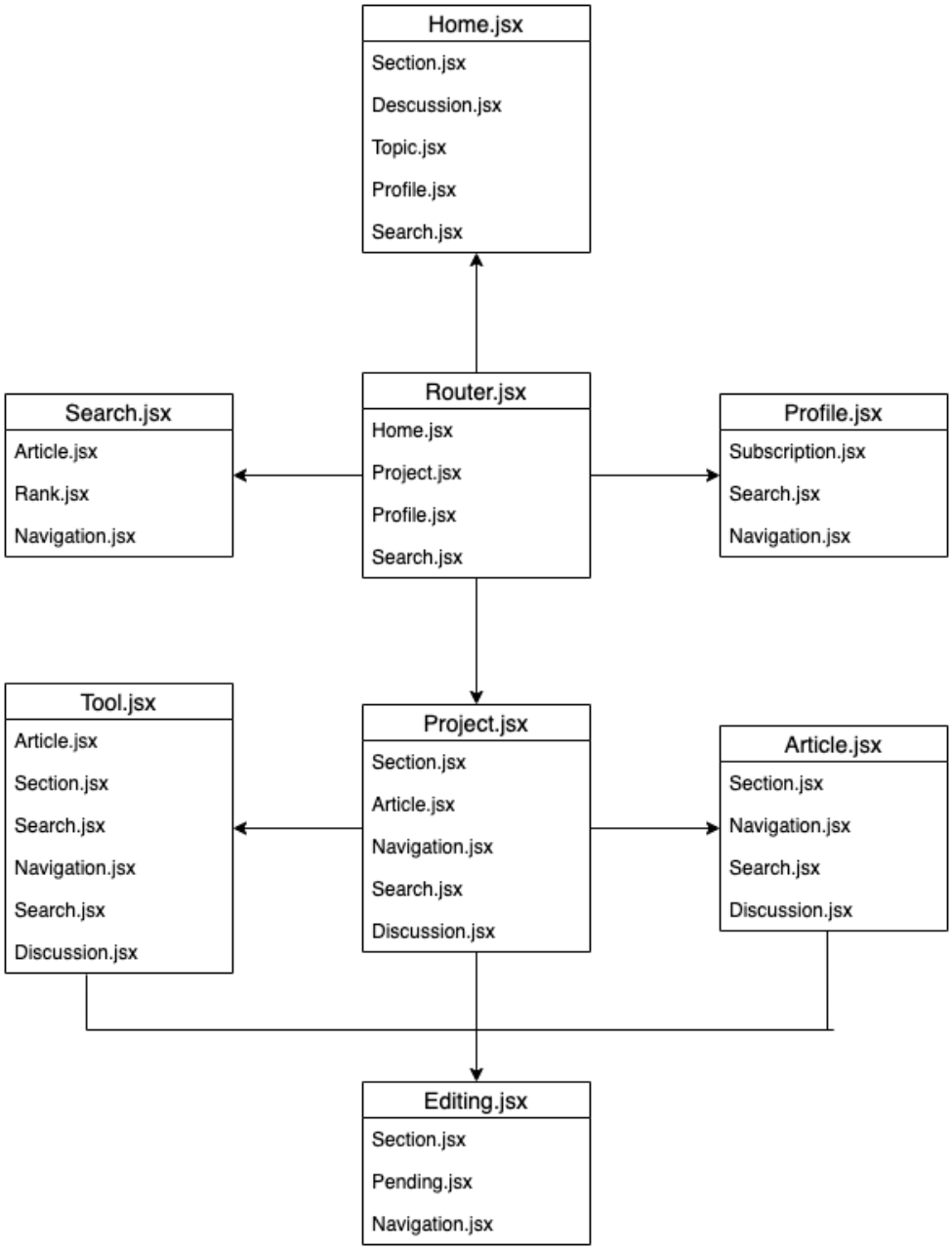
Front-end Class Design

As we use the react to develop our project, there are many components in the front-end.

Our project have 8 functions and each functions is shown by one main component. So the components can be simple as : Home, Project, Tool, Article, Search, Profile, Editing and Pending.

We use the route(index.jsx) to show our logical view and the Conceptual Class Diagram is as below:

Link to Edit : https://drive.google.com/file/d/1jX92joZnZ08N0rt1iuLK-zwwCfeiHxWq/view?usp=sharing

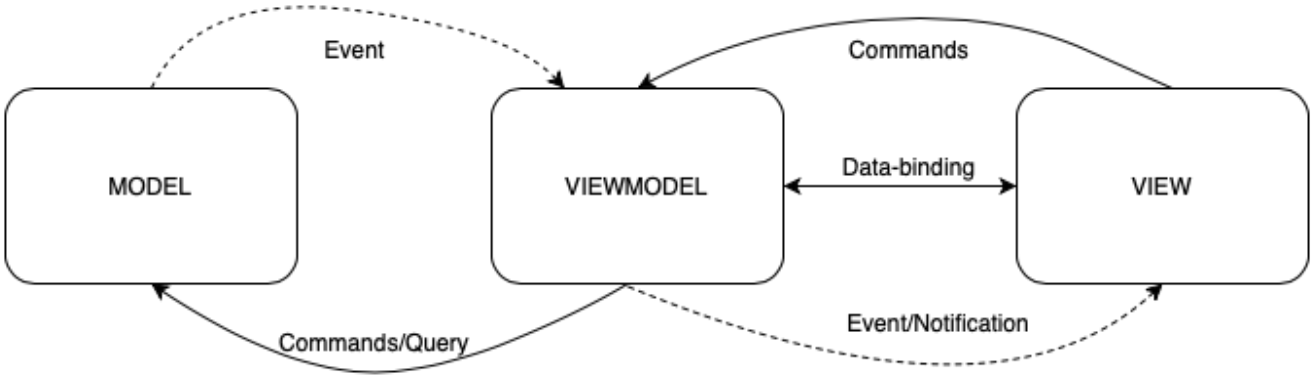


Research Front-end Patterns

Compared with the MVC(Model View Controller)pattern and MVP(Model View Presentation)pattern, MVVM(Model View View-Model) can make the code loosely coupled, maintainable and easy to test.

In MVVM architecture patterns, DOM element is only interested in a certain part of the model (i.e. only part of the app state). By binding each element to its responsible part of the model (View-Model layer), it can directly listen to change of (read) / update (write) its part of the model. Using this pattern, we achieved separation of concerns — separating business logic (Model) from presentational layer (View). The following chart is the MVVM structure pattern.

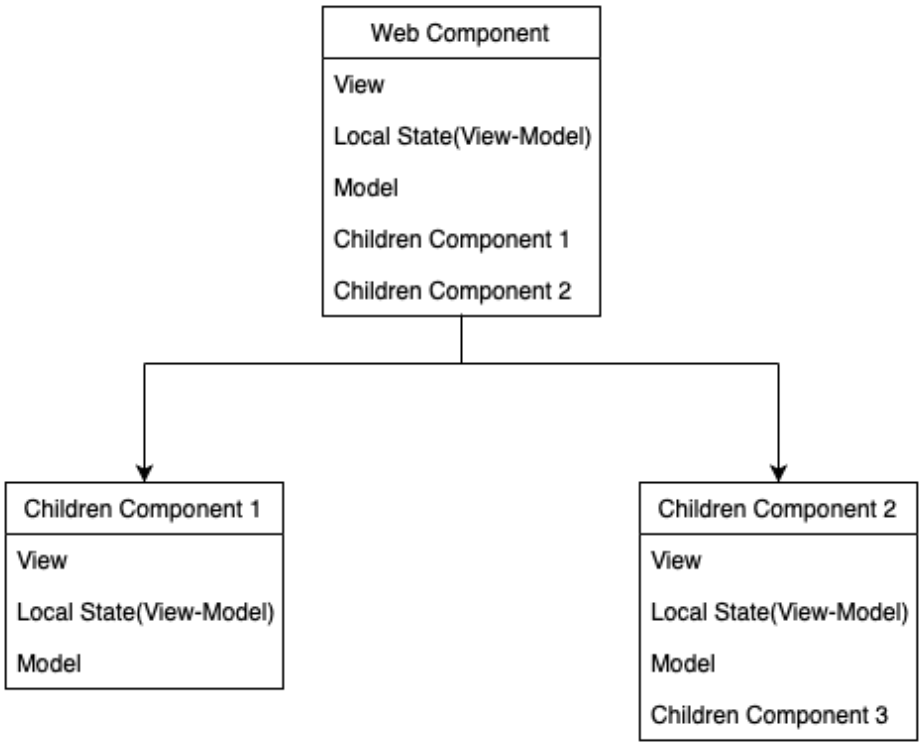
(Here is the MVVM structure link https://drive.google.com/file/d/17X6LOc8LsBT6rYiTjMWmLD2lt3kyWNob/view?usp=sharing)



Presentation layer patterns

For React framework, View-Model layer and View layer make up the Presentation layer together. The **View** layer is the structure, performance and layout of our Training website(such as the CSS,HTML). It will receive users' interactions(clicking, keyboard, etc) and hands these information to View-Model through data-binding and performs the change from the VM layer. **ViewModel** is an object that exposes bind-able properties and methods which are consumed by the view. It is described as a state of data in the model.

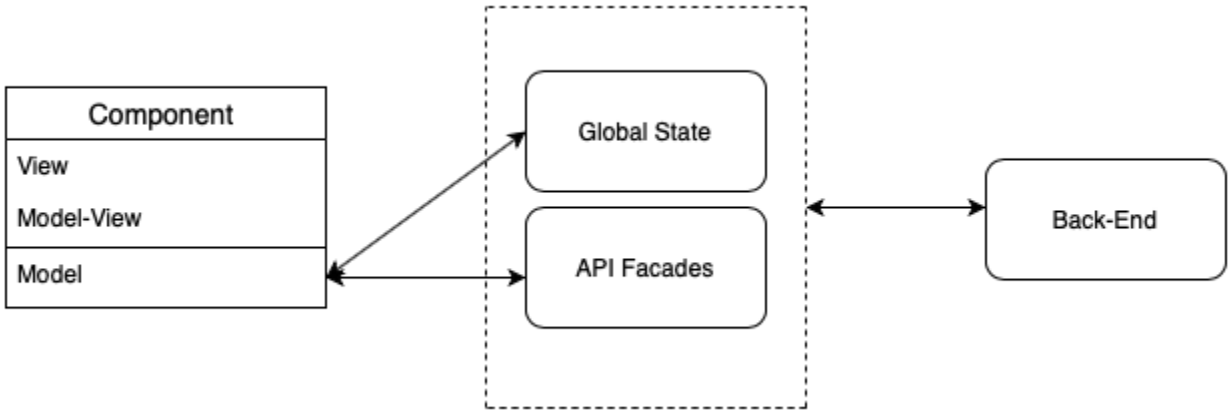
(Here is the Presentation layer link: https://drive.google.com/file/d/1av-ikD5e5n6BuPJ4tZLitn95I6pCYGAa/view?usp=sharing)



Model(Business logical) layer patterns

Model layer is a data access layer which represents real state content.It composes all data from the network layer, databases, services and serve them in an easy way. The main responsibility of this layer is to interact with the Back-end through the API to update the new state. The presentation layer will use these states from users and clients to render our website. You can go to the Research Backend & Database Patterns to view the domain layer and get knowledge of both behaviour and data.

(Here is the Model layer link:https://drive.google.com/file/d/1jAcM6gEPVJC4UCX99cp9mTVYa_6A3UOw/view?usp=sharing)



Security patterns

In order to ensure to access the data in the system successfully and ensure the integrity and confidentiality of the data, the system need to ensure apply some security patterns. In this stage, we consider that the website needs to have two types of security, namely, authentication and authorization of all relevant logic of users in the system.

- **Authentication:** Need to verify that each request send comes from an authenticated user and that the user is legitimate. For example, if students want to post new articles to the website, he first need to perform as a student, so he need to verify his identity through login verification, enter the website, and then perform the required operation.
- **Authorization** Need to verify that each request is executed on behalf of an authorized user, and that members are authorized to perform the actions they request. For example, teaching staff can upload new resources to the website, and he has the right to edit the page in editing page.

Façade Pattern

Some single function accesses require multiple objects to be called at the same time. In order to solve the problem of multiple calls, the remote facade mode is a good choice. By invoking services remotely, can reduce the number of interactions and the complexity of an application.

Remote facade, by providing a thin, coarse-grained facade for fine-grained objects, is an elegant and maintainable design. Once the granularity of the remote facade interface is defined, it's easy to understand. In addition, the remote look and feel is highly compatible, and this pattern is particularly compatible with the domain model pattern.

Back-end

Back-end Design

The backend design includes the backend domain design and the backend class design. In the back-end design, we treat data in an object-oriented way. And the relation between objects is showed in the back-end domain diagram. Also, we split our code into several layers: the routers, the controllers, the middleware, and the proxies. The details of their relations are shown in the back-end class diagram.

Database Design

The database design contains our database design and the patterns that we applied to design the database.

The diagram below is the structure of our database design and here is a link to edit: https://drive.google.com/file/d/1YdoBjteQwrmcFpqA2MAGQJd9JjGjdvGj/view?usp=sharing

Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

The following diagrams represent the current version of the our back-end design, and may subject to change during the development phase.

Communication Diagrams

A *communication diagram* is an extension of object *diagram* that shows the objects along with the messages that travel from one to another. In addition to the associations among objects, *communication diagram* shows the messages the objects send each other.

It offers the same information as a sequence diagram, but while a sequence diagram emphasizes the time and order of events, a communication diagram emphasizes the messages exchanged between objects in an application.

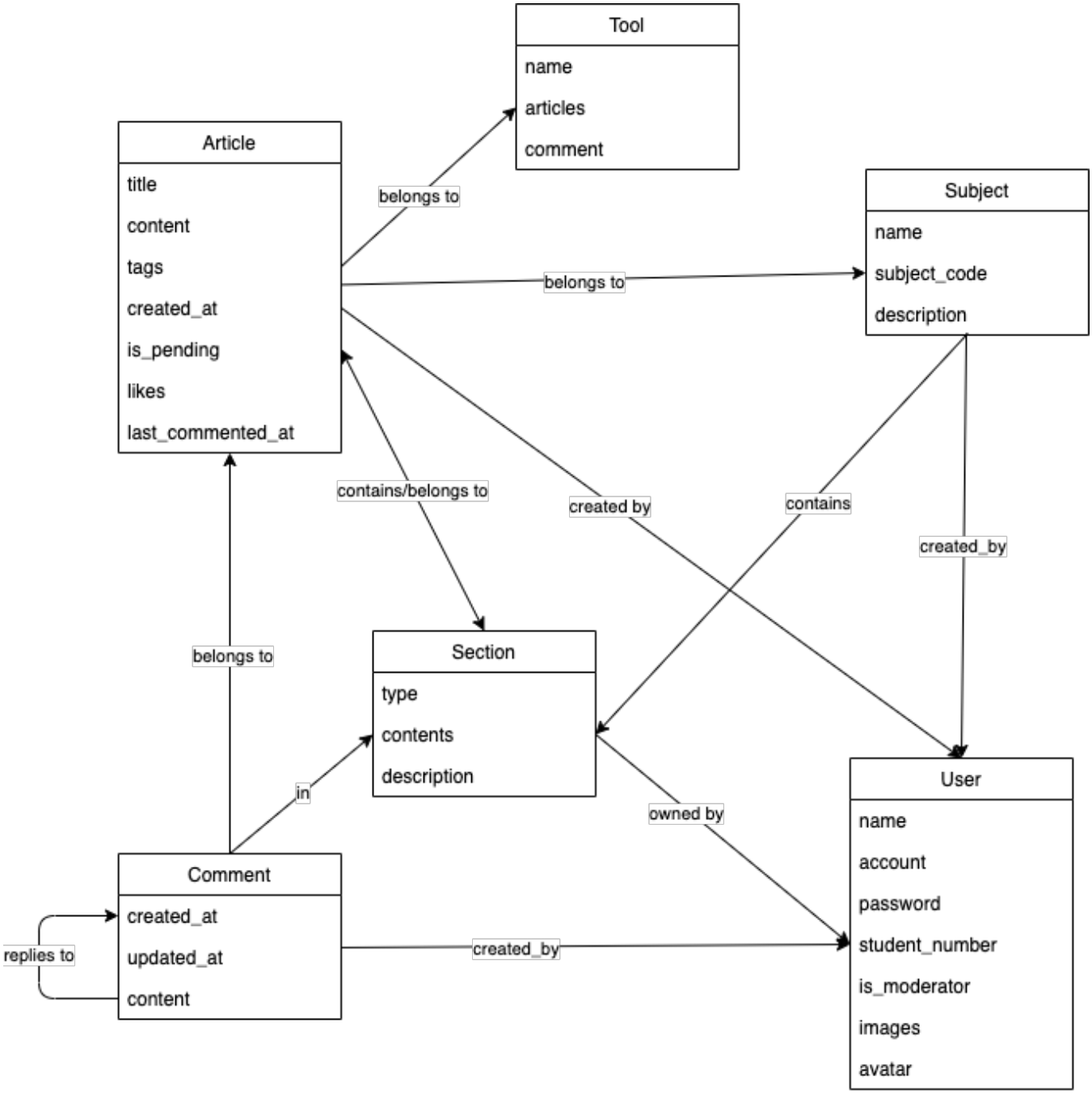
Back-end Design

The backend design includes the backend domain design and the backend class design. In the back-end design, we treat data in an object-oriented way. And the relation between objects is showed in the back-end domain diagram. Also, we split our code into several layers: the routers, the controllers, the middleware, and the proxies. The details of their relations are shown in the back-end class diagram.

