Sprints

1. Sprint 1
    a. Sprint 1 Planning
    b. Stand-up Meeting Note
        i. Sprint 1-1 Stand-up
        ii. Sprint 1-2 Stand-up
        iii. Sprint 1-3 Stand-up
        iv. Sprint 1-4 Stand-up
    c. Sprint 1 Review
    d. Sprint 1 Retrospective
2. Sprint 2
    a. Sprint 2 Planning
    b. Stand-up Meeting Note
        i. Sprint 2-1 Stand-up
        ii. Sprint 2-2 Stand-up
        iii. Sprint 2-3 Stand-up
    c. Sprint 2 Review
    d. Sprint 2 Retrospective
3. Sprint 3
    a. Sprint 3 Planning
    b. Stand-up Meeting Note
        i. Sprint 3-1 Stand-up
        ii. Sprint 3-2 Stand-up
    c. Sprint 3 Review
    d. Sprint 3 Retrospective

Sprint 1

1. Sprint Planning
2. Stand-up Meeting Note
    a. Sprint 1-1 Stand-up
    b. Sprint 1-2 Stand-up
    c. Sprint 1-3 Stand-up
    d. Sprint 1-4 Stand-up
3. Sprint Review
4. Sprint Retrospective

Sprint Planning

From the user stories, the development team has separated the whole project into many small tasks. In sprint 1, the team focus on the basic functions that can build up the website. The tasks are clustered by their main ideas, in sprint 1, the team mainly focus on “user management”.

In sprint 1, the team aim to achieve the following circumstance,

As a user, he/she can register an account by typing the basic information and choosing a password. After the user clicks the link in the verifying email, he/she could log in the account. Moreover, the user can choose or create tags that can best describe himself/herself. For the security reason, user not only can reset the password but also can get password recovery email when forgetting the password.

The detail of sprint 1 plan is shown in the following table,

Function	As a user, I could create a new account, so that I can access functionality that requires an account	register with username, name, email, password. The account needs to be activated by email verification
	As a user, I could log in to the account, so that I can update personal information	login by username and password
	As a user, I could recover my password, so that I can get my password back when I forget.	get password recovery link via email

	As a user, I could reset my password, so that I can change my password	enter the old password and choose a new password
	As a user, I could verify my email, so that I can unlock more functionalities.	with verification link
	As a user, I could set my tags, so that I can get more relevant information and user recommendation from website	Create or select tags. Add or delete tags. view current tags.
Non-Function	Logo Design	Based on the main idea of the website
	Name	
	Server Build	Based on scalability
	Domain name	Easy to remember, linked with the website design idea
	Set up tech tool	Development tools and communication tools
	UI Design	Home Page; Search Page; Login; Register; Dashboard

After sprint 1, the team wish users can finish their user account management part.

Stand Up Meeting

1. Sprint 1-1 Stand up meeting note
2. Sprint 1-2 Stand up meeting note
3. Sprint 1-3 Stand up meeting note
4. Sprint 1-4 Stand-up meeting note

Sprint 1-1 Stand-Up notes

Date

2020-8-20

Participants

- @ Jiachen Zhou
- @ Xiancheng Yang
- @ Yunfei Jing
- @ zijian Zeng
- @ Shuyang Fan

Goals

- Setting up server
- Exchange public keys

Discussion topics

Role	Presenter	Notes
Architectural Lead	Shuyang Fan	Explore architectural design with NGINX and TiDB
Product Owner	Xiancheng Yang	List half questions for the following client meeting Contact client team to arrange a meeting time

Scrum Master	Jiachen Zhou	arrange meetings and coordinate with other members List half questions for the following client meeting, Setup server.
Testing Lead	Yunfei Jing	Starting to study Spring Boot and ReactJS, Set up server. **Azure**
Quality Lead	Zijian Zeng	coding style - Int(check style) https://plugins.jetbrains.com/plugin/1065-checkstyle-idea

Action items

- [ ]

Decisions

Sprint 1-2 Stand Up Note

Date

2020-8-27

Participants

- @ Jiachen Zhou
- @ Shuyang Fan
- @ Xiancheng Yang
- @ Yunfei Jing
- @ zijian Zeng

