COMP1640-Group-Report

Bao cao tai chinh (Trường Đại học FPT)

Group Report

COMP1640 - Enterprise Web Software Development

Subject Code: COMP1640

Subject Title: Enterprise Web Software Development

Team Name: *TeamName*

Submission Date: *dd/mm/yyyy*

Team Information

Full Name	UoG Id
Nguyễn Đức Dũng	1234567
Trân Phan Vĩnh Hoàng	1234567
Phan Đình Đạt	1234567
	1234567
Member 5	1234567
Member 6	1234567
Member 7	1234567

Contents

1. Introduction about the project

This is a project to talk about creating a management system and collecting employee ideas and initiatives and how managers can manage them. The ideas will be submitted and undergo a management review before the ideas can be recorded. Any ideas submitted successfully will be commented below and evaluated to see if the idea is feasible. The admin can manage everyone's accounts.

For system development, we choose Reactjs to develop because of the strengths they bring.

2. Roles

In this project, we have followed the Scrum method strictly and have divided the roles of each member in detail and according to the capacity of each member as follows:

- Nguyen Duc Dung

- Tran Phan Vinh Hoang

- Huynh Ngoc Hai Dang

- Phan Dinh Dat

- Nguyen Ngoc Linh

- Tran Van Quy

3. Tools and Technologies

In this project, we have used the following technologies and technologies to develop the project in the best way.

3.1. Technologies.

3.1.1. Reactjs.

React is a JavaScript library created in collaboration between Facebook and Instagram. ReactJS is a popular JavaScript library for developing user interface (UI) frameworks. By utilizing a different technique for page delivery, the device provides clients with faster reaction times while entering information. The advantage of using ReactJS for website architecture is that it tends to be supplied on the server rather than only running on the client. Even the ability to associate with one another exists. DOM refreshes have little effect on the instrument. It's also possible to compare the progressions between the worth of this render and the previous one.
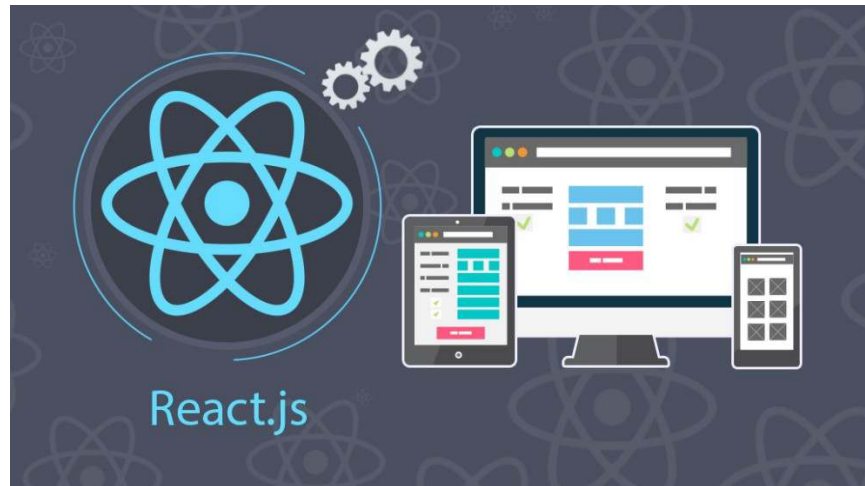
3.1.2. Nodejs:

Nodejs, also known as an asynchronous event-driven JavaScript runtime, was created with the goal of allowing developers to create scalable, quick, and low-cost applications of any size.

3.2. Tools:

3.2.1. Visual Studio Code.

Microsoft's Visual Studio Code is a completely free code editor available for Windows, Linux, and macOS. This is a one-of-a-kind and excellent combination of an IDE and a code editor. Visual Studio Code supports the web and a wide range of web applications. It also has its own editor and website designer, making it ideal for developing website-related applications. C/C++, C#, Visual Basic, HTML, F#, CSS, JavaScript, and other languages are supported.

3.2.2. GitHub:

GitHub is a project management system and code repository, as well as a social network for programmers only. Source code can be cloned from a repository by programmers. This is also a code repository server service, and anyone can sign up for an account to set up their own repository to work with.

GitHub allows us to see what other members have committed and what they have not committed. And contribute to the development of more branches in the team's source code. The principle is that the team leader can split the project into many branches to develop it, but you must eventually merge into the Master branch to get a complete project.

Figure 3 - Git Branch. Source: https://vn.got-it.ai/blog/huong-dan-cach-tao-branch-trong-git#:~:text=GitHub%20l %C3%A0%20m%E1%BB%99t%20n%E1%BB%81n%20t%E1%BA%A3ng,branch%20m%E1%BA%B7c %20%C4%91%E1%BB%8Bnh%20ch%C3%ADnh%20th%E1%BB%A9c.

3.2.3. Trello:

For group projects, where many members work together, especially in the ongoing Covid-19 situation, we need to have a software to assign work and monitor the completion of each member's work. That's why we used Trello – Effective Remote Work Support is an extremely useful and easy-to-use application that helps users organize and manage their Online work efficiently and quickly.