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

Domain-Layer Design

The mongoose framework allows us to treat the data in an object-oriented way. Therefore, we used the **Domain Model** pattern in our domain layer design. The domain model constructs a model of the business domain. It has the advantage of extensibility and reusability and thus matches our needs in the project. Our domain design is shown in the domain diagram below:

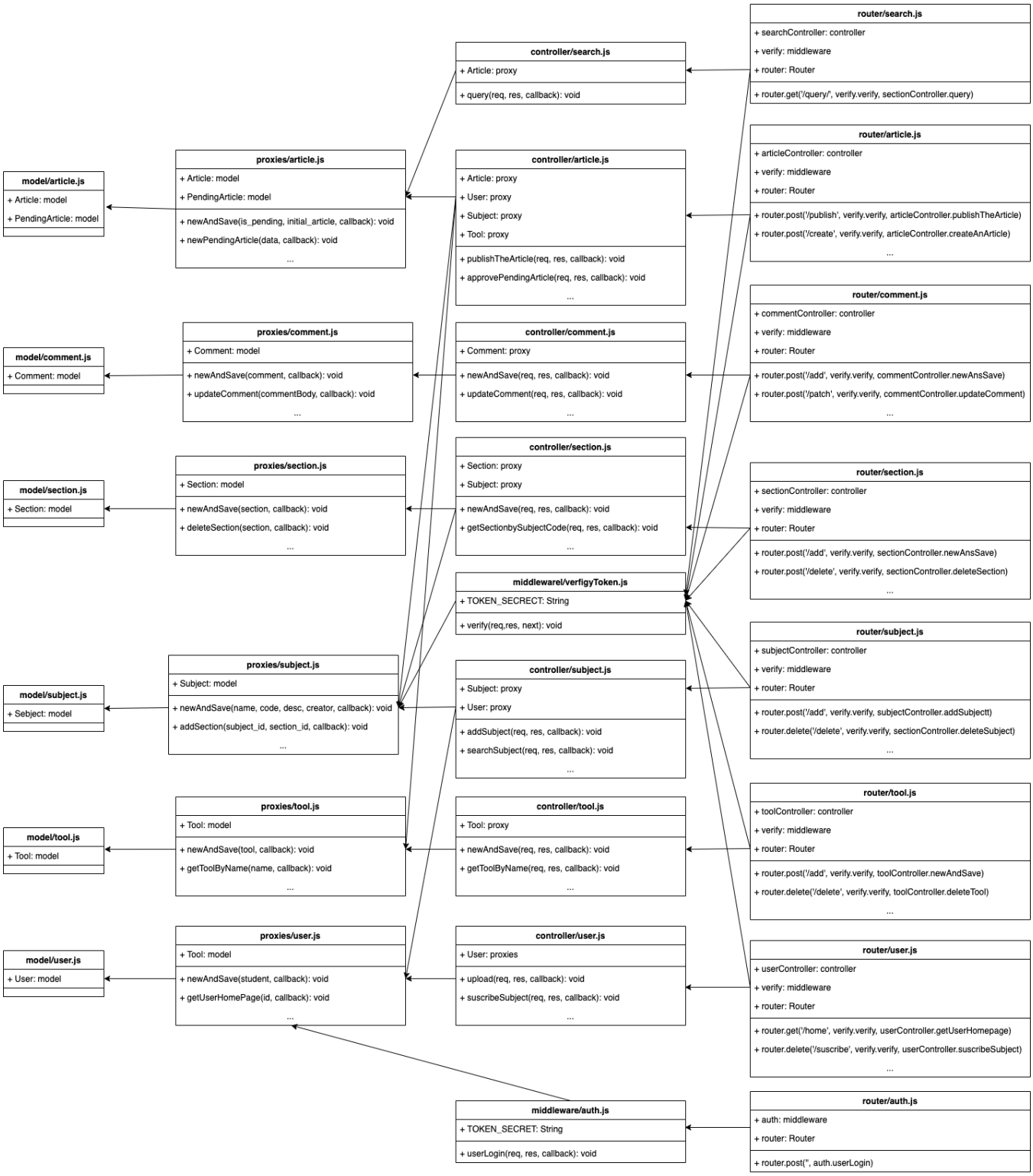


We categorized the objects in our system into four classes:

- **User:** A user can be a student, a moderator or a administrator. Students are normal users of the website. Moderators have the additional responsibility to manage the website (e.g. create subject, approve articles ,etc.). And administrators are able to manage the accounts of students and moderators.
- **Subject:** Each subject has its own page which contain different sections. Moderators can create and manage subject. And student can enroll into a subject.
- **Tools**: Each tool has its own page which contain different sections.
- **Article:** An article can be an article or a pending article. A pending article is an article created by a student and has not been approved by the moderator to publish to the website. An article can be related to a subject or a tool. Tags can be attached to the articles.
- **Section:** Sections can be created by moderators under a subject/tool/article page. A section may be a comment section, an article section, a tool section, etc.
- **Comment:** Comments are contained in sections. They are made be student to comment on articles/subjects/tools.

Back-end Class Design

The back-end class diagram show below demonstrates our backend implementation. However, as there are too many function to include in the diagram, in each class, only two functions were displayed in the diagram as an example to illustrate the design pattern.



There are five layers in our backend design:

- Database Models:
  - We export models in MongoDB as models so that the tables in the database can be treated in an object-oriented way with the help of the Mongoose framework and can be used by the external classes in the system.
  - Models that correspond to a table in the database, each of them contains one or more mongoose object which represents a database table in MongoDB.
- Proxy
  - Each proxy is an intermediate class that provides necessary predefined functions for the controllers interact with the database models
  - Each proxy contains a database model as well as the data IO functions regarding the model.
- Controller

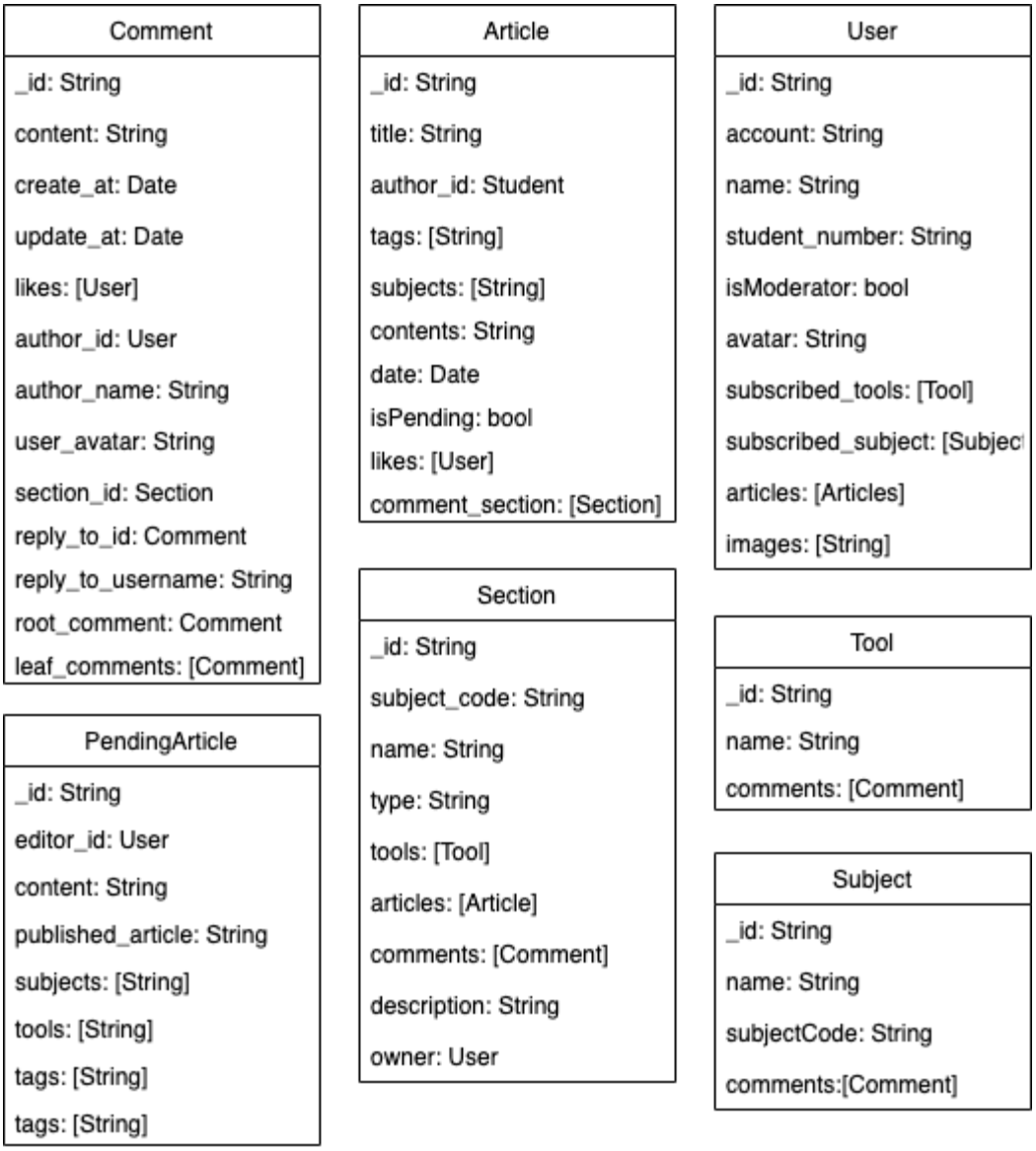
  - Each controller contains one or more proxies for it to access the data in the database.

    - The controllers contain a set of functions that uses a HTTP request and a HTTP response as arguments. Having read the request contents, the function will process the request and create a response to sent back to the caller.
- Router
    - The routers dispatch different requests to the corresponding functions in controllers and collect responses from those functions to send back to the server.
    - Each router defines a router that can perform routing using the Express framework.
- Server
    - There is only one server in the system
    - The server has all the routers and redirects incoming HTTP messages to the corresponding router base on the prefix of its address.

Database Design

The database design contains our database design and the patterns that we applied to design the database.

The diagram below is the structure of our database design and here is a link to edit: https://drive.google.com/file/d/1YdoBjteQwrmcFpqA2MAGQJd9JjGjdvGj/view?usp=sharing

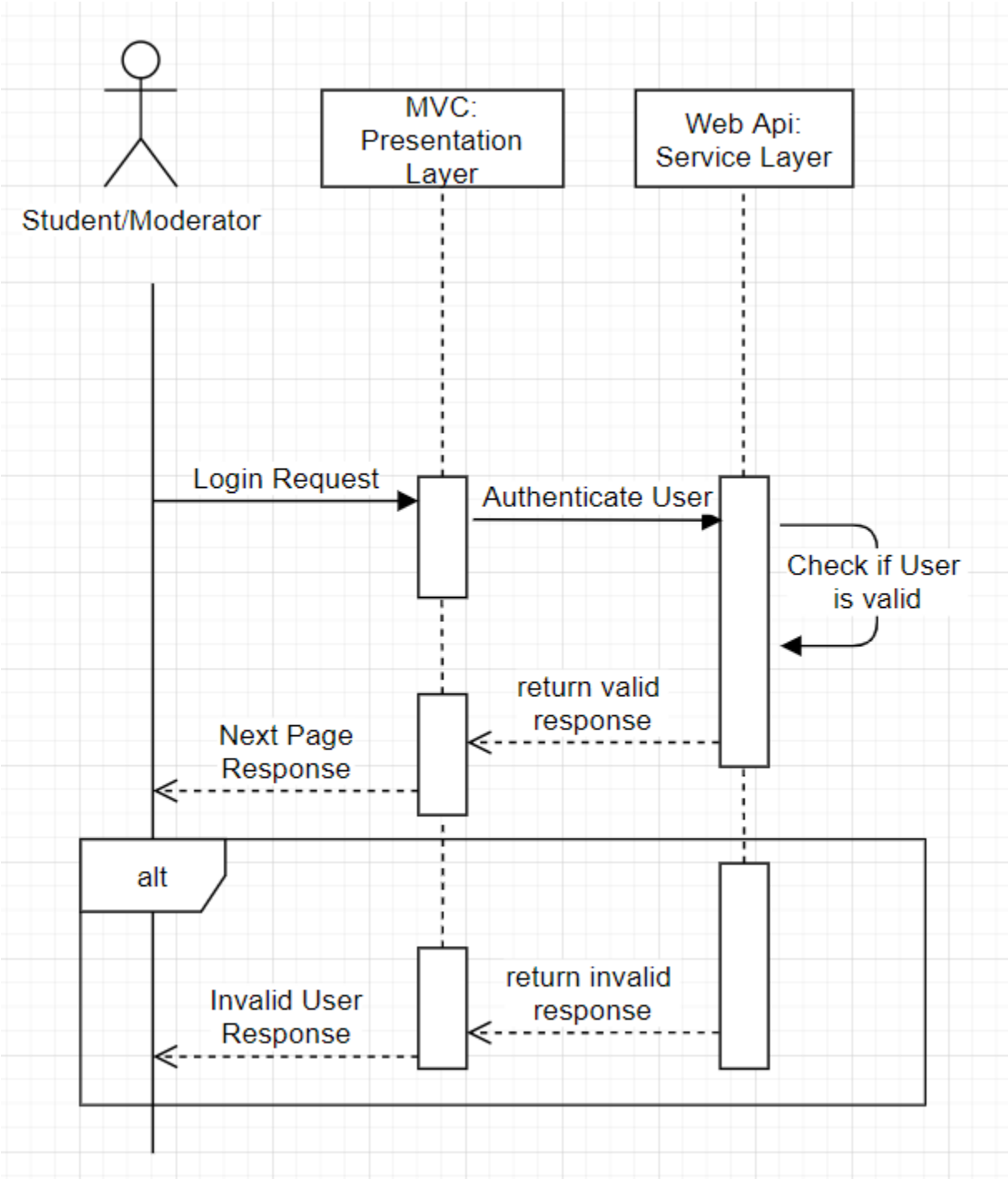


Sequence Diagram

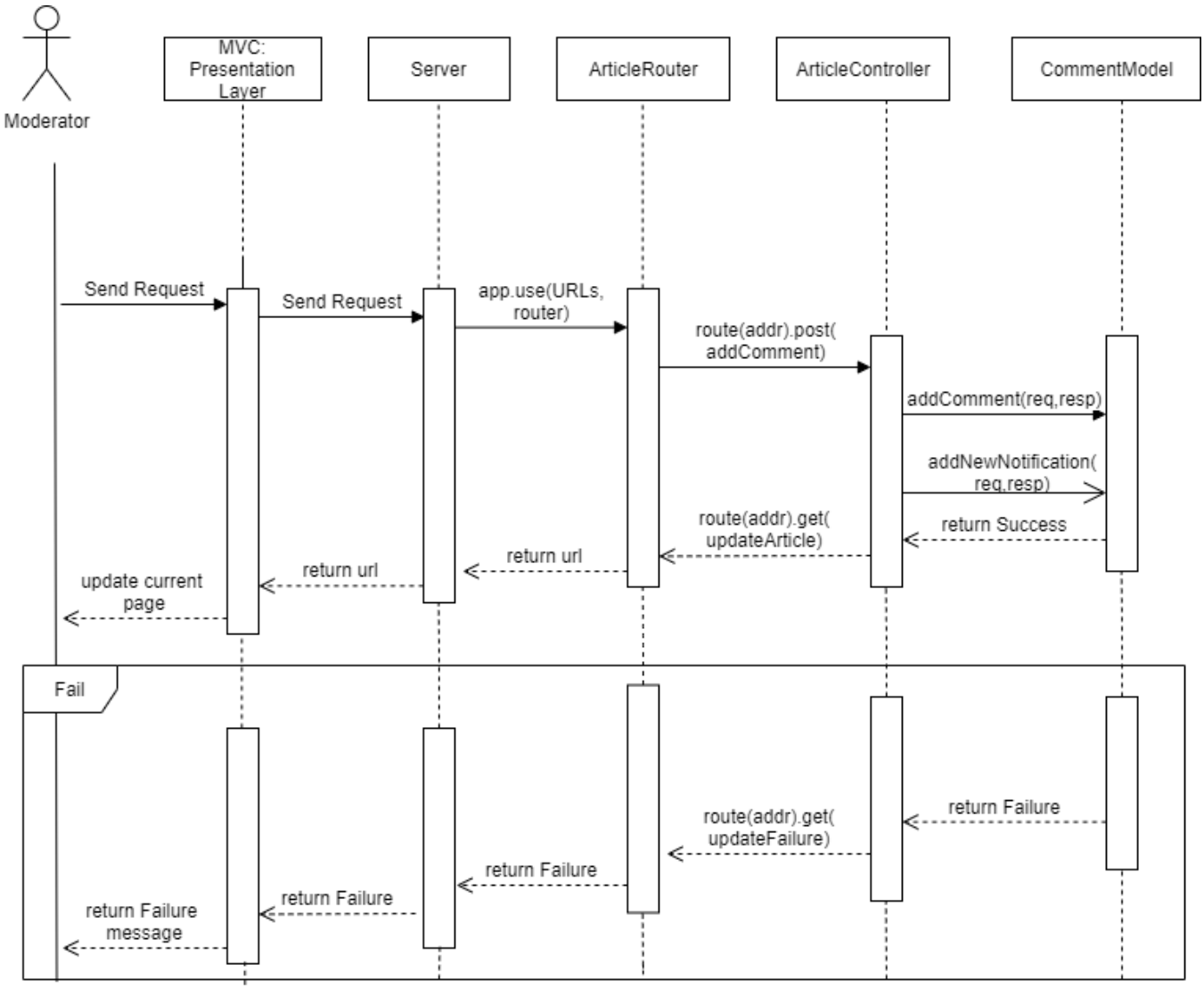
A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

The following diagrams represent the current version of the our back-end design, and may subject to change during the development phase.