Discussion topics

Role	Presenter	Notes
Architectural Lead	Shuyang Fan	Set up reverse proxy for development server Set up database and other middleware(NGINX, Redis)
Scrum Master	Jiachen Zhou	Learn Spring Framework Set up and update database Working on verification token in sign up part
Testing Lead	Yunfei Jing	Set up Travis CI/CD Login and sign up features. use local storage to maintain sessions.
Quality Lead	Zijian Zeng	App skeleton and home page. protect routes with react-dom-router.
Product Owner	Xiancheng Yang	Organise client meetings, write meeting notes and summarize client requirements. UI design for register, login, homepage Motivation Table and Graph

Sprint 1-3 Stand-Up Meeting

Date

2020-9-3

Participants

- @ Xiancheng Yang

Host

- @ Shuyang Fan

Goals

Weekly Stand-Up Meeting

- Talk about what each member did during last week

Discussion topics

Role	Presenter	Notes
Architectural Lead	Shuyang Fan	• Configure Travis to run the test, build a docker image then push it to Github Package (https://github.com/Haswf/COMP30022BackEndDev/packages/377289) • Set up watchtower to deploy the latest image on the development server • Implement interceptor to verify username in JWT matches username in URI path • Implement login endpoint (/authentication/login)
Scrum Master	Jiachen Zhou	• Finish implementing the basic functions for registration (/authentication/signup) • Implement email and username validators with standardized exception handler • Implement a rough structure of password reset endpoint (/users/{username} /password-reset)
Testing Lead	Yunfei Jing	• Finished Login and Signup
Quality Lead	Zijian Zeng	• Flipping effect using React Spring for pages transition. • Autocomplete Searching Bar for the Searching page. • Alert Bar
Product Owner	Xiancheng Yang	• Set up the reuse plan • Finalise user stories and Trello with team members • Search page UI design • Search page implement at front-end

Sprint 1-4 Stand-up Meeting Note

Date

2020-9-10

Participants

- @ Xiancheng Yang
- @ Yunfei Jing
- @ Shuyang Fan
- @ Jiachen Zhou
- @ zijian Zeng

Host

- @ Xiancheng Yang

Goals

Weekly Stand-Up Meeting

- Talk about what each member did during last week

Achievement

Tag system - create, use existing, add, delete

Email verification - email verification link to jump to the welcome page

Password reset

Password forgot with email recovery - email

Sprint 1 quality assurance coding

Spring Testing

Currently working on (Partial completed)

Search - by name or tag

Portfolio Editor

Integration Testing

Discussion topics

Role	Presenter	Notes
Architectural Lead	Shuyang Fan	• Set up Email as a service • create a full-text index and implement fuzzy search by keyword • Implement endpoints for modifying user tags
Scrum Master	Jiachen Zhou	• Email verification by email at the time of registering and can change the role of the user. • Password recovery by email if forgot password
Testing Lead	Yunfei Jing	• Reset Password after logging in • Modify and Update user tags with Autocomplete search box • Integrate Rich-text Editor to user dashboard • Integration Testing (on progress)
Quality Lead	Zijian Zeng	• Logo setup • Email verification page • Password recovery page • searching page
Product Owner	Xiancheng Yang	• Writing sprint 1 documentation • Sprint 1 quality assurance coding • Logo

Sprint Review

Sprint 1 Scheduled

Function	Register	register with username, name, email, password. The account needs to be activated by email verification	Completed
	Login	login by username and password	Completed
	password recovery	get password recovery link via email	Completed
	password reset	enter the old password and choose a new password	Completed
	email verification	with verification link	Completed
	tag system	Create or select tags. Add or delete tags. view current tags.	Completed
	Change profile	Change information about users	Not Completed*
Non-Function	Logo Design	Based on the main idea of the website	Completed
	Name		Completed
	Server Build	Based on scalability	Completed
	Domain name	Easy to remember, linked with the website design idea	Completed
	Set up tech tool	Development tools and communication tools	Completed
	UI Design	Home Page; Search Page; Login; Register; Dashboard	Partial Completed*

*Need to solve client conflict about the target audience of this project, hence the information contents about users will be confirmed.

*Portfolio editor page UI design need to confirm with clients in the following meeting.