Figure 4 - Trello. Source: https://trello.com/

3.2.4. Dbdiagram.io:

In order to be able to design databases before starting to build the system, we designed the database first through the dbdiagram.io website. From there, the relationships between fields and tables can be visualized in advance, making the system more tight and secure.

Introduce Technologies and Development Stack of the project.

4. Group Information summary

Team Name	*TeamName*
Group Repository	*Tonydang223/Evaludating-App (github.com)*
Group Board URL	*https://trello.com/b/hrReYDQR/idea-management-system*
Deployment URL	
Front End Tech	*React*
Back End Tech	*Node, etc.*
Database Tech	*MySQL, MongoDB, SQL Server etc.*
Screen Cast URL	*www.myscrencast.com*

4.1 Team Members information

Full Name	Role
Hai Dang	Developer, Product Owner
Nguyen Duc Dung	Scrum Master, Developer
Tran Phan Vinh Hoang	Developer
Member 4	Developer
Member 5	Developer
Member 6	Developer
Member 7	Tester

5. Agile Documentation

5.1 Meetings

- Include Meeting minutes of All Sprint Planning
- Include Meeting minutes of All Spring Review
- Include Meeting Minutes of All Sprint Retrospectives

5.1.1. Sprint Planning.

Sprint Planning I.

Meeting Minutes

Participants (8/8): Nguyen Duc Dung, Nguyen Ngoc Linh, Huynh Ngoc Hai Dang, Phan Dinh Dat, Tran Phan Vinh Hoang, Dinh Nhan, Tran Van Quy, Le Dinh Huy.

Meeting type: Sprint Planning.

Minutes taker: Nguyen Ngoc Linh

Objective: The team members will review its backlog and decides what items to prioritize for the sprint project.

Topics	Participants	Time
Learn about the specification and requirements of the exercise	Scrum team	5 minutes
Decide on the duties of each member	Scrum team	15 minutes
Prioritizing Product Backlog	Linh, Dung	15 minutes
Picking out Product Backlog items for Sprint 1 Backlog	Development team	15 minutes
Calculate team capacity	Scrum team	15 minutes
Discuss any new information that may impact the plan	Scrum team	15 minutes
Q&A between the members and ending the meeting	Scrum team	10 minutes

Notes:

- This is the first discussion meeting before starting to implement the project and apply the Agile Scrum Method to a large project, we will adhere to the Scrum methods and principles.

- The Product Owner and Scrum Master finished processing the Product backlog early. In order for the project to be completed on time because the implementation period is only two months, we decided to proceed as soon as possible.

- We will carefully carry out the preparation steps and research the most careful method to adhere to the Scrum principles.

- All members will estimate the project implementation time and outline the problems in advance to handle when any problems occur and have a timely solution.

Tools: Visual Studio Code, GitHub, Whimsical, Dbdiagram, Trello.

Languages: JavaScript (Reactjs), EJS, CSS, Shell.

Figure 5 - Sprint Planning 1.

Sprint Planning II.

Location: High Five Coffee – Châu Thị Vĩnh Tế

Date: March 1th 2022

Time: 8:00AM – 9:30 AM

Meeting Minutes

Participants (8/8): Nguyen Duc Dung, Nguyen Ngoc Linh, Huynh Ngoc Hai Dang, Phan Dinh Dat, Tran Phan Vinh Hoang, Dinh Nhan, Tran Van Quy, Le Dinh Huy.

Meeting type: Sprint Planning.

Minutes taker: Nguyen Ngoc Linh

Objective: The team members will review its backlog and decides what items to prioritize for the sprint project.

Topics	Participants	Time
Prioritize unfinished work.	Dung , Linh	10 minutes
Select product backlog items for the Sprint 2 Backlog.	Development team	20 minutes
Analyze capacity to hand over tasks to team members.	Scrum team	15 minutes
List out any new information that may affect system development.	Scrum team	15 minutes
Review the issues and concerns raised during the meeting.	Scrum team	10 minutes
Q&A between members and closing the meeting.	Scrum team	10 minutes
Prioritize unfinished work.	Scrum team	10 minutes

Notes:

- This is the project's second execution plan. We began the meeting with Dung reminding us of the project's main goal, after which he and Linh would replace and prioritize the backlog products.

- The backlog has been refined, and members of the improvement team are participating in the widgets they will create for this dash and estimating the time it will take to complete the task.

- We check every member's supplies and calculate crew mobility after anyone chooses their travel gear and agrees to anyone's mission. Quy suggests that he should have his own set of kinship topics for the duration of this article in order to shorten his running time, but he implies that this will not happen. Furthermore, the crew advised that there be no obstacles and that the journey be risk-free.

- All implementation plans in the council are reached through consensus expressed in the group leader's style.