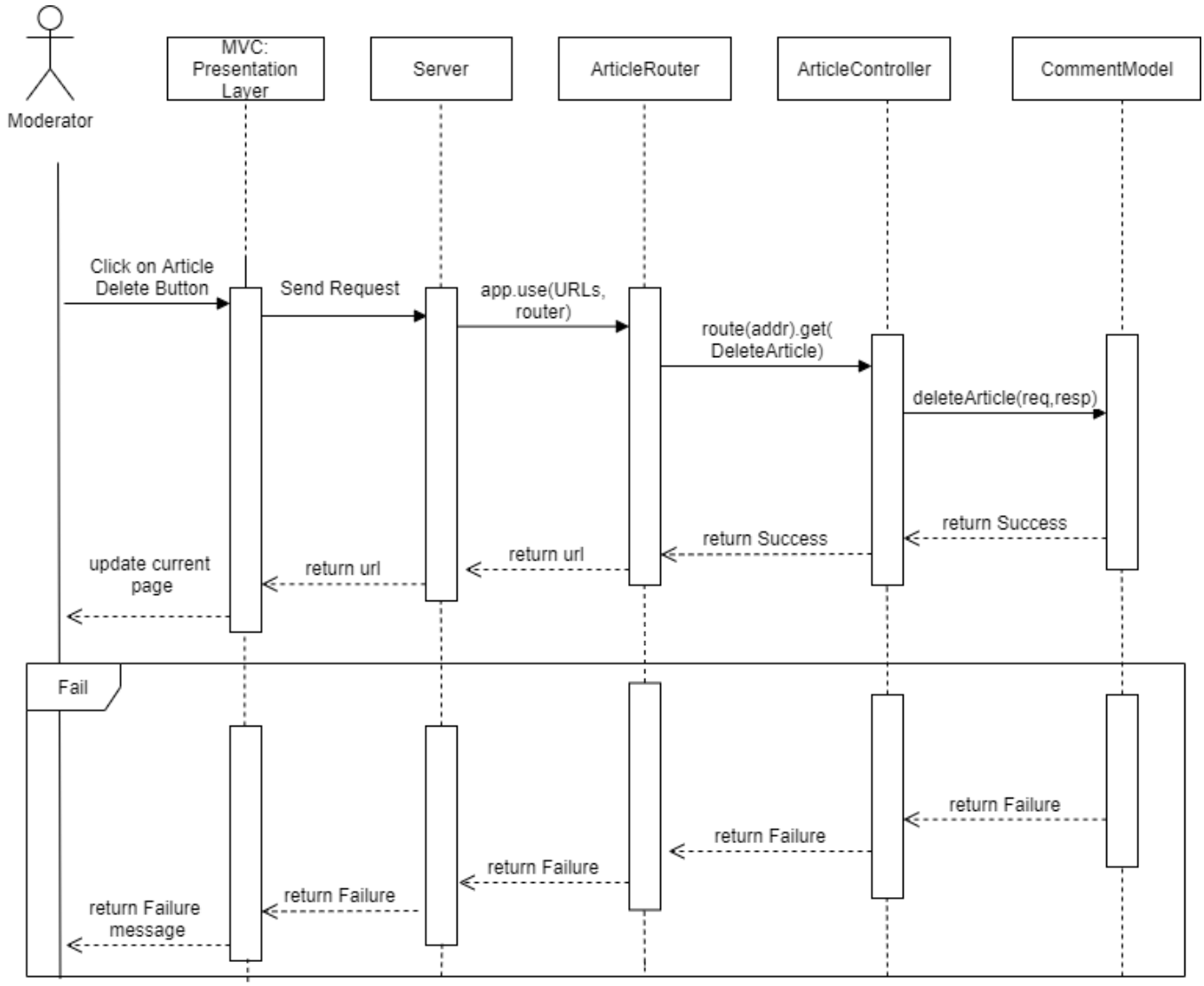
Student/ Moderator Login



Student/Moderator Comment on An Article



Moderator Delete An Article


Moderator
MVC: Presentation Layer
Server
ArticleRouter
ArticleController
CommentModel
Click on Article Delete Button
Send Request
app.use(URLs, router)
route(addr).get(DeleteArticle)
deleteArticle(req,resp)
return Success
return Success
return url
return url
update current page
Fail
return Failure
return Failure
return Failure
return Failure
return Failure message

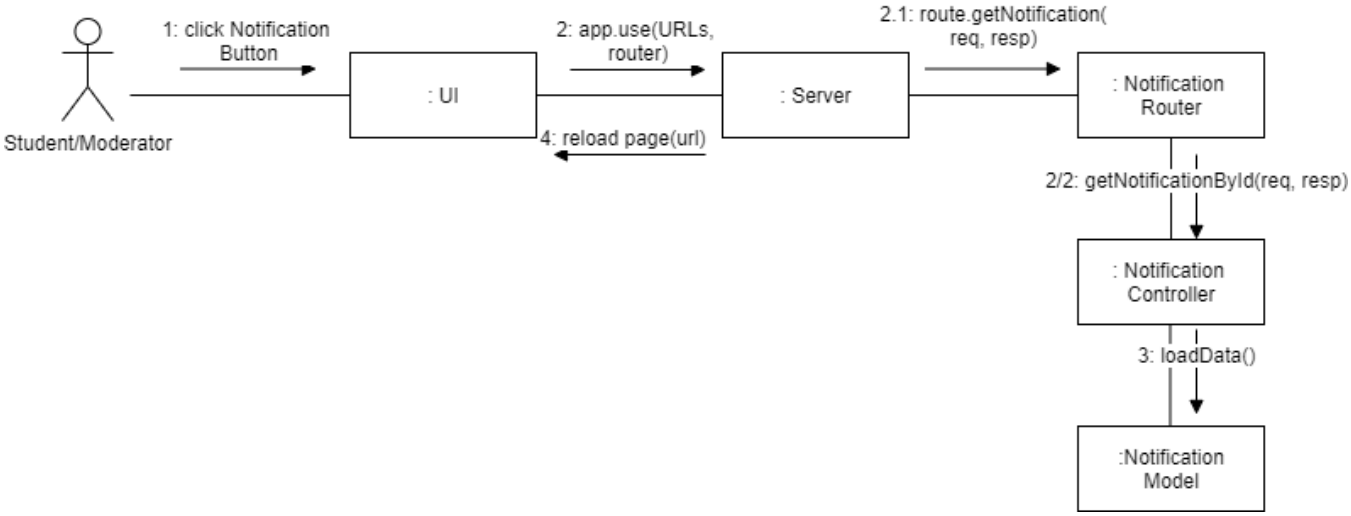
Communication Diagrams

A *communication diagram* is an extension of object *diagram* that shows the objects along with the messages that travel from one to another. In addition to the associations among objects, *communication diagram* shows the messages the objects send each other.

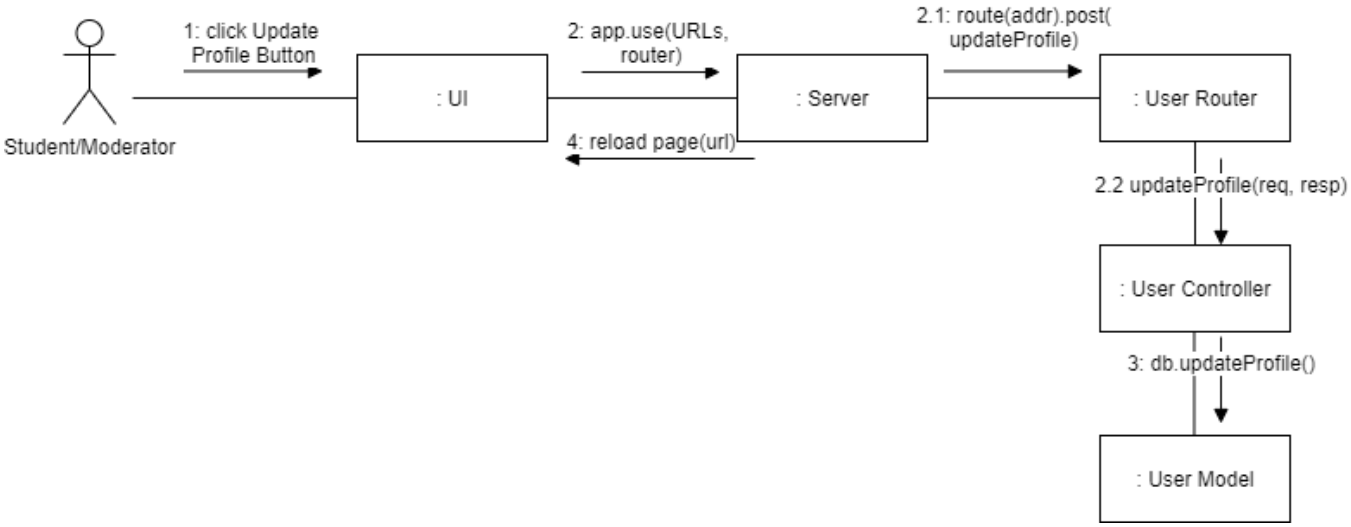
It offers the same information as a sequence diagram, but while a sequence diagram emphasizes the time and order of events, a communication diagram emphasizes the messages exchanged between objects in an application.

The following diagrams represent the current version of the our back-end design, and may subject to change during the development phase.

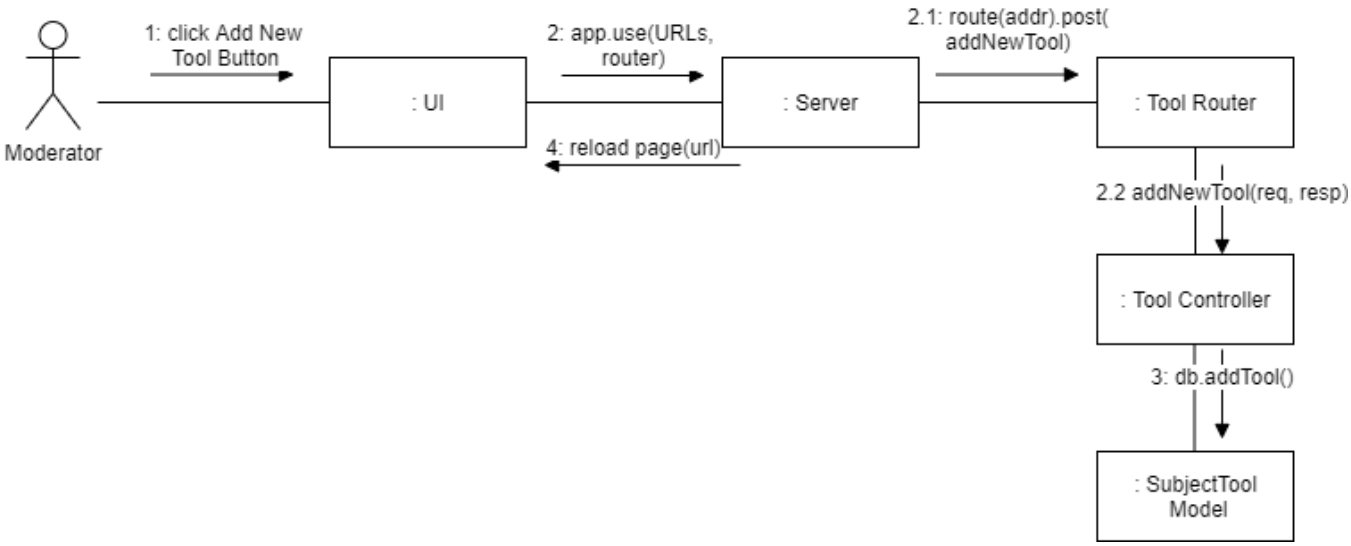
Student/Moderator Read Notification



Student/Moderator Update Personal Profile



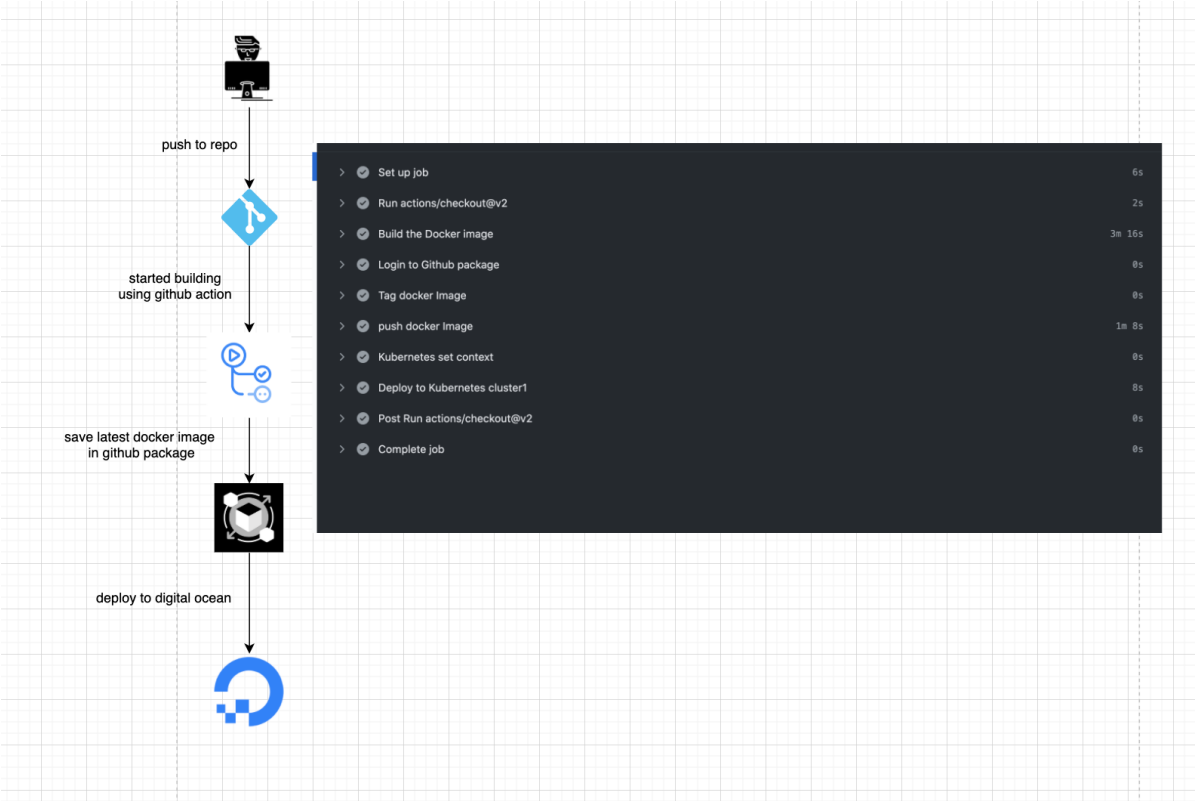
Moderator Add A New Tool


Moderator
1: click Add New Tool Button
: UI
2: app.use(URLs, router)
: Server
2.1: route(addr).post(addNewTool)
: Tool Router
4: reload page(url)
2.2 addNewTool(req, resp)
: Tool Controller
3: db.addTool()
: SubjectTool Model

Deployment

Deployment Plan

Deployment Plan



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

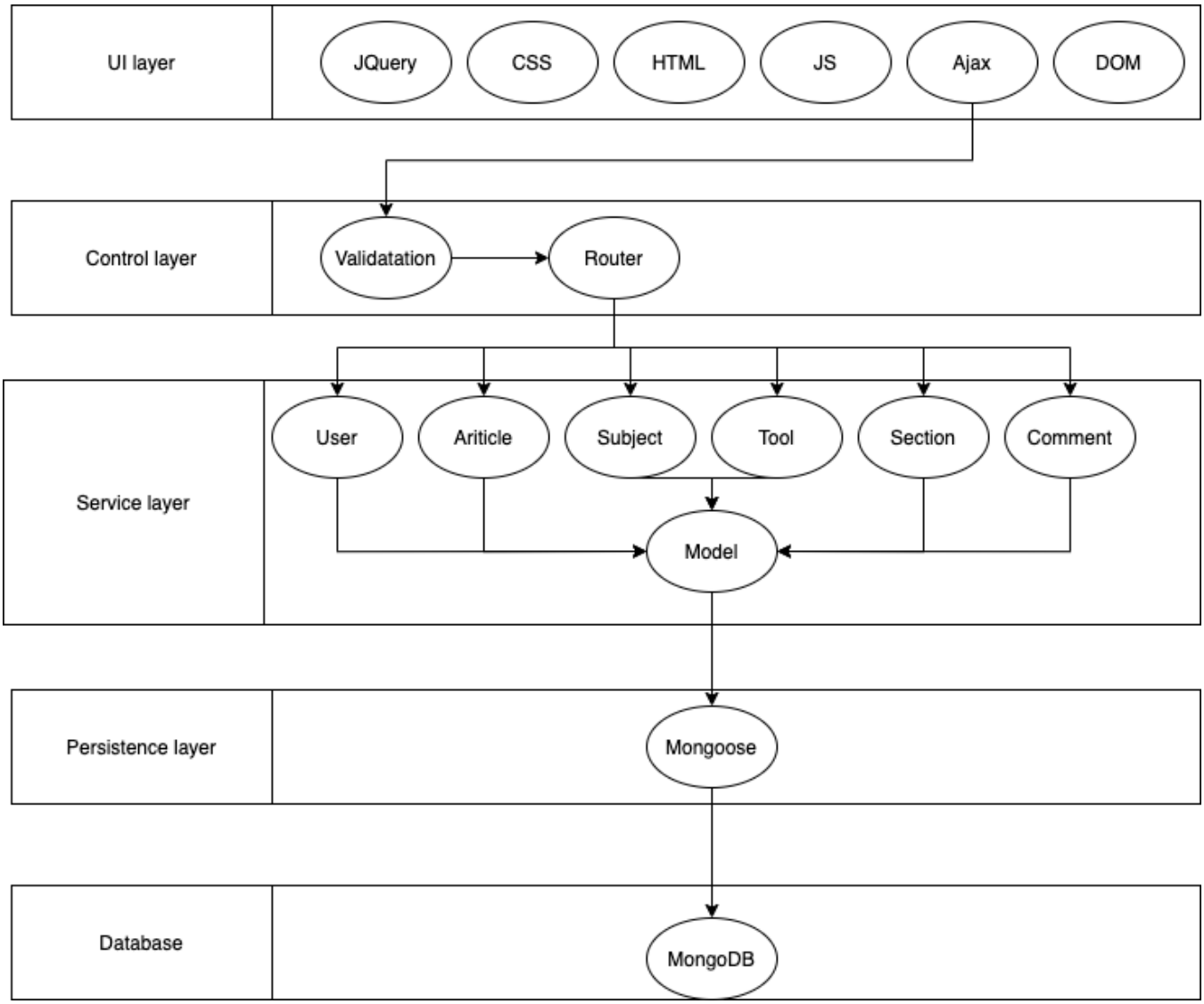
It contains all the core application logic, with a focus on implementing business rules and containing the business value.