Not Scheduled

Search function	Search portfolio via name or tag	Completed
Portfolio Editor	Add text, pictures, videos to create portfolios, could drag elements to change position	Partial Completed

Design Idea (Logo, Name)

The name of the website is Forty-Two. Because the ASCII code 42 is for the asterisk symbol, being a wildcard for everything. The team wish user can put their entire experience into their portfolio, get more opportunities for jobs.

The logo has nine squares, the eight small squares represent personal information, personal description, tags, education, working experience, resume, introduction video and projects experiences. The big solid square represents the user's future, which is filled with promise and possibilities.

Sprint Highlight

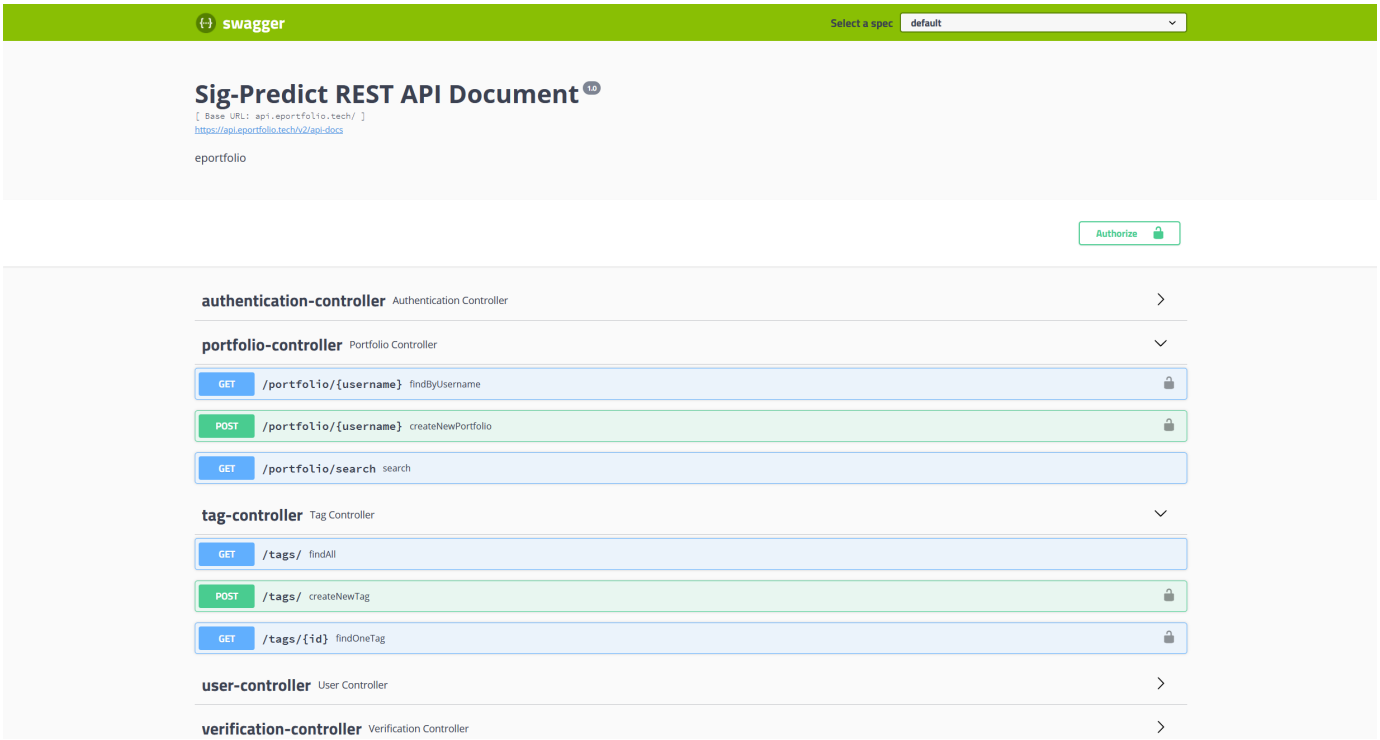
Quick Review

Based on the sprint 1 plan, we fully finished the user management part, which users can manage their accounts. All the functions showing in the plan are finished expect "user can change their profile after login". The reason is two client groups still have conflict about the target audience. We decide to do this function later after solving the conflict with second client meetings. In addition, some functions not scheduled in sprint 1 has been developed, which are search portfolios and portfolio editor.