- Finally, hold a Q&A session with event attendees who want to help improve functionality and answer any remaining questions. The process then comes to an end, and the race begins.
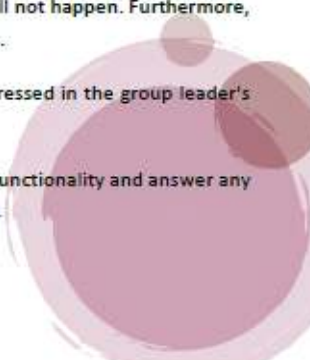
Figure 6 - Sprint Meeting 2.

Sprint Planning III.

Sprint Planning IV.

5.1.2. Sprint Review.

5.1.3. Sprint Retrospectives.

5.2 Product Backlogs

- Include Product Backlog of the Project

5.3 Sprint Backlog

- Include Sprint Backlog of each Sprint

5.4 Burndown Chart

- Include Burndown chart of the Project with brief comment for each burndown chart

6. Design Document

6.1 Use Case Diagram.

6.1.1 Admin diagram.
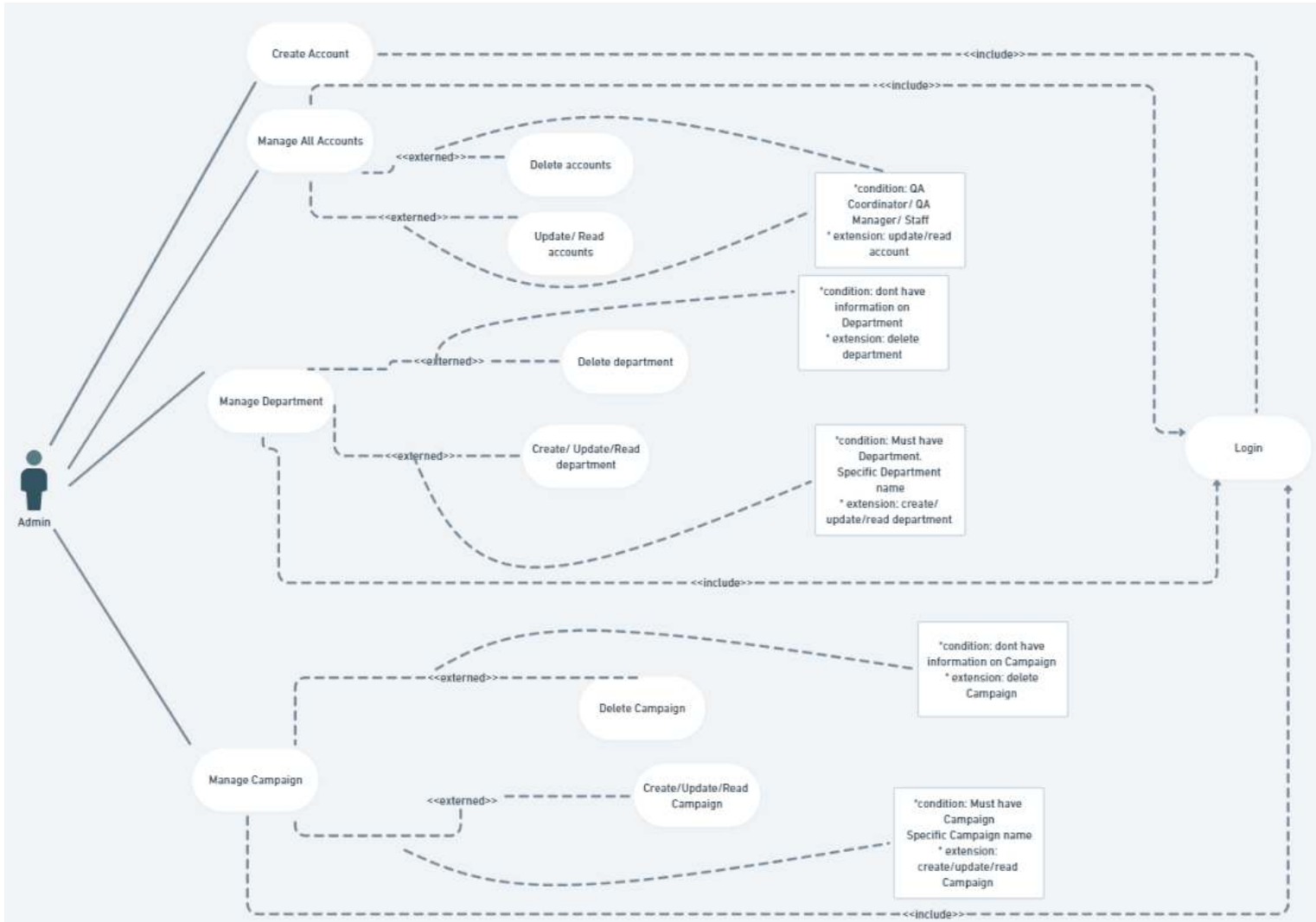
Figure 7 - Admin diagram.

6.1.2 QA Coordinator diagram.
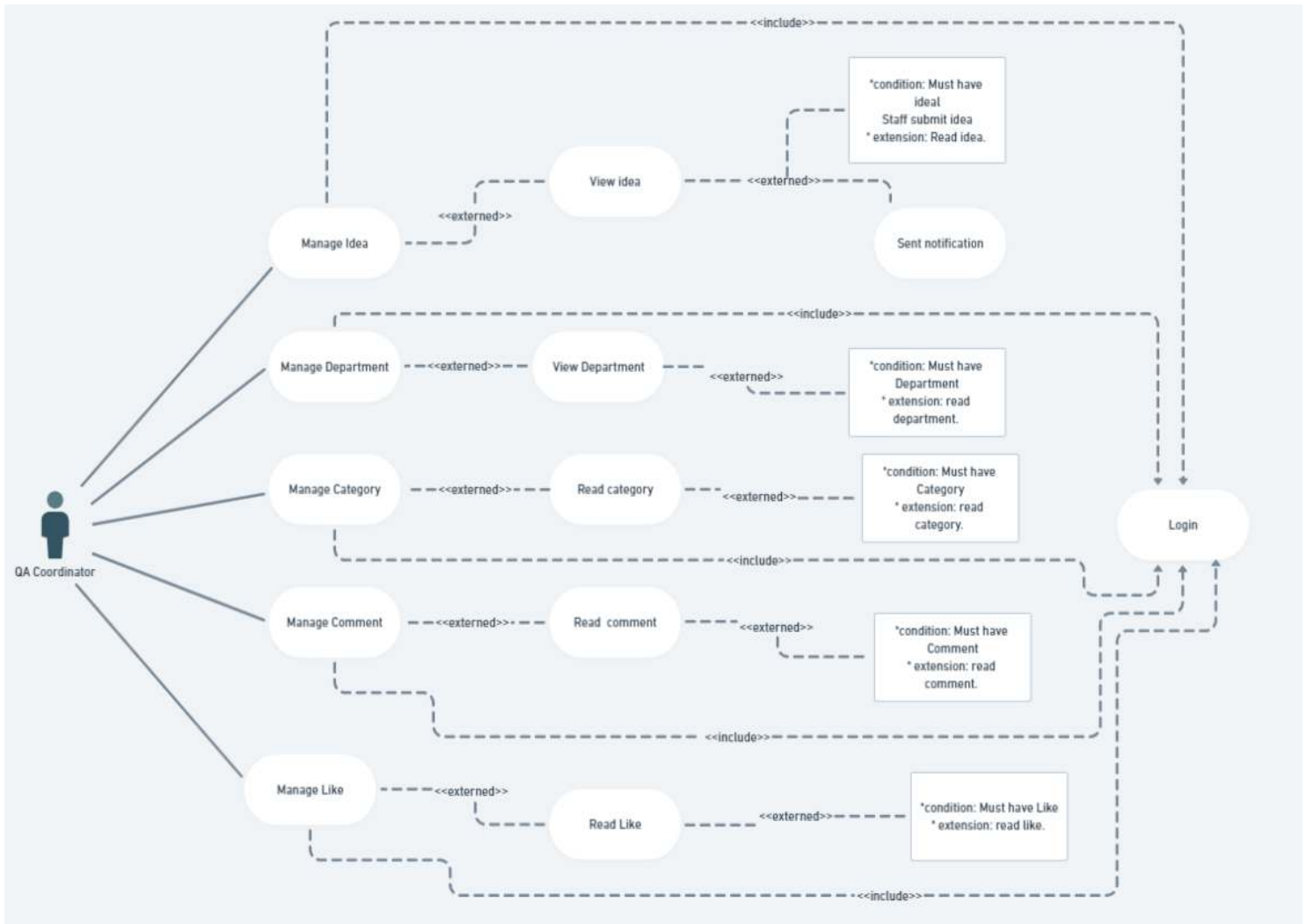
Figure 8 - QAC diagram.

6.1.3 QA Manager diagram.
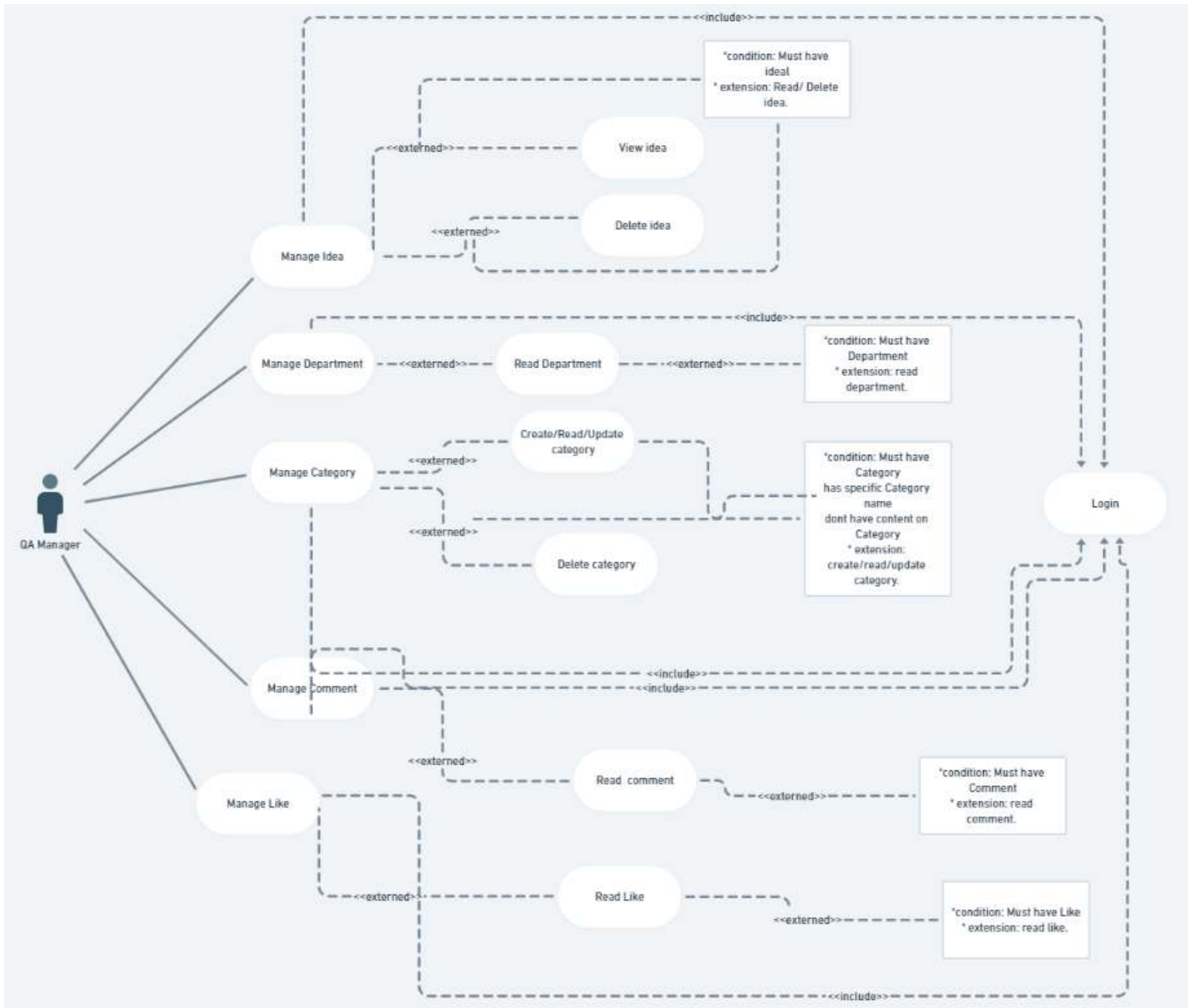
Figure 9 - QAM diagram.

6.1.4 Staff diagram.

6.2 ER Diagram.