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

Quality Assurance

Code Review Checklist

		Checklist name	Note	Yes /No
1	**General**	Does the code work?	• Does it perform its intended function • Does the logic is correct?	
2		Is all the code easily understood?		
3		Does it conform to your agreed coding conventions? (Follow Airbnb Javascript style guild)	• Dose location of braces is correct? • Dose variable and the function name are correct? • Dose line length is under limits? • Indentations correctly? • Formatting? • Comment?	
4		Is there any redundant or duplicate code?		
5		Is the code as modular as possible?		
6		Can any global variables be replaced?		
7		Is there any commented out code?		
8		Do loops have a set length and correct termination conditions?		
9		Do the names used in the program convey intent?		
10		No hard coding should be used with a heavier focus on constants and configuration values.		
11		Does it avoid many multiple if/else block?		
12	**Performance**	Are there any obvious optimizations that will improve performance?		
13		Can any of the code be replaced with the library or built-in functions?		
14		Can any logging or debugging code be removed?		
15	**Security**	Are all data inputs checked (for the correct type, length, format, and range) and encoded?		
16		Are output values checked and encoded?		
17		Are invalid parameter values handled?		
18	**Testing**	Is the code testable?	• The code should be structured so that it doesn't add too many or hide dependencies. • Is it unable to initialize objects, test frameworks can use methods etc.	
19		Do tests exist and are they comprehensive?		
20		Do unit tests actually test that the code is performing the intended functionality?		
21		Could any test code be replaced with the use of an existing API?		

History/Vision

Version	Published	Changed By	Comment
CURRENT (v. 2)	**25 May, 2020 12:01**	Patawat Watakul	Organise the file
v. 1	25 May, 2020 11:54	Patawat Watakul	

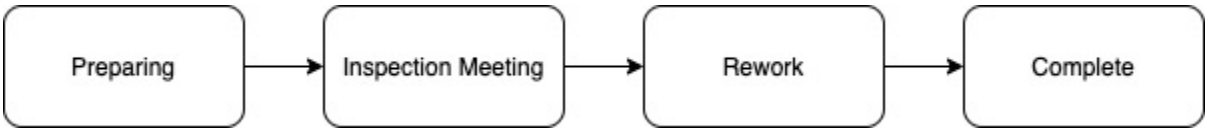
References:

- https://nyu-cds.github.io/effective-code-reviews/03-checklist/
- https://medium.com/cuelogic-technologies/code-review-process-best-practices-3eeecab26ded
- https://google.github.io/eng-practices/review

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

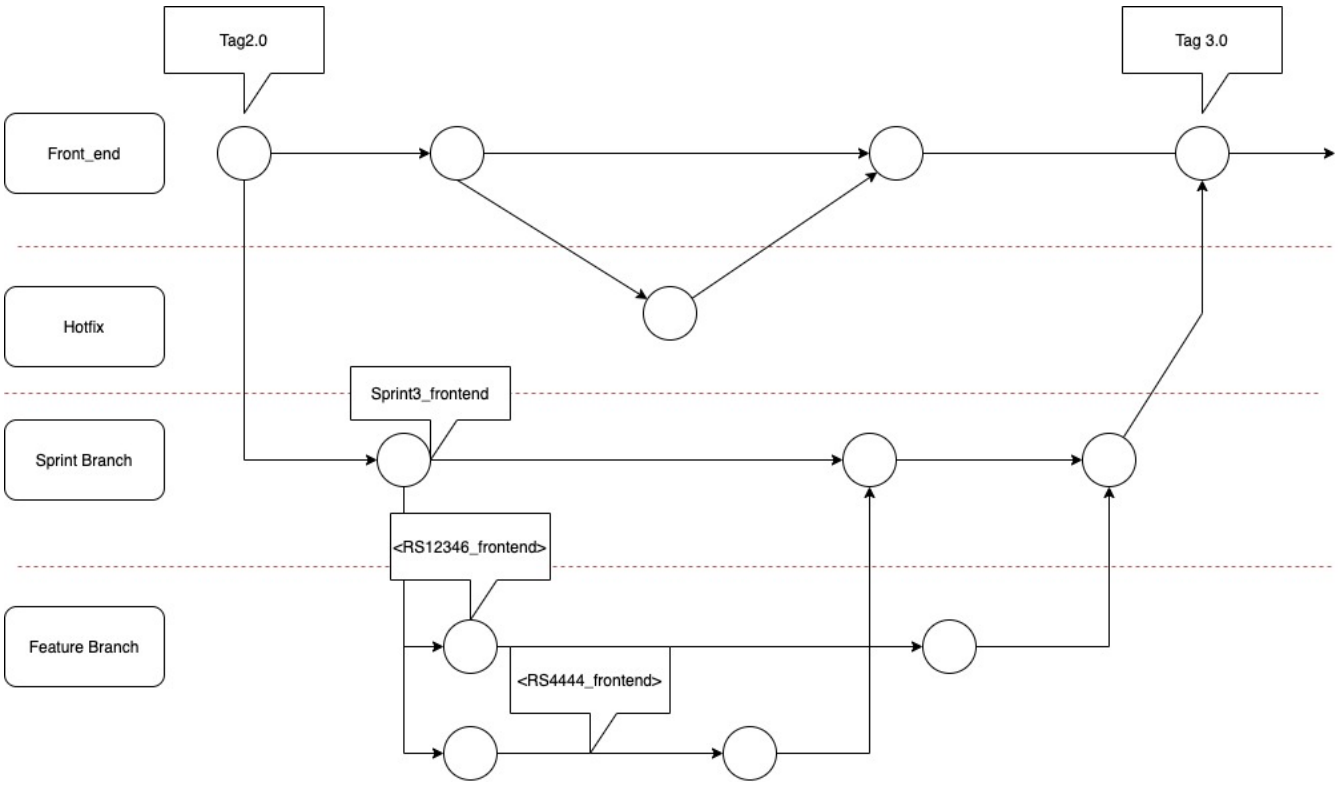
Complete

- Merge the code into the Main branch.

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

Resources

API authentication

Every request must include in the header "auth-token" followed by an authentication token, stored under localStorage.getItem("accessToken") after the login has authenticated the user.

API for article

Publish an article

Request URL: /article/publish.

Request Method: POST

Request Arguments:

Argument name	Necessity	Type
title	yes	string
content	yes	string
tags	yes	[string]
subjects	if tools yes, no	[string]
tools	if subjects yes, no	[string]
article_id	no	string

article_id:

if this publish aims to submit a new article, then article_id is not needed (And cannot be one)

else if this publish aims to submit an edit to an existing article, then the _id of the existing article should be sent along

Return arguments:

Argument name	Necessity	Type
auth_token	yes	string
article_id	(only if success = true)	string
success	yes	bool
error_info	no	string

Get an article by id

Request URL: /article/get/:id

Request Method: GET

Return arguments:

if success

Argument name	Necessity	Type
success	yes	bool
returnValuesForArticle	only if succeed	JSON
auth_token	yse	string

returnValuesForArticle example:


```
"returnValuesForArticle": {

        "title": "How to edit an article",
        "like_number": 0,
        "create_at": "2020-08-26T05:45:54.302Z",
```

"content": "initial contents",
"comments": [],
"id": "5f45f712fc902a2f60067844",
"is_liked": false,
"current_article": "5f45f712fc902a2f60067843",
"pending_article": [],
"rejected_articles": [],
"history_articles": []

},

if not success

Argument name	Necessity	Type
success	yes	bool
error_info	no	string
auth_token	yes	string

Get an pending article by id

Request URL: /article/getPendingArticle/:id

Request Method: GET

Return arguments:

if success

Argument name	Necessity	Type
success	yes	bool
returnValuesForArticle	only if succeed	JSON
auth_token	yse	string

returnValuesForArticle example:

"returnValuesForArticle": {

"title": "How to edit an article",
"like_number": 0,
"create_at": "2020-08-26T05:45:54.302Z",
"content": "initial contents",
"comments": [],
"id": "5f45f712fc902a2f60067844",
"is_liked": false,
"subjects": "5f45f712fc902a2f60067843",
"tools": [],
"tags": [],

},

if not success

Argument name	Necessity	Type
success	yes	bool
error_info	no	string
auth_token	yes	string

Create an article

Request URL: /article/create

Request Method: GET

Return arguments:

if success:

Argument name	Necessity	Type
success	yes	bool
tagsForSubjects	yes	[string]
tagsForTools	yes	[string]
success	yes	bool
auth_token	yes	string

if not success

Argument name	Necessity	Type
success	yes	bool
error_info	no	string
auth_token	yes	string

Edit an article

Request URL: /article/edit/:id

Request Method: GET

Return arguments:

If success

Argument name	Necessity	Type
success	yes	bool
returnValuesForArticle	yes	JSON
tagsForSubjects	yes	[string]
tagsForTools	yes	[string]
auth_token	yes	string

returnValuesForArticle example:

```
"returnValuesForArticle": {
        "title": "Intro to ML",
        "content": "#Heading1\n ##Heading2\nsljfoeijflaejgoeawjgaofj",
        "id": "5f44c6bc04e5b4cc822db5df"
},
```

If not success

Argument name	Necessity	Type
success	yes	string
err_info	yes	string
auth_token	yes	string

Edit an pending article

Request URL: /article/editPendingArticle/:id

Request Method: GET

Return arguments:

If success

Argument name	Necessity	Type
success	yes	bool
returnValuesForArticle	yes	JSON
auth_token	yes	string

returnValuesForArticle example:


```
"returnValuesForArticle": {
    "title": "Java Tutorial 1757",
    "content": "#Heading1\n ##Heading2\nsljfoeijflaejgoeawjgaofj",
    "id": "5f82fac8eb0346316ac7cf8f"
},
```


If not success

Argument name	Necessity	Type
success	yes	string
err_info	yes	string
auth_token	yes	string

Delete an article

Request URL: /article/delete/:id

Request Method: GET

Return arguments:

Argument name	Necessity	Type
success	yes	bool
reason	yes	string
auth_token	yes	string

Get all published articles names of a subject or a tool

Request URL: /article/getNamesOfArticles/:subject_code_or_tool_name

Request Method: GET

Return arguments:

if succuss:

Argument name	Necessity	Type
success	yes	string
articles	yes	[{ "name":name,

		"id":id }]
auth_token	yes	string

if not success

Argument name	Necessity	Type
success	yes	string
err_info	yes	string
auth_token	yes	string

:subject_code_or_tool_name : the code of a subject, for example "SWEN90009"; the name of a tool, for example "Java"

Approve a pending article

Request URL: /article/pendingArticle/approve/:pending_article_id

Request Method: GET

Return arguments:

if succuss:

Argument name	Necessity	Type
success	yes	true

articles	yes	json, e.g.
		
		{
		"subjects": [
		"SWEN20003",
		"SWEN90007"
		],
		"tools": [
		"Java"
		],
		"tags": [
		"Java",
		"JVM"
		],
		"is_pending": false,
		"likes": [],
		"comments": [],
		"_id": "5f607b43355fa756e66cc809",
		"create_at": "2020-09-15T08:28:51.176Z",
		"title": "Java Tutorialasdfghj",
		"author_id": "5f322c715952dd35b0285c2d",
		"content": "#Heading1\n ##Heading2\nsljfoeijflaejgoeawjgaofj",
		"__v": 0
		}

auth_token	yes	string

if not success

Argument name	Necessity	Type
success	yes	string
err_info	yes	string
auth_token	yes	string

Reject a pending article

Request URL: /article/pendingArticle/reject/:pending_article_id

Request Method: GET

Return arguments:

if succuss:

Argument name	Necessity	Type
success	yes	true
auth_token	yes	string

if not success

Argument name	Necessity	Type
success	yes	string
err_info	yes	string
auth_token	yes	string

Get all pending articles names of a subject or a tool

Request URL: /article/getNamesOfPendingArticles/:subject_code_or_tool_name

Request Method: GET

Return arguments:

if succuss:

Argument name	Necessity	Type
success	yes	string
articles	yes	[{ "name":name, "id":id }]
auth_token	yes	string

API for Comment

Get a comment (for testing)

Request URL: /comment/get/

Required arguments:

Argument name	Necessity	Type
_id	yes	ObjectID

Request Method: GET

Return Arguments:

if success

Argument name	Necessity	Type
topic_id	yes	ObjectID
author_id	yes	ObjectID
root_comment_id	yes	ObjectID
reply_to_id	no	ObjectID
reply_to_name	no	ObjectID
content	yes	String
create_at	no	Date
update_at	no	Date
likes	no	[ObjectID]

else

Argument name	Necessity	Type
success	yes	Bool
error_info	no	String

Create a comment

Request URL: /comment/add

Request Method: POST

Arguments:

Argument name	Necessity	Type
topic_id	yes	ObjectID
author_id	yes	ObjectID
root_comment_id	no	ObjectID
reply_to_id	no	ObjectID
reply_to_name	no	ObjectID
content	yes	String

Return arguments:

Argument name	Necessity	Type
success	yes	bool
error_info	no	string

Delete a section

Request URL: /comment/delete

Request Method: DELETE

Required arguments:

Argument name	Necessity	Type
id	yes	ObjectID

Return arguments:

Argument name	Necessity	Type
Argument name	**Necessity**	**Type**
success	yes	bool
error_info	no	string

Update a section

Request URL: /comment/patch

Request Method: PATCH

Arguments:

Argument name	Necessity	Type
topic_id	no	ObjectID
author_id	no	ObjectID
content	no	String

Return arguments:

Argument name	Necessity	Type
success	yes	bool
error_info	no	string

API for homepage

Home Page

API for Followed Subjects of user

- ID
- Name
- Description

Ex.


```
"subjects":[
{
"id": 1,
"title": "SWEN10000",
"des": "SWEN10000",
},
{
"id": 2,
"title": "SWEN10000",
"des": "SWEN10000",
}
]
```


API for Followed Topics of user

- ID
- Name
- Description


```
"topics":[
{
"id": 1,
"title": "REACT",
"des": "test"
}
]
```

Api for images

Create new images

Request URL: /user/upload

Method : POST(Form data)

Headers

Argument name	Necessity	Type
auth-token	yes	String

Form data

Argument name	Necessity	Type
image	yes	File

Response

Argument name	Necessity	Type
Success	yes	Boolean
imageurl	yes	String

Get images

Request URL: /user/images

Method : GET

Headers

Argument name	Necessity	Type
auth-token	yes	String

Response

Argument name	Necessity	Type
Success	yes	Boolean
images	yes	[String]

API for Section

Get a section

Since sections depend on a subject, one cannot get a section directly.

Instead, get corresponding sections from the subject

Request URL: /subject/get/

Required arguments:

Argument name	Necessity	Type
_id	yes	ObjectID

Request URL: /subject/get/:subject_code

Request Method: GET

Return Arguments:

if success

Argument name	Necessity	Type
name	yes	String
subject_code	yes	String
description	yes	[String]
sections	yes	[ObjectID]

else

Argument name	Necessity	Type
success	yes	Bool
error_info	no	String

Create a section

Request URL: /section/add

Request Method: POST

Arguments:

Argument name	Necessity	Type
subject_code	yes	String
type	yes	String
tools	no	[ObjectID]
articles	no	[ObjectID]
comments	no	[ObjectID]
description	no	[String]

Return arguments:

Argument name	Necessity	Type

success	yes	bool
error_info	no	string

Delete a section

Request URL: /subject/delete

Request Method: DELETE

Note: When a moderator edits a subject page, deleting a section or sections and confirming with the changes,

the corresponding sections will be deleted through the cascade delete approach.