Positive Highlights

In sprint 1, there are several things that we did well. At first, our team is well organised. Every group members have been assigned a role based on their strengths, everyone did their own works well and also work with other members if need. In addition, our group did teamwork well. We have three scheduled meeting per week, team members talk about what they had finished and the challenges they were facing. It is also a good opportunity to connect front-end and back-end works because we have to two git repositories, one for the front-end and one for the back-end. These two parts are developed independently.

Furthermore, our code quality is well-performed. For example, our API page is organised by swagger, the front-end team can view all API directly. Not only the swagger, but we also use Travis CI/CD to build pipeline automatically. These automatic tools improve our development efficiency.



Moreover, every team member finished their work on time, which allowed us to finished the sprint 1 plan ahead of the schedule and started sprint 2 works.

Reflect on Roadblocks and Improvement

There are also some challenges we can improve in the coming sprints. At first, we use some new technologies in this project, such as Azure server, TiDB and Spring framework. We are not familiar with these technologies at the beginning of development. For example. the docker usually crushes without modification. After a few weeks, we have improved our skills in using them and found out the problems that caused crushed. In addition, our documentation structure is disorganized at the beginning, in the following sprints, we will make our documentation up to date.

Moreover, the clients' conflict also affected our development. Because we got double clients, there are more functions to do and some conflict. For example, portfolio content requirements are different. Hence, some of the tasks had to wait until discussing with clients. In the following sprints, we will solve this problem by increasing the communication times with clients.

Other Important Feedback

According to our supervisor, we also need to focus on testing. In this sprint, we have done some integration testing for front end and also some spring testing. However, we should find some automatic testing tools later. We also should record the important group meetings, not only stand-up meeting and client meeting. The remaining functions need the front-end team and back-end team to work closer.

Take Action

In the sprint 2, here are some things to improve,

- Sprint 2 plan
- Keep communication with clients
- More communication between front-end and back-end
- Recording meeting

Retrospective Sprint 1

Shuyang Fan

Good

- Good communication via Slack and Github Pull Request

- Establish fully automatical CI/CD pipeline to simplify deployment

Not Good

- Insufficient attention to testing

Improvement

- Should start implementing test earlier

Jiachen Zhou

Good

- Nice teamwork with pull requests
- Great progress among groups

Not Good

- spend too much time on interesting stuff so may not have enough time to focus on the main progress

Improvement

- Need to have more rest and sleep

Zijian Zeng

Good

- Professional software development works
- Well cooperation

Not Good

Yunfei Jing

Good

- Continuous Integration and Deployment using Travis CI

Not Good

Improvement

- Need have a more specific workflow

Xiancheng Yang

Good

- Set up a complete plan at the beginning and followed the plan well
- Every team member take the responsibility for their won works and finished on time

Not Good

- The plan set is below our ability
- Documentation needs to update period

Improvement

- Evaluate our ability to do the sprint 2 plan
- Communicate more with clients to clarify requirements

Sprint 2

1. Sprint Planning
2. Stand-up Meeting Note
    a. Sprint 2-1 Stand-up
    b. Sprint 2-2 Stand-up
    c. Sprint 2-3 Stand-up
3. Sprint Review
4. Sprint Retrospective

Sprint 2 Planning

In sprint 2, the team focus on the portfolio functions that users can view and create portfolios. The main topic of sprint 2 that the team mainly focus on “portfolio management”.

In sprint 2, the team aim to achieve the following circumstance,

As a user, he/she can search portfolios by the search bar on the homepage. In addition, they also can choose tags on the portfolio to search portfolios within a field. Moreover, the user can create their own portfolio based on the current templates or create their own template. They can upload pictures and videos into their portfolios.