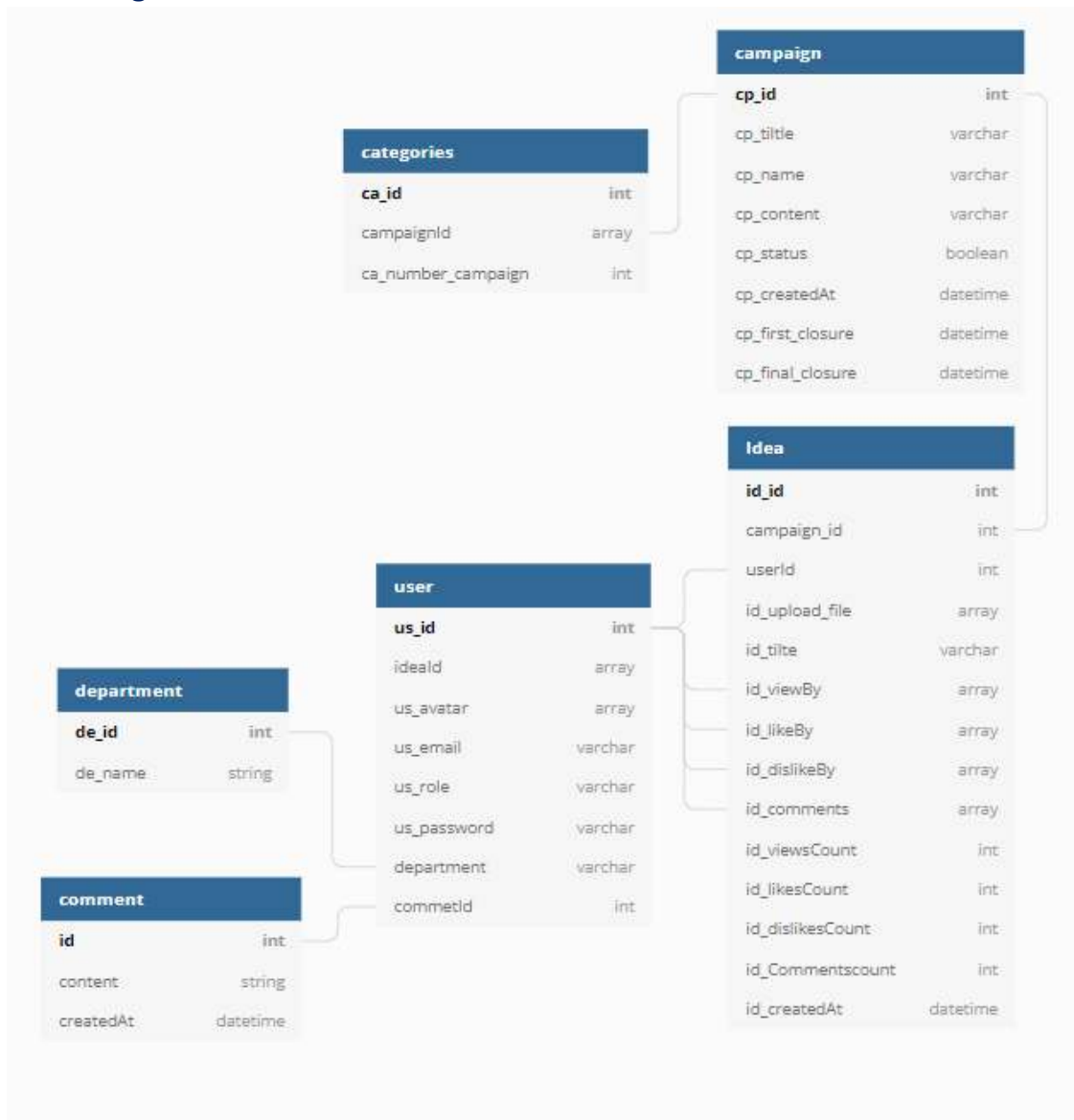
Figure 10 - ERD Diagram.

Data Validation:

6.3 Sitemap

6.4 Wireframes.

With the requirements of the topic, we will have to create a system to collect ideas from staff with roles such as Admin, QA Coordinator, QA Manager and Staff.

6.4.1 Admin screen.

Admin login to the system with the default account that has been set up.

The system will take the Admin to a separate admin page, where the administrator maintains system data

Eg:

-Admin manages all accounts (add, edit, delete, details…)

-Admin has arranged opening and closing dates for each school year, has a detailed information sheet of Staff

-Admin sees the statistical data table and can set the number of ideas for each department

Figure 11 - Admin login.

Figure 12 - Admin: all account management.

1.1.1 QA Coordinator screen.

QA receives an account directly from the Admin, they log in to the system, then the system will take the QA to their management page.