Required arguments:

Argument name	Necessity	Type
name	no	string
subject_code	no	string
description	no	[string]
sections	yes	[ObjectId]

Return arguments:

Argument name	Necessity	Type
success	yes	bool
error_info	no	string

Update a section

Request URL: /section/patch

Request Method: PATCH

Arguments:

Argument name	Necessity	Type
subject_code	no	String
type	no	String
tools	no	[ObjectID]
articles	no	[ObjectID]
comments	no	[ObjectID]
description	no	[String]

Return arguments:

Argument name	Necessity	Type
success	yes	bool
error_info	no	string

API for Subject

Get a subject

Request URL: /subject/get/:subject_code

Request Method: GET

Request URL: /subject/get

Request Method: GET

Required Arguments:

Argument name	Necessity	Type
_id	yes	ObjectID

Return Arguments:

if success

Argument name	Necessity	Type
name	yes	String
subject_code	yes	String
description	yes	[String]
sections	yes	[ObjectID]

else

Argument name	Necessity	Type
success	yes	Bool
error_info	no	String

Create a subject

Request URL: /subject/add

Request Method: POST

Required arguments:

Argument name	Necessity	Type
name	yes	String
subject_code	yes	String
description	no	[String]

Return arguments:

Argument name	Necessity	Type
success	yes	bool
result	yes	JSON (If succeeds)
error_info	no	string

Delete a subject

Request URL: /subject/delete

Request Method: DELETE

Note: When a moderator deletes a subject, corresponding sections will also be deleted through the cascade delete approach.

Required arguments:

Argument name	Necessity	Type
_id	yes	ObjectID

Return arguments:

Argument name	Necessity	Type
success	yes	bool
error_info	no	string

Update a subject

Request URL: /subject/patch

Request Method: PATCH

Required arguments:

Argument name	Necessity	Type
name	no	string
subject_code	no	string
description	no	[string]
sections	no	[ObjectId]

Return arguments:

Argument name	Necessity	Type
success	yes	bool
result	yes	JSON (If succeeds)
error_info	no	string

Search a subject

Request URL: /subject/search

Request Method: PATCH

Required arguments:

Argument name	Necessity	Type
name	no	string

Return arguments:

Argument name	Necessity	Type
success	yes	bool

result	yes	JSON (If succeeds)
error_info	no	string

API for User Profile

Get subscribed subjects/tools/ moderated subjects

Request URL: /user/home

Request Method: GET

Requirement: the user needs to be logged in first and a valid token is provided.

Required Arguments:

Argument name	Necessity	Type

Return Arguments:

if success

Argument name	Necessity	Type
name	yes	String
subject_code	yes	String
description	yes	[String]
subscribed_tools	yes	[ObjectID]
subscribed_subjects	yes	[ObjectID]
moderated_subjects	yes	[ObjectID]

the same API for getting the home page of a user, these **three bold properties** are populated with their corresponding details.

else

Argument name	Necessity	Type
success	yes	Bool
error_info	no	String

Get pending articles of a user

Request URL: /article/getAllPendingArticlesByUserId

Request Method: GET

Return arguments:

If success:

Argument name	Necessity	Type
success	yes	bool
articles	yes	JSON
auth_token	no	string

articles example:


```
"articles": [
    {
        "name": "Java Tutorial 1757",
        "id": "5f82fac8eb0346316ac7cf8f"
    },
    {
```

"name": "Java Tutorial 1757",
"id": "5f82fad5eb0346316ac7cf90"
}
]

if not success

Argument name	Necessity	Type
success	yes	bool
error_info	no	string
auth_token	yes	string

Get published articles of a user

Request URL: /article/getAllPublishedArticlesByUserId

Request Method: GET

Return arguments:

If success:

Argument name	Necessity	Type
success	yes	bool
articles	yes	JSON
auth_token	no	string

articles example:

"articles": [
{
"name": "Java Tutorial 1757",
"id": "5f82fac8eb0346316ac7cf8f"
},
{
"name": "Java Tutorial 1757",
"id": "5f82fad5eb0346316ac7cf90"
}
]

if not success

Argument name	Necessity	Type
success	yes	bool
error_info	no	string
auth_token	yes	string

Get all comments of a user

Request URL: /user/comments

Request Method: GET

Note:

Requirement: the user needs to be logged in first and a valid token is provided.

Required arguments:

Argument name	Necessity	Type

Return arguments:

Argument name	Necessity	Type
success	yes	bool
comments	yes	[ObjectID]
error_info	no	string

example:


```
{"success": true,

"comments": [
    {
        "leaf_comments": [],
        "likes": [],
        "_id": "5f6b5854785e671d887029ab",
        "author_id": "5ed34efd9293881d50280b1d",
        "section_id": "5f699fecd3e0cc6ee406a112",
        "content": "this is the very first comment",
        "create_at": "2020-09-23T14:14:44.198Z",
        "update_at": "2020-09-23T14:14:44.199Z",
        "author_name": "test.16.8.2020",
        "user_avatat": "testAvatar",
        "__v": 0
    },

 {
        "leaf_comments": [],
        "likes": [],
        "_id": "5f6b58834a84847bc046a523",
        "author_id": "5ed34efd9293881d50280b1d",
        "section_id": "5f699fecd3e0cc6ee406a112",
        "content": "this is the very first comment",
        "create_at": "2020-09-23T14:15:31.446Z",
        "update_at": "2020-09-23T14:15:31.450Z",
        "author_name": "test.16.8.2020",
        "user_avatat": "testAvatar",
        "__v": 0
    }

]

}
```

Use Cases

1.1 Moderator View

UC001 Moderator Signs In

Pre-conditions

The moderator has a valid account on the system.

Main Events Flow

1. The instructor visits the sign-in page.
2. The instructor inputs the username and password.
3. The instructor confirms the login operation.
4. The system validates the account information. [AF01]
5. The system signs in the account and redirects to the home page.

Alternative Flows

[AF01] Account information is incorrect

1. The system detects that the account information provided by the user is incorrect.
2. The system displays a message showing that the username and/or password are wrong.
3. Go to the main event flow step 2.

Post-conditions

The moderator is signed in and redirected to the home page. The system stores a token which will be used for authentication dependent operations.

UC002 Moderator Views Moderating Subject List

Pre-conditions

The moderator is signed in and is moderating the subjects to be shown on the page.

Main Event Flow

1. The moderator visits the home page**.**

Post-conditions

All subjects that are currently moderated by the moderator will be shown on the home page with their names, details, and links.

UC003 Moderator Adds New Subject

Pre-conditions

The moderator is signed in and on the home page.

Main Event Flow

1. The moderator clicks the "Add Subject" button.
2. The moderator fills in the detail of the new subject.
3. The moderator clicks the “Publish" button to publish the new subject.

Alternative Flows

[AF01] The moderator wants to cancel the operation

1. The moderator clicks the "Add Subject" button but decides not to add the subject.
2. The moderator clicks the "Close" button to go back to the home page.

Post-conditions

The new subject is then created and saved, and it will be shown on the moderator's home page.

UC004 Moderator Views Detailed Subject Information

Pre-condition

The moderator is signed in and on the home page, and the moderator is moderating the subject to be visited.

Main Event Flow

1. The moderator clicks the subject link.

Post-conditions

The moderator will be directed to the subject details page.

UC005 Moderator Edits Subject Description

Pre-conditions

The moderator is signed in and on the subject detail page of the subject to be edited.

Main Event Flow

1. The moderator clicks the "Edit Description" button.
2. The moderator edits the description of the subject.
3. The moderator clicks the "Publish" button to publish the new description.

Alternative Flows

[AF01] The moderator wants to cancel the operation

1. The moderator clicks the "Edit Description" button but decides not to edit the description.
2. The moderator clicks the "Close" button to go back to the subject details page.

Post-conditions

The subject description is then changed and saved, and the new description will be updated on the subject details page.

UC006 Moderator Deletes Comments

Pre-conditions

The moderator has a valid account on the system and already logged in.

Main Events Flow

1. The moderator locates his/her own comment on the current page. [AF01]
2. The moderator deletes the comment. [AF02]
3. The system validates the operation.
4. The system then shows the page again but with the corresponding comment deleted.

Alternative Flows

[AF01] Moderator deletes others' comments

1. The moderator finds an inappropriate comment on the current article/subject/tool page.
2. The moderator deletes it.
3. The system validates the operation.
4. The system then shows the page with the corresponding comment deleted.

[AF02] Moderator deletes a root comment.

1. The system validates the operation.
2. The system then shows the page again but with the corresponding comment and comments replied to it deleted.

Post-conditions

UC007 Moderator Approve Article

Pre-conditions

The moderator has entered the pending article page.

Main Events Flow

1 The student/moderator clicks the approve button to edit it.

2 The system shows the article has been approved - [EF01]

Alternative Flows

None

Additional Information

Exception Flow

[EF01] Cancel or reject the confirmation

1. The system shows that the article has been processed by the other moderator.

Post-conditions

If the article is proved, it can be shown for students in the respective subjects or tools.

UC008 Moderator Reject Article

Pre-conditions

The moderator has entered the pending article page.

Main Events Flow

1 The student/moderator clicks the reject button to edit it.

4 The system shows the article has been rejected - [EF01]

Alternative Flows

None

Additional Information

Exception Flow

[EF01] Cancel or reject the confirmation

1. The system shows that the article has been processed by the other moderator.

Post-conditions

UC009 Moderator Get Pending Article list

Pre-conditions

The moderator has entered the subject page.

Main Events Flow

1 The moderator clicks the pending article button.

2 The system shows the list of articles waiting for the process.

Alternative Flows

None

Additional Information

Exception Flow

Post-conditions

1.2 Students View

UC101 Student Signs In

Pre-conditions

The student has a valid account on the system.

Main Events Flow

1. The instructor visits the sign-in page.
2. The instructor inputs the username and password.
3. The instructor confirms the login operation.
4. The system validates the account information. [AF01]
5. The system signs in the account and redirects to the home page.

Alternative Flows

[AF01] Account information is incorrect

1. The system detects that the account information provided by the user is incorrect.
2. The system displays a message showing that the username and/or password are wrong.
3. Go to the main event flow step 2.

Post-conditions

The moderator is signed in and redirected to the home page. The system stores a token which will be used for authentication dependent operations.

UC102 Student Views Enrolled Subject List

Pre-conditions

The student is signed in and is enrolled in the subjects to be shown on the page.

Main Event Flow

1. The student visits the home page.

Post-conditions

All subjects that the student is currently enrolled in will be shown on the home page.

UC103 Student Views Detailed Subject Information

Pre-condition

The student is signed in and on the home page, and the student is enrolled in the subject to be visited.

Main Event Flow

1. The student clicks the subject link.

Post-conditions

The student will be directed to the subject details page.

UC104 Student Deletes Comments

Pre-conditions

The student has a valid account on the system and already logged in. The student has at least one existing comment on the current article/ subject/ tool page.

Main Events Flow

1. The student locates his/her own comment on the current page.
2. The student deletes the comment. [AF01]
3. The system validates the operation.
4. The system then shows the page again but with the corresponding comment deleted.

Alternative Flows

[AF01] User deletes a root comment.

1. The system validates the operation.
2. The system then shows the page again but with the corresponding comment and comments replied to it deleted.

Post-conditions

1.3 Moderator/Student View

UC201 User Add Comments

Pre-conditions

The user has a valid account on the system and already logged in.

Main Events Flow

1. The user visits an article/ subject/ tool page.
2. The user finds a comment section at the bottom of the page.
3. The user adds a new comment and posts it.[AF01]
4. The system validates the user account information.
5. The system then shows the corresponding comment.

Alternative Flows

[AF01] User replies to someone's comment

1. The system validates the user account information.
2. The system then shows the corresponding comment with the information of whom it replies to.

Post-conditions

UC202Publish Article

Pre-conditions

The student/moderator has created the article and added tags.

Main Events Flow

1 The student/moderator clicks the 'publish' button to publish it.

2 System comes back to the initial page.

Alternative Flows

None

Additional Information

Exception Flow

Post-conditions

If the article is published by the moderator, the article will be shown in the respective subject or tools. If the article is published by the student, the article will be processed by the moderator and will be shown in the pending list for the moderator.

UC203 Create Article

Pre-conditions

The student/moderator has entered the subject or tool page.

Main Events Flow

1 The student/moderator clicks the ‘create’ button.

2 System displays a panel for the input of the article name.

3 The student/moderator types the title of the article.[EF1]

4 System displays a page for the input of the context and tags of the article.

Alternative Flows

None

Additional Information

Exception Flow

[EF01] Cancel or reject the confirmation

1. The student/moderator cancels or rejects the input of the article name, and the system will return to the initial page.

Post-conditions

The student/moderator clicks the ‘publish’ button.

UC204 Get Article

Pre-conditions

The student/moderator has entered the subject or tool page.

There are some articles about the subject or the tool.

Main Events Flow

1 The student/moderator clicks the title of the article.

2 System shows the article page.

Alternative Flows

None

Additional Information

Exception Flow

[EF01] Cancel or reject the confirmation

1. The student/moderator cancels or rejects the confirmation, and the system will not publish the article and return to the initial page.

Post-conditions

UC205 Edit Article

Pre-conditions

The student/moderator has entered an article page.

Main Events Flow

1 The student/moderator clicks the edit button to edit it.

2 The system presents the user with editable pages - [EF01]

Alternative Flows

1 The system shows the article cannot be edited.

Additional Information

Retrospectives

Add Retrospective

Title	Date	Participants
2020-10-05 Retrospective	05 Oct 2020	Kaung Khant Thu Yi Wang Patawat Watakul Yiyang JIN Gabriel Cheung Grant Boon Hwa Lim Jinxin TIAN
2020-08-31 Retrospective	31 Aug 2020	Matthew O'Halloran Grant Boon Hwa Lim Kaung Khant Thu Patawat Watakul Gabriel Cheung Jinxin TIAN Congran LI Yi Wang Xujia Zhang
2020-06-06 Retrospective	06 Jun 2020	Patawat Watakul Kaung Khant Thu Grant Boon Hwa Lim Congran LI Gabriel Cheung Jinxin TIAN Matthew O'Halloran Yiyang JIN Xujia Zhang @Yi Wang
2020-05-11 Retrospective	11 May 2020	Matthew O'Halloran

2020-06-06 Retrospective

Date	06 Jun 2020
Participants	Patawat Watakul Kaung Khant Thu Grant Boon Hwa Lim Congran LI Gabriel Cheung Jinxin TIAN Matthew O'Halloran Yiyang JIN Xujia Zhang @Yi Wang

Retrospective

What did we do well?

- Everyone attended every meeting
- *Can directly communicate to the responsible team member*
- All the team members are active and responsible for their job.
- Everyone have their own role and know their responsibility.
- Sub team role are well defined.

What should we have done better?

- Do sprint planning well in Jira before starting the sprint. Brainstorm more ideas to split the task.
- Create subteam meetings more to have better decisions
- Subteam need to communicate better.
- Share more ideas with client about the project decisions

Actions

2020-05-11 Retrospective

Date	11 May 2020
Participants	Matthew O'Halloran

Retrospective

What did we do well?

- Pivot off the back of updated prototype and feedback from the client
- Delegate into sub-teams for individuals to work on tasks
- Appointed team leads to manage workflow
- Have open discussions about where we saw that product
- Create a thorough digital prototype, reflected in the user stories

What should we have done better?

- Been more proactive in gathering requirements and designing goal model
- Have a clearer idea of our approach for the design
- Communicate frequently through slack

2020-08-31 Retrospective

Date	31 Aug 2020
Participants	Matthew O'Halloran Grant Boon Hwa Lim Kaung Khant Thu Patawat Watakul Gabriel Cheung Jinxin TIAN Congran LI Yi Wang Xujia Zhang

Retrospective

What did we do well?

- Worked well in sub-teams and sub-teams communicated well with each other
- Attended meetings
- Took minutes at all meetings
- Devops practice was allowing deployment and for people to view a live version

What should we have done better?

- Could have been better at updating documentation, mainly on Confluence
- Need a tool for communicating amongst teams. Maybe get everyone onto Postman account
- Improve API
- Could have been more proactive instead of waiting until last minute
- Didn't strictly follow review policy
- Git policy was not consistent

Actions

- [ ]

2020-10-05 Retrospective

Date	05 Oct 2020
Participants	Kaung Khant Thu Yi Wang Patawat Watakul Yiyang JIN Gabriel Cheung Grant Boon Hwa Lim Jinxin TIAN

Retrospective

What did we do well?

- Compare to previous sprint we use Jira better with sub task.
- Using git branches is better than previous sprint.
- Finish plan task on time efficiently.
- Most of the team member attended meetings.
- Good communication between sub teams.
- Good communication on slack.
- Good efficiency because we did both project and endeavour tasks
- Backend team communication to frontend is good
- Good api documentation

What should we have done better?

- Communication between backend and frontend.
- Didn't paid attention to documentation and confluence.
- Things get delay for a couple of time before starting the sprint (may be not too bad).
- Restarting. backend everytime when it crashes is bad.
- We should make agenda for meeting to be more productive.
- Testing was not emphasize enough leading to server crashes.

Actions

- [ ] *Schedule api meeting to finish the api*
- [ ]

Meetings

Google Drive folder containing all meeting notes: https://drive.google.com/open?id=1_uLrdbpx6dZ1t5xDZVvif55KDCDnWJfo

Client meeting

Page Description:	Record of client meetings
Page Status:	PROGRESS

Supervisor meeting

Page Description:	Record of supervisor meetings
Page Status:	PROGRESS

Team meeting

Page Description:	Record of team meetings
Page Status:	PROGRESS

Client meeting

Page Description:	Record of client meetings
Page Status:	PROGRESS

2020-03-13 Meeting notes

2020-03-30 Meeting notes

2020-06-01 Meeting notes

2020-09-01 Meeting notes

2020-03-13 Meeting notes

Date

13 Mar 2020

Attendees

- Patawat Watakul
- Grant Boon Hwa Lim
- Gabriel Cheung
- Kaung Khant Thu
- Yiyang JIN

Goals

- Make a schedule for the next client meeting with the client (Prof. Leon)
- Ask the question about the project (Background, Motivation, example, purpose, target users, related resources)
- Ask the target for this project

Discussion items

Time	Item	Who	Notes
3 mins	Requesting more detail on the SCORE competition	Everyone	• **Client Comment** • no longer relevant as University which sponsors the competition can not enter teams.