The detail of the sprint 2 plan is shown in the following table,

Function	As a user, I could change profile information, so that I can update my infomation	personal details delete the account
	As a user, I could see homepage, so that I can see the functionality and introduction of this website.	Change the animation add the search bar Portfolio view by tags
	As a user, I could use portfolio editor, so that I can create my own portfolio.	Templates Add media font, size, colour, layout
	As a user, I could search portfolio, so that I can find some specific portfolios I am interesting in.	Search by tags Search by names
	As a user, I could redirect to 404 Page, so that I can’t view invalid URL.	Redirect to the 404 page if get an invalid URL
	As a user, I could view portfolios, so that I can see other users' experience.	profile page
Non-Function	Homepage redesign	Remove animation
	Theme colour	
	Refactor navigation bar	

Stand-Up Meeting Note

1. Sprint 2-1 Stand-Up Meeting Note
2. Sprint 2-2 Stand-Up Meeting Note

3. Sprint 2-3 Stand-Up Meeting Note

Sprint 2-1 Stand-Up Meeting Note

Date

2020-9-17

Participants

- @ Shuyang Fan
- @ Xiancheng Yang
- @ zijian Zeng
- @ Yunfei Jing
- @ Jiachen Zhou

Host

@ Jiachen Zhou

Achievement

- Refine and refactor the codebase
  - Eliminate TODOs from Sprint 1
- Increase 10% of the coverage rate in the backend codebase
  - The back-end test cases have covered the most crucial functionality of the website
- Create separate environments and configuration
  - local - for local development only, connect to the local database
  - development - for the development environment(i.e. dev.eportfolio.tech), connect to development database
  - test - for testing, use an embedded in-memory database
- Meet clients
- Discussion topics

Presenter	Notes
Shuyang Fan	• Adapt Jsend JSON format for HTTP response • Set up Azure storage for portfolios • Implement a blob storage controller and service
Xiancheng Yang	• Organize client meetings and record the notes • Finalize sprint 1 documentation • UI design
Zijian Zeng	• Reformat and clean code structure after sprint 1 • fronted end planned for sprint 2 • added some test cases
Yunfei Jing	• Reformat and clean code structure after sprint 1 • add update user information • research on how to design the editor for portfolio
Jiachen Zhou	• Reformat and clean code structure after sprint 1 • Connect Mongo DB on both local and server • move portfolio table to Mongo DB

Sprint 2-2 Stand-Up Meeting Note

Participants

- @ Yunfei Jing
- @ Xiancheng Yang
- @ Shuyang Fan
- @ Jiachen Zhou
- @ zijian Zeng

Host

- @ Yunfei Jing

Achievement

- Deploy SonarQube for code inspection
- The editor is fully functional
- Create a prototype of the Home page
- Create a prototype of Explore

Individual Progress

Role	Presenter	Notes
Architectural Lead	Shuyang Fan	• Set up SonarQube on https://sonarqube.eportfolio.tech/projects • Implement endpoints for accessing the template • Adjust search endpoint to limit search scope based on user status(i.e. public, verified-user, user, private) • Add job to generate mock content • Add endpoint for renewing the expiring token
Scrum Master	Jiachen Zhou	• create the database for like and comment • Implement endpoints and tests for like • Implement endpoints and tests for comment
Testing Lead	Yunfei Jing	• change profile avatar • fully functional editor • preview of portfolio
Quality Lead	Zijian Zeng	• Fix front-end issues • Create a prototype of Explore • Refactor search page
Product Owner	Xiancheng Yang	• Homepage design and implement • Working on the profile page • Footer design and implement