-Here each department will have a QA responsible for managing the process for their department and encouraging contributions to help their employees come up with many good ideas.

-Each employee's idea is sent, will go to QA's mail to read and review(delete) before being posted to the system (here to help avoid spam and inappropriate posts).

1.1.2 QA Manager screen.

The system has a Quality Assurance Manager account created by the Administrator for the Principal.

QAM logging into the system will take you to a separate page to monitor the quality assurance process

- QAM adds additional categories at any time and may remove categories, but only if they are not in use.

-QAM at the end of the day downloads the data already packaged in a CSV file as a ZIP file.

1.1.3 Staff screen.

Each employee is assigned 1 by Admin at a separate login account. After logging in Employees are taken to a page dedicated to them.

- Hire Staff (Students and Support) can all voice their ideas, they can have many ideas. When they come up with an idea, they can upload additional supporting documents.

- But when submitting an idea, they must agree to the terms and conditions given.

- Each Staff can see all the ideas posted on the system, here they can comment to help contribute to the idea (like or dislike) but only once for any idea to avoid spamming the post many times to annoy others.

Figure 13 - Staff login.

Figure 14 - Staff post Idea screen.

Figure 15 - Staff view idea post and comment.

1.2 Activity Diagram / Sequence Diagram.

2. Implementation

2.1 Screenshot of the codebase.

2.1.1 Admin codebase.

2.1.2 QA Coordinator codebase.

2.1.3 QA Manager codebase.

2.1.4 Staff codebase.

2.2 Screenshot of the final Product.

2.2.1 Admin screenshot.

2.2.2 QA Coordinator screenshot.

2.2.3 QA Manager screenshot.

2.2.4 Staff screenshot.

2.3 Test Cases

2.3.1 Admin test cases.

a. Test plan.