20 mins	General project background, target users, purpose and motivation	Everyone	**Client Comment (General idea)**
			• A lot of resources on the web, you can teach yourself but would be good if you can do better to **give guidance o**n what the tools to use are.
			• eg: if the client tells you to use flutter where do you learn about it? We need a repository of ideas
			• eg: undergrad capstone students were allowed to choose their own tools. but they had to find everything themselves, no resources that they could reference.
			• We don't want it to just be LMS though, how can we make it better?
			• no particular platform decided nor does he have any 'list of resources' to include
			• Should be more like canvas or LMS than confluence where we can put resources where students can find them
			• The program needs a way for people give feedback
			• needs to be able to remove/add content as required
			
			Question (by Kaung): If populated only by us/students, we might not be populating with the best resource as a google search might return a better tutorial?
			• Client response:
			  • this should be a start off point
			  • the problem when looking at Google is you aren't sure if the resource is decent
			  • The tool is a trusted source, a student can always look at Google
			  • eg: How to solve a problem in prolog was a book that had both good and bad examples.
			  • we might need a vetting procedure, QUALITY OVER QUANTITY
			• Client cares about there being a good quality place for people to look at.
			  • then I'll make a link to the repository for 90013
			  • Students can then give feedback on each of the 'resource' and admin can modify as necessary
			  • eg: "I am about to teach SWEN90013, I know students will lead technologies, do I need to do any updates to the repository, is there anything I need to look at? Do I need to make deletions /changes?
			  • it's NOT meant to be the definitive source, it should just be a starting point.
			
			Question: Do we need categorisation by subject?
			  • yes that's a good thing/suggestion but that might not always work because if the subject's really broad like the project subject then the lecturer may not be aware of all good resources.
			  • probably yea
			
			Question: Can we ask lecturers of each specific subject on suggestions for good resources to fill the site with?
			  • yes that's a good thing/suggestion but that might not always work because if the subject's really broad like the project subject then the lecturer may not be aware of all good resources.
			
			Client: In terms of what's written, is it clear what we need to do?

			• Question: Are we hosting resources ourselves or are we linking to external resources?
			  • eg: If there's a youtube video, surely you won't host it
			  • We want to improve stack overflow
			  • eg: by voting mechanism etc…
			  • Answer: license agreement? what would work best?
			
			Client comment: one thing that often happens is sometimes people don't have the pre-reqs when taking a subject so would be good if they could read up on resources beforehand as well
			
			Final comments: Client is only here Mon, Tues and Fri. Start brainstorming + build motivational model (once he talks about it in class) (next meeting should be week 4, Eduardo should be here too for this).

5 mins	Make a schedule for the next client meeting with the client (Prof. Leon)	Everyone	Next meeting 27/3 with following meeting 8/4

*** Note link: https://docs.google.com/document/d/1Wz376ghWyQCuDxd7yUtEBnx_xMCg_vCjUtI_N5N7X14/edit?usp=sharing

Action items

- [x] Create time schedule for next client meeting on Google Calendar

Date

13 Mar 2020

Attendees

- Patawat Watakul
- Grant Boon Hwa Lim
- Gabriel Cheung
- Kaung Khant Thu
- Yiyang JIN

Goals

- Make a schedule for the next client meeting with the client (Prof. Leon)
- Ask the question about the project (Background, Motivation, example, purpose, target users, related resources)
- Ask the target for this project

Discussion items

Time	Item	Who	Notes
3 mins	Requesting more detail on the SCORE competition	Everyone	• **Client Comment** • no longer relevant as University which sponsors the competition can not enter teams.

20 mins	General project background, target users, purpose and motivation	Everyone	**Client Comment (General idea)**
			
			• A lot of resources on the web, you can teach yourself but would be good if you can do better to **give guidance o**n what the tools to use are.
			• eg: if the client tells you to use flutter where do you learn about it? We need a repository of ideas
			• eg: undergrad capstone students were allowed to choose their own tools. but they had to find everything themselves, no resources that they could reference.
			• We don't want it to just be LMS though, how can we make it better?
			• no particular platform decided nor does he have any 'list of resources' to include
			• Should be more like canvas or LMS than confluence where we can put resources where students can find them
			• The program needs a way for people give feedback
			• needs to be able to remove/add content as required
			
			Question (by Kaung): If populated only by us/students, we might not be populating with the best resource as a google search might return a better tutorial?
			
			• Client response:
			  • this should be a start off point
			  • the problem when looking at Google is you aren't sure if the resource is decent
			  • The tool is a trusted source, a student can always look at Google
			  • eg: How to solve a problem in prolog was a book that had both good and bad examples.
			  • we might need a vetting procedure, QUALITY OVER QUANTITY

			• Client cares about there being a good quality place for people to look at.
			  • then I'll make a link to the repository for 90013
			  • Students can then give feedback on each of the 'resource' and admin can modify as necessary
			  • eg: "I am about to teach SWEN90013, I know students will lead technologies, do I need to do any updates to the repository, is there anything I need to look at? Do I need to make deletions /changes?
			  • it's NOT meant to be the definitive source, it should just be a starting point.
			
			Question: Do we need categorisation by subject?
			
			  • yes that's a good thing/suggestion but that might not always work because if the subject's really broad like the project subject then the lecturer may not be aware of all good resources.
			  • probably yea
			
			Question: Can we ask lecturers of each specific subject on suggestions for good resources to fill the site with?
			
			  • yes that's a good thing/suggestion but that might not always work because if the subject's really broad like the project subject then the lecturer may not be aware of all good resources.
			
			Client: In terms of what's written, is it clear what we need to do?
			
			  • Question: Are we hosting resources ourselves or are we linking to external resources?
			  • eg: If there's a youtube video, surely you won't host it
			  • We want to improve stack overflow
			  • eg: by voting mechanism etc…
			  • Answer: license agreement? what would work best?
			
			Client comment: one thing that often happens is sometimes people don't have the pre-reqs when taking a subject so would be good if they could read up on resources beforehand as well
			
			Final comments: Client is only here Mon, Tues and Fri. Start brainstorming + build motivational model (once he talks about it in class) (next meeting should be week 4, Eduardo should be here too for this).

5 mins	Make a schedule for the next client meeting with the client (Prof. Leon)	Everyone	Next meeting 27/3 with following meeting 8/4

*** Note link: https://docs.google.com/document/d/1Wz376ghWyQCuDxd7yUtEBnx_xMCg_vCjUtI_N5N7X14/edit?usp=sharing

Action items

- [ ] Create time schedule for next client meeting on Google Calendar

2020-03-30 Meeting notes

Date

30 Mar 2020

Attendees

- Matthew O'Halloran
- Patawat Watakul
- Grant Boon Hwa Lim
- Gabriel Cheung
- Kaung Khant Thu
- Yiyang JIN
- Xujia Zhang
- Jinxin TIAN
- Congran LI
- Yi Wang

Goals

- Make a schedule for the next client meeting with the client (Prof. Leon)
- Show our goal model to the client.
- Discuss our design idea with the client

Discussion items

Time	Item	Who	Notes
15 mins	Show our goal model to the client.	Matthew O'Halloran	**Client Comment** • Add the role of maintainer • Add functions to remove comments and resources. • Changed a few expressions
10 mins	Introduce the features of the stack share website so as to get some design idea	Kaung Khant Thu	**Client Comment (General idea)** ▪ We are not recreating the stack share ▪ we can borrow ideas from it

10 mins	Get feedback from the client	Everyone	On the project
			▪ Maintainer and crowd need new persona
			▪ Finding more important than ranking
			▪ Create a leader board
			▪ Resources need to be Comprehensive
			  ▪ give suggestions about where people can go.
			  ▪ e.g. The lecturer can give a guidance about what student need for the subject
			On the teamwork
			▪ Need to decide the role
			▪ Start to work on the user story/persona
			▪ Useful comment will be rewarded in teamwork

Date

30 Mar 2020

Attendees

- Matthew O'Halloran
- Patawat Watakul
- Grant Boon Hwa Lim
- Gabriel Cheung
- Kaung Khant Thu
- Yiyang JIN
- Xujia Zhang
- Jinxin TIAN

- Congran LI
- Yi Wang

Goals

- Make a schedule for the next client meeting with the client (Prof. Leon)
- Show our goal model to the client.
- Discuss our design idea with the client

Discussion items

Time	Item	Who	Notes
15 mins	Show our goal model to the client.	Matthew O'Halloran	**Client Comment** • Add the role of maintainer • Add functions to remove comments and resources. • Changed a few expressions
10 mins	Introduce the features of the stack share website so as to get some design idea	Kaung Khant Thu	**Client Comment (General idea)** ▪ We are not recreating the stack share ▪ we can borrow ideas from it

10 mins	Get feedback from the client	Everyone	On the project
			▪ Maintainer and crowd need new persona
			▪ Finding more important than ranking
			▪ Create a leader board
			▪ Resources need to be Comprehensive
			  ▪ give suggestions about where people can go.
			  ▪ e.g. The lecturer can give a guidance about what student need for the subject
			On the teamwork
			▪ Need to decide the role
			▪ Start to work on the user story/persona
			▪ Useful comment will be rewarded in teamwork

2020-06-01 Meeting notes

Date

08 May 2020

Attendees

- Patawat Watakul
- Gabriel Cheung
- Grant Boon Hwa Lim

Goals

- Make a confirmation on the digital prototype.

Discussion items

Time	Item	Who	Notes
10 min	- Moderator Version:	Gabriel Cheung	• Much better • Likes add section • Add section: • Add tutorials to add section page dropdown • How to add the pictures when adding the tools-Student • version: - Better than last time
10 min	Client thoughts:	Gabriel Cheung	• How risky are we prepared to be • Can we try to implement for undergrad next year • Would be a good way to get real feedback- Get Eduardo's feedback on the prototype • Need to estimate (by next Friday) is what we can get done by end of sprint 1 (refer to point above regarding getting undergrads to try it) • Would really like for this to happen- Level of testing if we get undergrads to test it will be higher
2 min	Discuss on personas	Grant Boon Hwa Lim	• He said this thing is not his responsibility and let ask the supervisor

Action items

- [ ]

2020-09-01 Meeting notes

Date

01 Sep 2020

Attendees

- Matthew O'Halloran
- Grant Boon Hwa Lim
- Gabriel Cheung
- Yi Wang
- Patawat Watakul
- Xujia Zhang

Goals

- Demonstrate current version to Leon
- Discuss feedback with the client

Discussion items

Time	Item	Who	Notes
10 mins	Show our currently deployed version	Matthew O'Halloran	
15 mins	Get feedback from the client	Everyone	• Likes the current progress • Would like a recommended articles section • Prioritize getting the student perspective • Should assign a person to proofread the English and spelling, focus on improving look and feel • search then suggesting articles in that order of priority • Considering handing over to 90082 next semester
10 mins	Demonstrating testing, backend documentation	Matthew O'Halloran	

Supervisor meeting

Page Description:	Record of supervisor meetings
Page Status:	PROGRESS

2020-03-10 Meeting notes

2020-03-17 Meeting notes

2020-03-24 Meeting notes

2020-03-31 Meeting notes

2020-04-07 Meeting notes

2020-04-28 Meeting notes

2020-08-07 Meeting notes

2020-08-14 Meeting notes

2020-09-11 Meeting notes

2020-09-25 Meeting notes

2020-03-10 Meeting notes

Date

10 Mar 2020 11:00 - 12:00

Attendees

- Patawat Watakul
- Grant Boon Hwa Lim
- Gabriel Cheung
- Kaung Khant Thu
- Matthew O'Halloran
- Yiyang JIN

Action items

- [ ] Ask the client for additional details about the project
- [ ] Send email to the client asking for additional resources then asking him to meet
- [ ] Ideally meet the client again before the end of week 6 to validate personas, user stories etc
- [ ] Set up the homepage with photos, preferred name, etc.

2020-03-17 Meeting notes

Present: Kaung, Gabriel, Patawat, YiYang, Matthew, Jinxin, YiWang

Apologies: Grant, Congran, XuJia

Date: 17/3

Previous meeting minutes: https://docs.google.com/document/d/1zl_22wwPHSyYEuHHVEGX1FFwAlljeTrBLuGZFt_3mso/edit

Notes:

Eduardo: Anything that we didn't quite understand with the previous client meeting?

- we weren't allowed to do the SCORE competition anymore

Eduardo: Do we know the goals of the current project? What we're designing?

- we should build a website/repository that contains all the tools that students in the uni might need to use
- kind of like w3 schools?
    - except more people can upload resources like students, tutors etc… but hasn't been clearly specified WHO can upload
    - if everyone uploads then there will be too much resources

Eduardo's understanding of the system:

- it's a repository for a bunch of things including coding and stuff
- My first question when listening to project description:
    - if this is a repository for documents etc… what's the difference between this and canvas?
    - W3 school has a goal of also teaching, here's source code, this will do this etc…
        - it's not only a repository but also teaches people how to code/improve their coding skills
    - so I asked, Leon, if we're doing that but it's for software engineers/technical people, I think we should have a repository that could also have something that is more specific to our audience
    - In my mind, when we see stack overflow, sometimes you can post an answer and your answer will be voted the right answer for the question. However, it could be the correct answer but not the best way to do it OR it could be the 'best way' but people don't understand how it works
    - So why not a repository where people could vote
        - eg: if I upload a source code, could we perhaps break down that answer to see how 'good' it is? Could we semantically analyse the code?
    - It's part of our job to do some research because even Leon isn't 100% sure what it should look like
    - more related to SWE principals

Matt: understanding isn't that we're making the tools but we're providing a platform for people to access the tools

- more crowdsourcing some of the tools and using people's experience to vote on how good those tools are

Kaung: https://www.producthunt.com/ shares a tool and people rank if it's good or not, maybe ours could be something like that?

Could use a roadmap as basis for structuring site?

- users click into each specific part of the roadmap and we have suggestions of tools to use for each step

Matt: product hunt style could be good

Eduardo: stackshare.io

- they have tools and tech that is crowd sourced
- so what's the goal of this project? because we shouldn't be doing something that already exists (like stackshare)

Matt: should we interview some external software engineers to see what ideas they have? what tools they use etc…? what they think would be good for them/useful for them

Eduardo: if we're designing this for ourselves, if we could integrate this to course instructor, students could recommend tools to next students

- we should address own problem at our school
- eg: we vote internally about tools we've used

There are massive discussions rn in regards to COVID-19 about what tools to have video conference, how to record lectures etc…

- if we had this repository/resource then everyone would be on the same page

Eduardo: maybe we could scope to school of engineering first then other schools can jump onboard if they want.

- we should design it so it can be extended to other domains
- needs to be modular

Right now, if someone jumps on board, if they jump on the confluence page, is there enough for them to know what is going on? is there enough for them to know what the project is about?

Eduardo: personally, if you do this inception phase really well and understand what this project is about then it'll help later on

TODO:

- send Leon a summary of what we understand about the system right now
    - ask for some verification/validation of sorts
    - pls make sure confluence is updated
    - **Deliverables:**
        - have something about the motivation of the project, differentials, contribution.
            - what is this project adding to the community
            - some personas
            - user stories
            - prototypes (to a certain degree, just paper prototypes or equivalent)

2020-03-24 Meeting notes

Date

24 Mar 2020

Attendees

- Patawat Watakul
- Congran LI
- Matthew O'Halloran
- Kaung Khant Thu
- Gabriel Cheung
- Grant Boon Hwa Lim
- Xujia Zhang
- Yiyang JIN

Role

- Patawat Watakul : Organise the meeting
- @Yiyang JIN: Do a minutes

Goals

- Update current work to the advisor
- Update the general group idea
- What plan our team want to do this week.
- Get other suggestions from the supervisor

Discussion items

Time	Item	Who	Notes

5 mins	• Update the current situation
	• Update several pages
	• Project scope
	• Design idea
	  • Product Hunt
	  • StackShare
		Matthew O'Halloran	• The supervisor updated the university situation(Covid-19)
			• The supervisor ask the question about the project propose and project idea
			• Matthew: Explained his idea which he already put on Confluence

25 mins	• What plans our team plan to do this week.	All	Supervisor: • He needed to get the goal model. • He suggested starting to use that tool for creating a goal model • The tool commented by Leon for do/feel • He noted that we need to finish the goal model my Thursday before sending it to Leon (Client) on Friday. • After we got confirmation from the client, we can assign the responsibility to each member.

Action items

- [ ] Identify the goal model of the project by using Who/Do/Be/Feel tool
- [ ] Assign the role after finishing the goal model.

Date

24 Mar 2020

Attendees

- Patawat Watakul
- Congran LI
- Matthew O'Halloran
- Kaung Khant Thu
- Gabriel Cheung

- Grant Boon Hwa Lim
- Xujia Zhang

Role

- Patawat Watakul : Organise the meeting
- @Yiyang JIN: Do a minutes

Goals

- Update current work to the advisor
- Update the general group idea
- What plan our team want to do this week.
- Get other suggestions from the supervisor

Discussion items

Time	Item	Who	Notes

5 mins	• Update the current situation
	• Update several pages
	• Project scope
	• Design idea
	  • Product Hunt
	  • StackShare
		Matthew O'Halloran	• The supervisor updated the university situation(Covid-19)
			• The supervisor ask the question about the project propose and project idea
			• Matthew: Explained his idea which he already put on Confluence

25 mins	• What plans our team plan to do this week.	All	Supervisor: • He needed to get the goal model. • He suggested starting to use that tool for creating a goal model • The tool commented by Leon for do/feel • He noted that we need to finish the goal model my Thursday before sending it to Leon (Client) on Friday. • After we got confirmation from the client, we can assign the responsibility to each member.