No.	Function	How to test	Expected test results	Actual test results
1	**Login to system**	Login as Administrator	Logged in successfully	Logged in successfully
2	**Login to system**	Log in with an account that doesn't exist	Login failed	Login failed

3	**Logout to system**	Logout the system while using user account	Signed out successfully	Signed out successfully
4	**Create account**	Create accounts for users with email, password, name and department, and role.	Create Account Success	Create Account Success
6	**Create account**	Create an account with missing information (name/password/email or role).	Can't create an account	Can't create an account
7	**Delete account**	Delete selected account	Deleted Account Success	Deleted Account Success
8	**Edit account**	Edit information for user account	Edited Account Success	Edited Account Success
9	**Edit account**	Edit information for user account with blank information	Can't edit information	Can't edit information
10	**Edit account**	Edit information for user account with same information with others	Can't edit information	Can't edit information
11	**Edit function**	Edit information for user account without inputting information	Can't edit information	Can't edit information
17	**Edit account**	Reset password for user account	Reset password successfully	Reset password successfully
18	**Create Department**			
19	**Create Department**			

Table 1 - Admin test logs.

b. **Test logs.**

No.	Function	How to test	Expected test results	Evidence results	Time to Test	Tester
1	**Login to system**	Login as Administrator	Logged in successfully			Quy
2	**Login to system**	Log in with an account that doesn't exist	Login failed			Quy

3	**Logout to system**	Logout the system while using user account	Signed out successfully			Quy
4	**Create account**	Create accounts for users with email, password, name and department, and role.	Create Account Success			Quy
6	**Create account**	Create an account with missing information (name/password/email or role).	Can't create an account			Quy
7	**Delete account**	Delete selected account	Deleted Account Success			Quy
8	**Edit account**	Edit information for user account	Edited Account Success			Quy
9	**Edit account**	Edit information for user account with blank information	Can't edit information			Quy
10	**Edit account**	Edit information for user account with same information with others	Can't edit information			Quy
11	**Edit function**	Edit information for user account without inputting information	Can't edit information			Quy
17	**Edit account**	Reset password for user account	Reset password successfully			Quy
18	**Create Department**					
19	**Create Department**					

2.3.2 QA Coordinator test cases.

a. **Test plan.**

No.	Function	How to test	Expected test results	Actual test results
1	**Login to system**	Login as QAC	Logged in successfully	Logged in successfully
2	**Login to system**	Log in with an account that doesn't exist	Can't Login	Can't Login
3	**View staff contribute**	View Department-based Staff contributions.	Show all information about contributions	Show all information about contributions

4	Receive notification	Get a Staff account and submit ideas.	Send notice to QAC account	Send notice to QAC account
5	Receive notification	Get a QAM account and delete ideas.	Nothing notices sent to QAC account	Nothing notices sent to QAC account

b. Test logs.

No.	Function	How to test	Expected test results	Evidence results	Time to Test	Tester
1	Login to system	Login as QAC	Logged in successfully			Quy
2	Login to system	Log in with an account that doesn't exist	Login failed			Quy
3	View staff contribute	View Department-based Staff contributions.	Show all information about contributions			Quy
4	Receive notification	Get a Staff account and submit ideas.	Send notice to QAC account			Quy
5	Receive notification	Get a QAM account and delete ideas.	Nothing notices sent to QAC account			Quy

2.3.3 QA Manager test cases.

a. Test plan.

No.	Function	How to test	Expected test results	Actual test results
1	Login to system	Login as QA Manager	Logged in successfully	Logged in successfully
2	Login to system	Log in with an account that doesn't exist	Login failed	Login failed
3	Oversee the process	Comprehensive view of the system sequence	Show all system status	Show all system status
4	Edit Categories	Edit all information in the Category field	Edit category successfully	Edit category successfully
5	Edit Categories	Edit all information with blank information	Can't edit category	Can't edit category

6	Add Categories	Enter all information in the Category field	Add category successfully	Add category successfully
7	Add Categories	Missing Category information	Can't add category	Can't add category
8	Delete Categories	Delete Category with no ideas in it.	Delete category successfully	Delete category successfully
9	Delete Categories	Delete Category with exist ideas in it.	Can't delete category	Can't delete category
10	Download exist data	Click the Export button, all available data is downloaded as a Zip file	Export data successfully	Export data successfully
18	Analyze data	View analyze data-based system database	View chart successfully	View chart successfully

b. **Test logs.**

No.	Function	How to test	Expected test results	Evid
1	Login to system	Login as QA Manager	Logged in successfully	
2	Login to system	Log in with an account that doesn't exist	Login failed	
3	Oversee the process	Comprehensive view of the system sequence	Show all system status	
4	Edit Categories	Edit all information in the Category field	Edit category successfull	
5	Edit Categories	Edit all information with blank information	Can't edit category	
6	Add Categories	Enter all information in the Category field	Add category successfull	
7	Add Categories	Missing Category information	Can't add category	
8	Delete Categories	Delete Category with no ideas in it.	Delete category successfully	
9	Delete Categories	Delete Category with exist ideas in it.	Can't delete category	

No.	Function	How to test	Expected test results	Actual test results	
10	**Download exist data**	Click the Export button, all available data is downloaded as a Zip file	Export data successfull		
11	**Analyze data**	View analyze data-based system database	View chart successfully		V su

2.3.4 Staff test cases.

a. **Test plan.**

No.	Function	How to test	Expected test results	Actual test results
1	**Login to system**	Login as Staff account	Logged in successfully	Logged in successfully
2	**Login to system**	Log in with an account that doesn't exist	Login failed	Login failed
3	**Logout to system**	Logout the system while using staff account	Signed out successfully	Signed out successfully
4	**Submit idea**	Submit ideas when filling out Title, content and attachments.	Submit successfully	Submit successfully
5	**Submit idea**	Submit an idea while leaving the Title field blank.	Can't submit	Can't submit
6	**Submit idea**	Submit an idea while leaving the Content field blank.	Can't submit	Can't submit
7	**Submit idea**	Submit an idea while leaving the attachment blank.	Submit successfully	Submit successfully
8	**Submit idea**	Submit an idea but click Agree to terms.	Submit successfully	Submit successfully
9	**Submit idea**	Submit ideas but do not click Agree to terms.	Can't submit	Can't submit
10	**Submit idea**	Submit more ideas	Submit successfully	Submit successfully
11	**Comment to idea**	Comment on ideas using Staff account	Comment successfully	Comment successfully

12	**Like/ Dislike to idea**	Like/ Dislike on ideas using Staff account	Like/ Dislike successfully	Like/ Dislike successfully

b. Test logs.

No.	Function	How to test	Expected test results	Evidence results	Time to Test	Tester
1	Login to system	Login as Staff account	Logged in successfully			Quy
2	Login to system	Log in with an account that doesn't exist	Login failed			Quy
3	Logout to system	Logout the system while using staff account	Signed out successfully			Quy
4	Submit idea	Submit ideas when filling out Title, content and attachments.	Submit successfully			Quy
5	Submit idea	Submit an idea while leaving the Title field blank.	Can't submit			Quy
6	Submit idea	Submit an idea while leaving the Content field blank.	Can't submit			Quy
7	Submit idea	Submit an idea while leaving the attactment blank.	Submit successfully			Quy
8	Submit idea	Submit an idea but click Agree to terms.	Submit successfully			Quy
9	Submit idea	Submit ideas but do not click Agree to terms.	Can't submit			Quy
10	Submit idea	Submit more ideas	Submit successfully			Quy
11	Comment to idea	Comment on ideas using Staff . account	Comment successfully			Quy
12	Like/ Dislike to idea	Like/ Dislike on ideas using Staff . account	Like/ Dislike successfully			Quy

2.4 Screencast link.

3. Evaluation

3.1 Assumptions

We made some assumptions while planning the development of the system, here are some:

- QA Manager can download compressed Staff posts from zip files.

- Admin can create Accounts for all and manage them (delete, edit).

- Different Staff accounts can like, comment on other posts and only delete their own posts.

- QA Manager can see all posts of all and has authority over Staff's posts.

- QA Coordinator can only view all posts by all.

3.2 System Strengths

The main strengths of our IDEAL project are:

- Beautiful, intuitive UI design, easy for users to access and use.

- Simple interface layout, easy for users to operate.

- New technology so low latency.

- Collected all Employee comments and categorized them by Category.

- High security because using BaaS (MongoDB), easy and fast data query.

- Clear and secure authorization should only display and access the parts according to different roles.

3.3 System weaknesses

Besides the advantages listed above, the system also has the following disadvantages:

- Although the interface is beautiful and intuitive, the UIs are designed by themselves without applying technologies like Bootstrap, so the interface is not eye-catching.

- Using MongoDB does not have the same binding properties as in RDBMS, so when working with MongoDB, you must be very careful. In addition, using MongoDB will consume more system resources than RDBMS.

3.4 Challenges

About technology:

- We use Reactjs, Nodejs but team members do not have much experience in this field, so it takes a while to learn more.

About team management:

- Initially, we did not divide the project into small stages to easily divide the specific schedule, causing delays.

- Do not regularly interact directly with project team members because of the Covid-19 epidemic situation, so they have to communicate online through different means.

- Because the Covid-19 epidemic lasted for a long time, some team members were infected, so there was not much time for group meetings, which led to a number of consequences that members could not keep up with the development of the project they had to give up. more time to keep up with the progress.

- This is also the last stage of the semester, so some members have to focus a lot on the final project, so they have little time for the project.

3.5 Lessons learned.

Lessons learned after completing the project are:

- Understand scrum working model and solutions, deploy work from it.

- How to work and hand over work when not present in person. For example, during Covid-19, we had to use a handover and work monitoring tool like Trello to be able to message and hand over with team members.

- Use different means of communication.

- Solve problems within the group. For example, in the group, there is a person who is quite weak in all areas, unable to complete the tasks assigned before. We reorganized a meeting and decided to give her 3 days to re-analyze her strengths and weaknesses. After that time, if you are able to continue participating in the group project, you will do otherwise, you will have to withdraw from the group to avoid affecting the collective results.

- Divide the tasks in the post according to each person's ability. For example, in our team, there is a person who has a very good team management ability and understands all the processes of the system, so the position of Scrum Master / Product Leader is very suitable for you. As for someone who is knowledgeable and experienced in UI / UX design, the Frontend Developer position is very suitable.