Action items

- [ ] Identify the goal model of the project by using Who/Do/Be/Feel tool
- [ ] Assign the role after finishing the goal model.

2020-03-31 Meeting notes

Date

31 Mar 2020

Attendees

- Patawat Watakul
- Kaung Khant Thu
- Yiyang JIN
- Matthew O'Halloran
- Gabriel Cheung
- Grant Boon Hwa Lim
- Yi Wang
- Xujia Zhang
- Congran LI

Goals

- Update the team progress
- Get suggestion from the supervisor about user story and personas
- Get suggestion about the team roles

Discussion items

Time	Item	Who	Notes
5 mins	Update client’s meeting with supervisor	@All	
5 mins	Get suggestion about the team roles	@All	**Supervisor's comments:** ▪ It's for hard to keep roles for current phase, ▪ can have basic roles for now, ▪ and split into sub-teams to complete task

10 mins	Discuss user story and personas
	• Do you have a suggestion for splitting the work?
	• What do you recommend to do with it?
		@All	**Works to do:**
			▪ finalise goal model, validate with the client
			▪ user story
			▪ persona
			▪ testcases
			▪ prototype
			Supervisor's comments:
			▪ rework the goal model to keep the following task consistent.
			▪ come up with a agenda of all tasks
			  ▪ assignment
			  ▪ due
			▪ collaborate across task
			▪ get familiar with the tools e.g. Jira
			▪ keep updating pages in confluence

2020-04-07 Meeting notes

Date

07 Mar 2020

Attendees

- Patawat Watakul
- Kaung Khant Thu
- Yiyang JIN
- Matthew O'Halloran
- Gabriel Cheung
- Grant Boon Hwa Lim
- Yi Wang
- Xujia Zhang
- Congran LI
- Jinxin TIAN

Goals

- Update the team progress
- Show our current user story to the supervisor
- Get feedback about our current works on the projects

Discussion items

Time	Item	Who	Notes
5 mins	Update client's meeting with supervisor	@All	
10 mins	Show our current goal model to the supervisor	@All	1. Still need to reword a bit e.g. "learn/ find new resource and tutorials" 2. The "post resource" can be separate as a new leaf 3. Link everything together a. Number the goal model and create user story according to the goal model b. if we have existing personas, we can use the name in persona to express roles in the user story 4. Need to confirm functionalities with the client
10 mins	Discuss about the prototype	@All	1. When we show the prototype to the client, we should let the client explore the prototype himself with our instructions 2. It is recommended that we do the prototype test for more times a. we can show the prototype to the supervisor fist b. try 3 person

Who	Do	Be	Feel
	• Moderate the quality of resource • Vote on the quality of resource • Check Leaderboard for the most participated student • Login using university credentials		• Accomplished

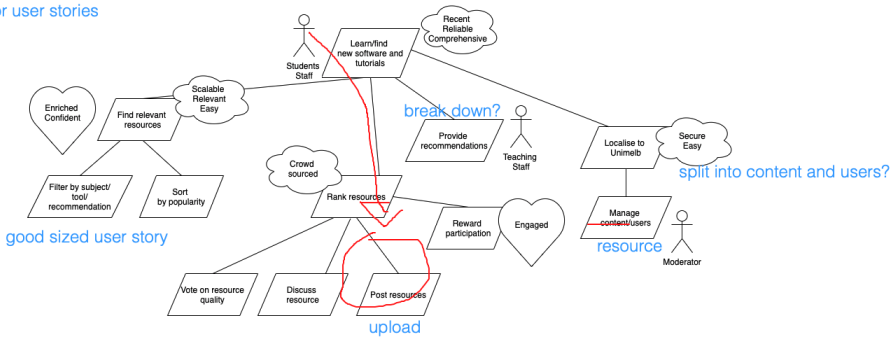
Goal Model

Too large for a user story

Link: https://drive.google.com/file/d/108Y5PuJ2tclf1SDxgYo2qmuR8iQRIupk/view?usp=sharing

Refer to INVEST for user stories

number
each leaf
for user story



break down?

split into content and users?

good sized user story

resource

upload

Ideas:

ideas about different roles:

- moderator: manage the content of the website: e.g. delete resources, ban users, etc.
- administrator: have full control of the entire website. assign roles
- lecturer and tutor: acknowledged/rank/post resources (guarantee the resources are reliable)

2020-04-28 Meeting notes

Date

28 Apr 2020

Attendees

- Kaung Khant Thu
- Patawat Watakul
- Gabriel Cheung
- Yiyang JIN
- Congran LI
- Jinxin TIAN
- Yi Wang
- Matthew O'Halloran
- Xujia Zhang

Discussion items

Inception deadline is on Thursday

Tips from Supervisor

When researching architecture

Research on

Architecture pattern -> layer pattern -> security pattern -> behaviour and structural pattern,define data or entity relation diagram

Focus working on these **diagrams**

- Domain diagram
- High level architecture diagram for all for client and sever
- Class diagramm
- Sequence diagram

Note from supervisor

Pattern

1. Presentation layer patterns
2. session layer patterns
3. security patterns

Research on different type of technology

Frontend(Example react)

Backend(Node js)

Database technology(SQL or NoSQL)

Team should split the team into different role and research separetly

Action items

- [x] Kaung Khant Thu Patawat Watakul Gabriel Cheung work on prototype till thursday and confirm with client
- [x] Kaung Khant Thu organize meeting with client
- [ ] Yiyang JIN will work on Jira task
- [ ] Matthew O'Halloran will review the quality of resources on confluence and update
- [x] Team will work on researching technology architecture
- [x] Team will select role

2020-08-07 Meeting notes

Date

07 Aug 2020

Attendees

- Patawat Watakul
- Matthew O'Halloran
- Grant Boon Hwa Lim
- Yiyang JIN
- @yi wang
- Kaung Khant Thu
- Jinxin TIAN
- Xujia Zhang
- Congran LI

Goals

- Feedback the first sprit and the presentation

Discussion items

Time	Item	Who	Notes
5 mins	• Feedback the first sprit and the presentation	Everyone	• Edouardo will look to the feedback for our team because he has not seen every detail of the mark for the first sprit.
10 mins	Discuss about the Sprit Plan and Last Sprit Retrospective.	Everyone	

10 mins	Discuss about the roles
	• The quality assurance
	• Tester
	• DevOps
			• He suggests ous assign the responsibility in each role
			  • Who takes responsibility for contact with Leon
			  • Who takes responsibility to do QA in Sprit 1

2 mins	Update the document		
4 mins	Endeavour		• Need to submit the team form • It's mandatory.

Action items

- [ ] Make a meeting schedule with the client (Prof. Leon).

2020-08-14 Meeting notes

Date

14 Aug 2020

Attendees

- Eduardo Araujo Oliveira (Supervisor)
- Matthew O'Halloran (Team Leader)
- Congran LI
- Gabriel Cheung
- Grant Boon Hwa Lim
- Jinxin TIAN
- Kaung Khant Thu
- Patawat Watakul
- Xujia Zhang
- Yiyang JIN (Note taking)
- Yi Wang

Goals

- Feedback the second weeks improvement
- Dev-op

Discussion items

Time	Item	Who	Notes
10 mins	Update the progress of front-end and back-end ▪ Split user stories as tasks on both frontend and backend	Everyone	• Create a task as a user stories in Jira, and create **subtasks** relate to front-end and back-end respectively. ☑ Need to be consistent with user journey • Consistency with what we agreed with Leon ☐ Needs further estimation • Jira is a lot better
10 mins	Get feedbacks about what need to be improved.	Everyone	• Need a set of consistent protocols on branch&features in Github ☐ team members need to know how to create name a task in Github ☐ In sprint2, there will be a release on Github and the features should be able to link to the task on Jira • Need to have a proper review process ☐ Need to be clear and formal ☐ Reviewers should and should be able to leave comments. And the reviews should be documented. ☐ The team should be able to access and validate the reviews

10 mins	Get suggestions about the markdown function	Everyone	• Get feedback from the client about other options
			  • Explain the trade-offs to the client
			  • Mind the time

Action items

- ☐ Confirm with Leon our plan on markdown
- ☐ Discuss the issues on Quality(git branch&feature, review protocol, etc)

2020-09-11 Meeting notes

Attendees

- Patawat Watakul
- Matthew O'Halloran
- Grant Boon Hwa Lim
- Yiyang JIN
- @yi wang
- Kaung Khant Thu
- Jinxin TIAN
- Xujia Zhang
- Congran LI

Meeting Notes:

- Discussed feedback for previous sprint
    - Overall no tests were provided and no coverage information was shown. Should have load testing as well.
    - Specific feedback will be sent out via email later.
    - Client is happy with current progress.

- Stand up update:
    - Backend api for pending articles and block article on edit is on track
    - Backend has finished search functionality for backend and is under review
    - Comment section is in progress
    - Overall not much change since Monday standup

2020-09-25 Meeting notes

Date

25 Sep 2020

Attendees

- Kaung Khant Thu
- Patawat Watakul
- Grant Boon Hwa Lim
- Gabriel Cheung
- Congran LI
- Xujia Zhang
- Jinxin TIAN
- Yi Wang
- Yiyang JIN
- Eduardo Araujo Oliveira

Goals

- Progress review on whether we can finish our task for this sprint and any one need resources.

Discussion items

Time	Item	Who	Notes
10	Jira progress and testing overview	Eduardo	• Jira progress • Work on test case with sample user • Write test case with student and moderator view combine • Block article from editing status
5	Progress on Block article	Grant Boon Hwa Lim	Not sure which api to call for to block article
5	Comment	Eduardo Araujo Oliveira	Comment status,
5	Comment	Congran LI	Backend don't comment on article part. Backend only have comment in section.
10	Suggestion	Eduardo Araujo Oliveira	Make github branches based on Jira story. Test cases to be link with Jira story.

Team meeting

Page Description:	Record of team meetings
Page Status:	PROGRESS

2020-03-21 Meeting notes

2020-03-28 Meeting notes

2020-04-04 Meeting notes

2020-04-11 Meeting notes

2020-05-02 Meeting notes

2020-05-18 Meeting notes

2020-06-02 Meeting notes

2020-08-09 Meeting notes

2020-08-15 Meeting notes

2020-08-31 Meeting notes

2020-09-09 Meeting notes

2020-09-16 Meeting notes

2020-10-05 Meeting notes

2020-10-12 Meeting notes

2020-03-21 Meeting notes

Date

21 Mar 2020

Attendees

- Xujia Zhang
- Grant Boon Hwa Lim
- Congran LI
- Kaung Khant Thu
- Patawat Watakul
- Matthew O'Halloran
- Yi Wang
- Xujia Zhang
- Jinxin TIAN
- @Yiyang JIN

Goals

- Make sure all members have an understanding over the project.
- Assign roles for the meetings

Discussion items

Time	Item	Who	Notes
10 mins	Project Scope	All Members	• Centralise the resources
10 mins	Confluence	All Members	• Clarify project scope • Update any new ideas • Update some design ideas
5 mins	Roles	All Members	• Patawat to be facilitator/agenda for supervisor and team meeting • Joyce to be minutes taker for supervisor and team meeting

Notes

We should send an email to Leon about our ideas/general understanding before the meeting

In client meeting, we can talk about what technology we've thought about and see what he thinks.

We should make a goal model after the lecture next week

Before meeting with Leon we should have an agenda

What exactly do we envision the project to be?

- Matt: there are a lot of students and tools in CS department. Want some way to centralise it so that there is no need to keep choosing different tools and lecturers/tutors can know what tools students are using.
    - hopefully design so we can extend to diff departments later on
    - starts off quite basic but as it grows it can be crowd sourced
- Kaung: same ideas as Matt
    - something extra though: apart from that, also think that he wants some videos there (tutorials)
- Gabriel: same as above
- Patawat: same
- Grant: same
- Everyone has the same idea more or less

Kaung: for the meetings, we didn't really have a structure.

- roles for the meetings should be assigned starting from next meeting onwards

Matthew will be in charge of sending emails to Leon

Is there anything to do in confluence?

- maybe clarify project scope
- as we iterate/come up with new ideas update it (highlight red as it changes so we can see how it has changed over time)as
- could also have some design ideascould

Decided to go with confluence order for meeting facilitator

- Patawat to be facilitator/agenda for supervisor and team meeting
- Joyce to be minutes taker for supervisor and team meeting

TODO

- [ ] Write agenda for client meeting
- [ ] Update ideas on confluence

Date

21 Mar 2020

Attendees

- Xujia Zhang
- Grant Boon Hwa Lim
- Congran LI
- Kaung Khant Thu
- Patawat Watakul
- Matthew O'Halloran
- Yi Wang
- Xujia Zhang
- Jinxin TIAN
- @Yiyang JIN

Goals

- Make sure all members have an understanding over the project.
- Assign roles for the meetings

Discussion items

Time	Item	Who	Notes
10 mins	Project Scope	All Members	• Centralise the resources
10 mins	Confluence	All Members	• Clarify project scope • Update any new ideas • Update some design ideas
5 mins	Roles	All Members	• Patawat to be facilitator/agenda for supervisor and team meeting • Joyce to be minutes taker for supervisor and team meeting

Notes

We should send an email to Leon about our ideas/general understanding before the meeting

In client meeting, we can talk about what technology we’ve thought about and see what he thinks.

We should make a goal model after the lecture next week

Before meeting with Leon we should have an agenda

What exactly do we envision the project to be?

- Matt: there are a lot of students and tools in CS department. Want some way to centralise it so that there is no need to keep choosing different tools and lecturers/tutors can know what tools students are using.
    - hopefully design so we can extend to diff departments later on
    - starts off quite basic but as it grows it can be crowd sourced
- Kaung: same ideas as Matt
    - something extra though: apart from that, also think that he wants some videos there (tutorials)
- Gabriel: same as above
- Patawat: same
- Grant: same
- Everyone has the same idea more or less

Kaung: for the meetings, we didn't really have a structure.

- roles for the meetings should be assigned starting from next meeting onwards

Matthew will be in charge of sending emails to Leon

Is there anything to do in confluence?

- maybe clarify project scope
- as we iterate/come up with new ideas update it (highlight red as it changes so we can see how it has changed over time)as
- could also have some design ideascould

Decided to go with confluence order for meeting facilitator

- Patawat to be facilitator/agenda for supervisor and team meeting
- Joyce to be minutes taker for supervisor and team meeting

TODO

- [ ] Write agenda for client meeting
- [ ] Update ideas on confluence

2020-03-28 Meeting notes

Date

28 Mar 2020

Attendees

- Patawat Watakul
- Congran LI
- Matthew O'Halloran
- Kaung Khant Thu
- Grant Boon Hwa Lim
- Gabriel Cheung
- Yiyang JIN
- Congran LI
- Yi Wang

Goals

- Finalise the goal model for sending it to Leon
- Arrange the client meeting and Team meeting on Monday
- Discuss the agenda of the client meeting on Monday.

Discussion items

Time	Item	Who	Notes
10 mins	Goal Model	Matthew O'Halloran	• Matthew explained overall of the goal model • Other member commented on the goal model • Kaung Khant Thu want to read it again and he will put further idea by tonight • Matthew will send the model to the client by tomorrow • Someone notes that the issue on the model and we must send the issue to Leon to figure
3 mins	Show the Sharestack's functions	Kaung Khant Thu	
3 mins	Discuss the agenda of client meeting on Monday	@all	• Get the feedback of goal model from the client • Kaung Khant Thu said he wants to ask other feature which the client possible need

Action items

- [ ] Send the goal model to the client by Sunday
- [ ] Create an agenda for a client meeting

Date

28 Mar 2020

Attendees

- Patawat Watakul
- Congran LI
- Matthew O'Halloran
- Kaung Khant Thu
- Grant Boon Hwa Lim
- Gabriel Cheung
- Yiyang JIN
- Congran LI
- Yi Wang

Goals

- Finalise the goal model for sending it to Leon

- Arrange the client meeting and Team meeting on Monday
- Discuss the agenda of the client meeting on Monday.

Discussion items

Time	Item	Who	Notes
10 mins	Goal Model	Matthew O'Halloran	• Matthew explained overall of the goal model • Other member commented on the goal model • Kaung Khant Thu want to read it again and he will put further idea by tonight • Matthew will send the model to the client by tomorrow • Someone notes that the issue on the model and we must send the issue to Leon to figure
3 mins	Show the Sharestack's functions	Kaung Khant Thu	
3 mins	Discuss the agenda of client meeting on Monday	@all	• Get the feedback of goal model from the client • Kaung Khant Thu said he wants to ask other feature which the client possible need

Action items

- [ ] Send the goal model to the client by Sunday
- [ ] Create an agenda for a client meeting

2020-04-04 Meeting notes

Date

04 Apr 2020

Attendees

Goals

- Finalise goal model before handing to the client on the next client meeting.
- Review current user story
- Confirm/update the schedule of our tasks
- Discuss the agenda for next client meetings

Discussion items

Time	Item	Who	Notes
15 mins	Goal Model	@All	
10 mins	Current version of user Story	@All	Went through the current user story together ▪ made some modification ▪ put an epic column beside the table ▪ discussed the expected granularity of each user story ▪ discussed the structure of the user story
5 mins	Schedule the upcoming tasks		▪ Goal model by thursday ▪ User story by sunday ▪ Review on Goal model on Monday after meeting ▪ Review on User story on Monday after ▪ Persona saturday ▪ Test case by next thursday ▪ Wireframe by next saturday ▪ Prototype by third week wednesday

Action items

- [x] Goal model by thursday
- [x] User story by Sunday
- [x] Personas by Saturday
- [x] Review on Goal model on Monday after meeting
- [x] Review on User story on Monday after
- [ ] Test case by next thursday
- [ ] Wireframe by next saturday
- [ ] Prototype by third week wednesday

Date

04 Apr 2020

Attendees

Goals

- Finalise goal model before handing to the client on the next client meeting.
- Review current user story
- Confirm/update the schedule of our tasks
- Discuss the agenda for next client meetings

Discussion items

Time	Item	Who	Notes
15 mins	Goal Model	@All	
10 mins	Current version of user Story	@All	Went through the current user story together ▪ made some modification ▪ put an epic column beside the table ▪ discussed the expected granularity of each user story ▪ discussed the structure of the user story
5 mins	Schedule the upcoming tasks		▪ Goal model by thursday ▪ User story by sunday ▪ Review on Goal model on Monday after meeting ▪ Review on User story on Monday after ▪ Persona saturday ▪ Test case by next thursday ▪ Wireframe by next saturday ▪ Prototype by third week wednesday

Action items

- [x] Goal model by thursday
- [x] User story by Sunday
- [x] Personas by Saturday
- [x] Review on Goal model on Monday after meeting
- [x] Review on User story on Monday after
- [ ] Test case by next thursday
- [ ] Wireframe by next saturday
- [ ] Prototype by third week wednesday

2020-04-11 Meeting notes

Attendees

- Grant Boon Hwa Lim
- Jinxin TIAN
- Gabriel Cheung
- Yiyang JIN
- Patawat Watakul
- Yi Wang
- Xujia Zhang
- Congran LI

Goals

- Walk through current prototype
- Get suggestions for improvement

Discussion items

Time	Item	Who	Notes
10 mins	Demonstration of prototype	Gabriel Cheung	
3 mins	Brainstorming and suggestions	@all	• add navigation bar • add subscription functionality to user stories
3 mins	Discuss future meetings	@all	
8 mins	Discuss Acceptance Criteria		

- [ ] Complete prototype with additional pages
- [ ] modify acceptance criteria

2020-05-02 Meeting notes

Date

02 May 2020

Attendees

- Kaung Khant Thu
- Grant Boon Hwa Lim
- Jinxin TIAN
- Yiyang JIN
- Matthew O'Halloran
- Congran LI
- Xujia Zhang
- Yi Wang

Goals

- Decide Roles in the team
- Decide subteam lead
- Architecture
- Split task
- Team Process
- Follow up Decision on the current state of the project based on client feedback
- Next Week Agenda
- Get a Summary

Discussion items

Time	Item	Who	Notes
15min	Decide Roles in the team		Scrum Master role • Matthew O'Halloran • Kaung Khant Thu Frontend • Grant Boon Hwa Lim • Congran LI • Jinxin TIAN • Patawat Watakul Backend • Xujia Zhang • Yiyang JIN • Yi Wang • Gabriel Cheung Devops Kaung Khant Thu Matthew O'Halloran
10min		Yiyang JIN	Backend lead Yiyang JIN Frontend lead Yiyang JIN Devops lead Kaung Khant Thu Scrum Master Matthew O'Halloran
10min	Technology to used in Architecture		Backend is node.js ,express Frontend is react Devops repo is github One repo for both frontend and backend
15min	Jira		Jira type is now a kanban board and cannot create a board at the moment Task are already created and need to be moved

5min	Next Week Agenda		Next team meeting will be on Tuesday after Supervisor Meeting

Action items

- [ ] Subteams will work separtely to come up with architecture and technology discussion

2020-05-18 Meeting notes

Date

18 May 2020

Attendees

- Yi Wang
- Jinxin TIAN
- Patawat Watakul
- Matthew O'Halloran
- Yiyang JIN
- Kaung Khant Thu
- Congran LI
- Grant Boon Hwa Lim
- Xujia Zhang
- Gabriel Cheung

Goals

- List current issues
- Go through the slides
- Modify slides for the first presentation

Discussion items

Time	Item	Who	Notes

20 min	Presentation	@all	• Eduardo gives use some feedback about the slides:
			  • Too many slides
			  • Polish slides
			  • Application architecture is too general

15 min	Current Issues	@all	• The uni is unlikely to give us access to uni API • Which testing tool should the team use? • Issue with the deployment, is heroku a suitable tool to use with Git?

Action items

- [ ]

2020-06-02 Meeting notes

Date

03 Jun 2020

Attendees

- Patawat Watakul
- Gabriel Cheung
- Grant Boon Hwa Lim
- Matthew O'Halloran
- Kaung Khant Thu
- @Yi Wang
- Congran LI

Goals

- Update each member tasks
- Discussed on feedback from supervisor

Discussion items

Time	Item	Who	Notes
2 min	Plan before the end of the sprit 1 and make a schedule for sprit review and retrospective	Matthew O'Halloran	• Schedule the retrospective on Saturday
3 min	Discuss the progress on DevOps and deployment issue	Kaung Khant Thu	
5 min	Discuss the issue on backend and frontend	Patawat Watakul	

Action items

- [ ]

2020-08-09 Meeting notes

Date

09 Aug 2020

Attendees

- Patawat Watakul
- Matthew O'Halloran
- Congran LI
- Grant Boon Hwa Lim
- Xujia Zhang
- Kaung Khant Thu
- Gabriel Cheung
- Jinxin TIAN
- Yiyang JIN

Goals

- Assign the roles for each member
    - Who takes responsibility to contact with Leon
    - Who takes responsibility for Endeavour
        - Have to submit something
    - Who takes responsibility for QA
- Sprint planning
- What tasks we have to do in this sprint
- Make a schedule for our team
- Discuss on Endeavour

Discussion items

Time	Item	Who	Notes
30 mins	Sprit Planning for sprint 2	all	• Research for editor tool(How to apply markdown library to the project) • We decided to work on the editing page. • We decided to work on the pending page.
10	Assign the role for a team member	all	• Patawat Watakul takes front end leader. • Kaung Khant Thu takes responsibility for DevOps. • Kaung Khant Thu takes responsibility for Back end leader. • Grant Boon Hwa Lim takes responsibility for QA • Matthew O'Halloran still takes responsibility to contact with Leon and also do the sprint leader.
5 min	Discuss on Endeavour	all	Matthew O'Halloran takes responsibility to look on Endeavour
10 min	Evaluate the score for each task	all	

Action items

- [ ] Regist the endeavour

2020-08-15 Meeting notes

Date

15 Aug 2020

Attendees

- Patawat Watakul
- Matthew O'Halloran
- Congran LI
- Grant Boon Hwa Lim
- Xujia Zhang
- Kaung Khant Thu
- Gabriel Cheung
- Jinxin TIAN
- Yiyang JIN
- Yi Wang

Goals

- Discuss about our sprint plan adjustment.
- Discuss the issue of markdown.
- Discuss our code review strategy.
- Discuss git update protocols.

Discussion items

Time	Item	Who	Notes
5 mins	Sprint update for sprint 2	all	• Frontend and Backend update the new plan and update on jira
5 mins	Discuss the issue of markdown.	all	• Confirm with the client about our plan
5mins	Discuss student's subject	all	▪ Student have access to only enrolled subjects or all subjects? ▪ Contact with the client for opinions
5 min	Discuss our code review strategy.	all	▪ Backend and frontend review code within the group ▪ the API test should be include in frontend test already
10 min	Discuss git update protocols.	all	▪ Create git branch for Jira task

Action items

- [x] Contact with Leon

2020-08-31 Meeting notes

Date

15 Aug 2020

Attendees

- Patawat Watakul
- Matthew O'Halloran
- Congran LI
- Grant Boon Hwa Lim
- Xujia Zhang
- Kaung Khant Thu
- Gabriel Cheung
- Jinxin TIAN
- Yiyang JIN
- Yi Wang

Goals

- Practice running through demonstration of deployed version
- Discuss priorities for next sprint
- Segue into sprint retrospective

Discussion items

Time	Item	Who	Notes
20m	Run through demo with Eduardo	all	• Prepare to have to guide Leon through in the same way
20m	Discuss feedback	all	• Thinks we should focus on search functionality • should provide article viewing functionality for students without logging in • what page should be displayed without logging in

Action items

2020-09-09 Meeting notes

Attendees

- Matthew O'Halloran
- Grant Boon Hwa Lim
- Xujia Zhang
- Kaung Khant Thu
- Gabriel Cheung
- Jinxin TIAN
- Yiyang JIN
- Yi Wang

Meeting Notes:

- No blockers at the moment
- Fixed bug for tags where tags weren't being saved properly

2020-09-16 Meeting notes

Attendees

- Patawat Watakul
- Yiyang JIN
- Grant Boon Hwa Lim
- Gabriel Cheung
- Xujia Zhang
- Kaung Khant Thu
- Yi Wang

Goals

- Discuss Endeavour
- Discuss testing

Meeting Notes:

- Find material about the Endeavour.
- Frontend and backend each have one person to test.
- Finish the risk assessment form.

2020-10-05 Meeting notes

Date

05 Oct 2020

Attendees

- Kaung Khant Thu
- Patawat Watakul
- Yiyang JIN
- Yi Wang
- Gabriel Cheung
- Grant Boon Hwa Lim
- Jinxin TIAN

Goals

- Presentation for endeavour (form a team)
- Acceptance testing item user journey and presentation to client (testing form team)
- Check confluence for update
- Retrospective for previous sprint (discuss)
- Sprint planning (user story planning form team)
- final project delivery plan (discuss)
- Testing plan? (for final deliverable)
- Create release tag [pdf document of confluence,our frontend link]

Action items

- [ ] Presentation for endeavour [**Patawat**,**Yiyang Jin**,Matt,congran,**gabriel**]
- [ ] Acceptance testing item user journey and presentation to client[**Matt,Xuijia,Yi wang**]
- [ ] Sprint planning team[**Kaung,Grant,jinxin**]
- [ ] Check confluence for update[**Congran**]
- [ ] Create release tag [**Kaung**]
- [ ] Ask edurado for presentation slides about releasing the project.

2020-10-12 Meeting notes

Date

12 Oct 2020

Attendees

- Kaung Khant Thu
- Gabriel Cheung
- Xujia Zhang
- Matthew O'Halloran
- Eduardo Araujo Oliveira
- Yiyang JIN
- Grant Boon Hwa Lim
- Yi Wang
- Congran LI

Goals

Discussion items

Time	Item	Who	Notes
15min	Suggestion	Eduardo	• Can we come up with simple login? • How can people see subject • How to subscribe subject. • Team should prioritise the task to align with client. • Work on admin functions to see the overview of the system. • Create doc folder on github and put confluence item on github. • Test folder for testing documents and test results. • Write readme file on how to deploy.
15min	Client suggestion	Matthew O'Halloran	Mention about client suggestion.

Testing

API Testing

Test Data

Base Url = 167.172.11.217

Moderator Login Information(M1)

"account" : "kaung",
"password" : 1234

Student Login Information(S1)

"account" : "student",
"password" : 123

Post man test Script list

https://www.getpostman.com/collections/df8a966fc81e0ea5a30c

ID	Test Case	Feature	Api	Precondition	Input Parameter	Expected Test Result	Test Result
T1	Moderator Should be able to Login with correct account information	Login	{{url}}/login method:POST	Moderator account should be created.	"account" : "kaung", "password" : 1234	Api should return a success result with Api token.	Pass
T1b	Student Should be able to login.	Login	{{url}}/login method:POST	Student account should be created	"account" : "student", "password" : 123	Api should return a success result with Api token.	Pass
T2a	Moderator should get home page information.	Home	{{url}}/user/home method:GET	Moderator should be already login. **M1** login information is used.	**Auth token from login information in the header.**	Api should return success status. Along with subscribed subjects and topics.	Pass
T2b	Student should get home page information.	Home	{{url}}/user/home method:GET	Student should be already login. **S1** login information is used.	**Auth token from login information in the header.**	Api should return success status. Along with subscribed subjects and topics.	Pass
T3a	Moderator should be able to publish a new article.	Article Publish	{{url}}/article/publish method:POST	Moderator should be already login. **M1** login information is used. The article title should not be already exists in the database	**Auth token from login information in the header.** **Body** title :{{$randomCompanyName}} (randomly generated title) tags:["SWEN90009","java","hello"] content: content_test subjects:["SWEN90009"] tools:["Java"]	Api should return success status and article id of the published article	Pass
T3b	Student should be able to publish a article which is going to become a pending article	Article Publish	{{url}}/article/publish method:POST	Student should be already login. **S1** login information is used. The article title should not be already exists in the database	**Auth token from login information in the header.** **Body** title :{{$randomCompanyName}} (randomly generated title) tags:["SWEN90009","java","hello"] content: content_test subjects:["SWEN90009"] tools:["Java"]	Api should return success status and article id of the pending article	Pass
T4a	Moderator should be able to get an article which is already exists in the database.	Article Get	{{url}}/article/get/{{:id}} method:GET	Moderator should be already login. **M1** login information is used.	**Auth token from login information in the header.** article_id: 5f70aa34e98af700132e669a	Api should return success status and article detail information,	Pass

T4b	Student should be able to get an article which is already exists in the database.	Article Get	{{url}}/article/get/{{:id}} method:GET	Moderator should be already login. **M1** login information is used.	**Auth token from login information in the header.** article_id: 5f70aa34e98af700132e669a	Api should return success status and article detail information.	Pass
T5a	Moderator should be able to get a pending article to get information	Pending Article Get	{{url}}/article /getPendingArticle/ {{id}} method:GET	Moderator should be already login. **M1** login information is used.	**Auth token from login information in the header.** pending_article_id: 5f70afdae98af700132e669e	Api should return success status and article detail information.	Pass
T5b	Moderator should be able to get	Article Delete	{{url}}//article/delete/{{ article_id}} method:GET	Moderator should be already login. M1 login information is used.	**Auth token from login information in the header.** article_id: 5f70aa34e98af700132e669a	Api should return success.	Fail 404 not found error.
T5b	Moderator should be able to edit an article.	Article Edit					
T6	Moderator should be able to approve a pending article	Pending Article	{{url}}/article/pending Article/approve/: pending_article_id method:GET	Moderator should be already login. M1 login information is used.	**Auth token from login information in the header.** pending_article_id: 5f716c45a a0c5a0013b1670f	Produce a new article id. Previous pending article should be deleted A new Article should be able to get from Get Article api.	pass
T7	Moderator should be able to reject a pending article	Pending Article Reject	{{url}}/article /pendingArticle/reject/ {{pending_article_id}} method:GET	Moderator should be already login. M1 login information is used. student pending article id	Auth token from login information in the header. pending article id:5f716df4aa0 c5a0013b16714	The pending article should be deleted .	Pass
T8	Moderator should be able to get all pending articles names of a subject or a tool	Pending Article get all articles based on subject or tool name	{{url}}/article /getNamesOfArticles/: subject_code_or_tool _name method:GET (what is the difference between subject get and this api?)	Moderator should be already login. M1 login information is used.	Auth token from login information in the header. subject_code_or_tool_name: SWEN90009	Api should return all the article related to the subject code.	Fail Api doesn't contain article list.

T9	Moderator should be able to add subjects.	Subject
		
		(Add Subject)
			{{url}}/subject/add
			
			method:POST
				Moderator should be already login.
				
				M1 login information.
					Auth token from login information in the header.
					
					Body:
					
					{
					"name": "{{$randomFullName}}"
					,
					"subject_code": "{{$randomFull Name}}",
					"description": "This is description"
					}
						Api should return a created subject information.
						
						The id of the subject should be generated.
							Pass

T10	Moderator should be able to update subjects.	Subject
		
		(Update Subject)
			{{url}}/subject/patch
			
			method:PATCH
				Moderator should be already login.
				
				The subject that is going to patch should already exists in the system.
					Auth token from login information in the header.
					
					Body:
					
					{
					"_id": "{{subject_id}}",
					"name": "{{subject_name}}",
					"subject_code": "{{SubjectCod e}}",
					"description": "This is description",
					"sections" : [
					"5f4294e677fbf33264060fab",
					"5f42985cd8717a41988a207b"
					,
					"5f42985dd8717a41988a207c"
					,
					"5f42985ed8717a41988a207d"
					
					]
					
					}
					
					subject id and subject codes are the one automatically generated in the add subject api.
					
					subject_id:5f71734a1f8790001 3a191f5
						Api should return patch subject information along with id.	Pass

					subject_code:DARLENE JONES		

T11	Moderator should be able to delete subjects.	Subject
		
		(Delete Subject)
			{{url}}/subject/delete
			
			method:DELETE
				Moderator should be already login.
				
				The subject that is going to patch should already exists in the system.
					Auth token from login information in the header.
					
					{
					"_id" : "{{subject_id}}"
					}
					
					subject_id: 5f71734a1f87900013a191f5
						Api should return whether delete is successful or not.
						
						In this case delete should be successful because subject exists.
							Pass

T12	Moderator should be able to get information about the created subject	Subject (Get Subject)	{{url}}/subject/{{SubjectCode}}	Moderator should be already login. The subject that is going to GET should already exists in the system	Auth token from login information in the header. SubjectCode:RAYMOND BERGSTROM	Api should return success response with detail of the subject information.	Pass

T13	Moderator should be able to add section related to subject.	Section
		
		(Add Section)
			{{url}}/section/add
			
			method:POST
				Moderator should be already login.
				
				The subject that section is going to be added should already be existed in the database.
					Auth token from login information in the header.
					
					Body:
					
					{
					"subject_code": "{{SubjectCode}}",
					"name": "Comments",
					"type": "Comments",
					"description": "This is description1213",
					"owner": "5f43a8cd8fa9d53f50226126"
					}
						Api should return whether section adding is successful or not.	Fail.
							
							Api is stuck and does not return any response

T14	Moderator should be able to delete the section.	Section (Delete Section)	{{url}}/section/delete method:DELETE	Dependent on section adding api.			