Sprint 2-3 Stand-Up Meeting Note

Participants

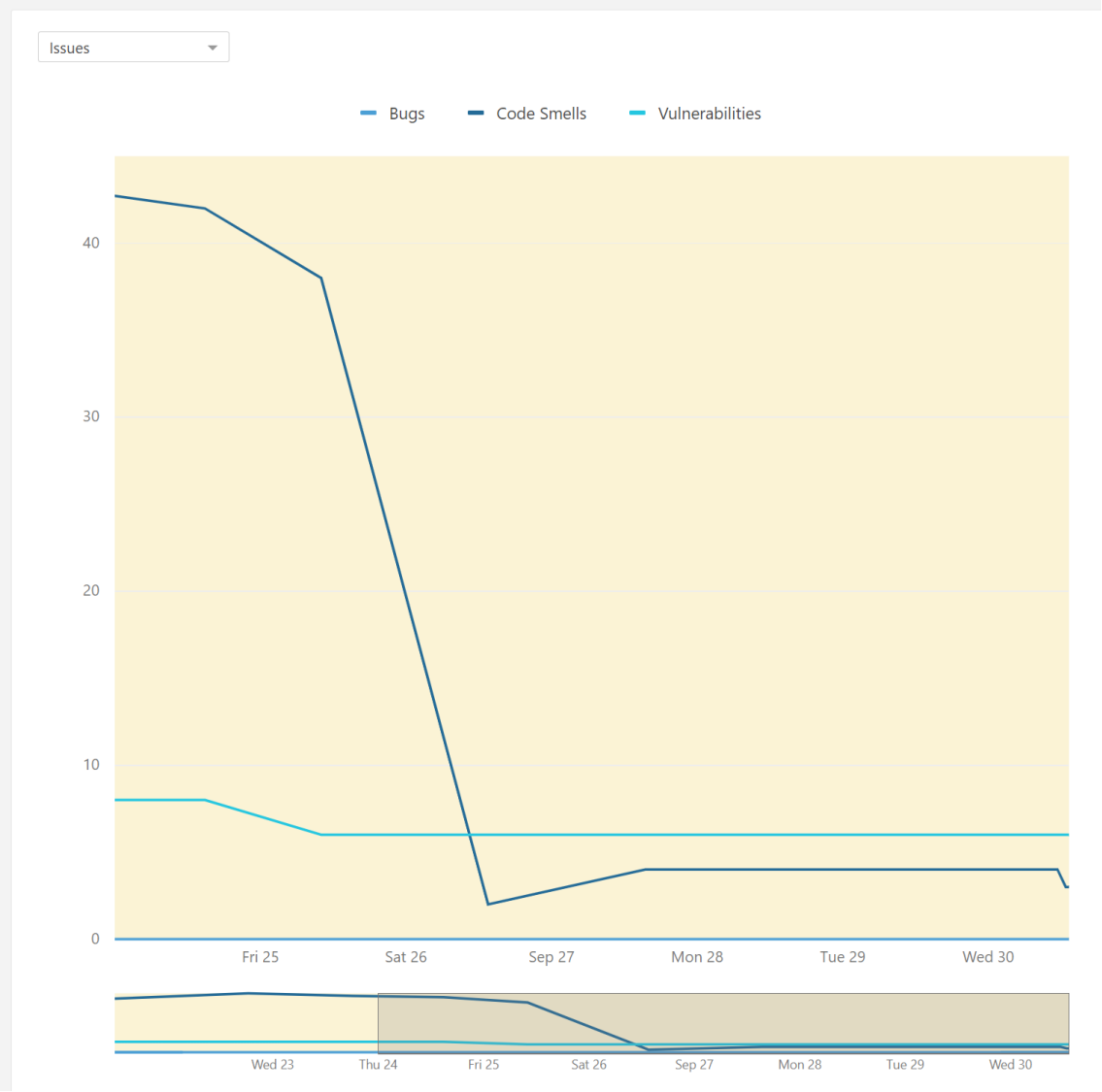
- @ Yunfei Jing
- @ Xiancheng Yang
- @ Shuyang Fan
- @ Jiachen Zhou
- @ zijian Zeng

Host

- @ zijian Zeng

Achievement

- Back-end
  - Implement endpoints for creating, deleting, updating comment
    - user can leave a comment
    - user can reply to other's comment
  - Refactoring
    - Fix security issues identified by SonarQube
    - Fix bad-smelling code
    - **Statistic from SonarQube**
    -



- Front-end
  - Redirecting 404 unfound Page (see https://dev.eportfolio.tech/notfound)
  - Explore page prototype (see https://dev.eportfolio.tech/explore)
  - Editor template/ preview functionality
  - Profile Sociality functionality
- Tool

- Introduce GitGuardian to scan credential in the codebase to prevent credential leaking
    - Credential leak scan is now part of CI/CD pipeline
    - Integrate GitGuardian with Slack to notify team

Individual Progress

Role	Presenter	Notes
Architectural Lead	Shuyang Fan	• Create a prototype for displaying comment • Change Spring security to consistently return 403 and 401.
Scrum Master	Jiachen Zhou	• Implement the endpoint for any user to follow another user with test cases • Implement a reply endpoint for users to reply to others' comments
Testing Lead	Yunfei Jing	• Template selection, upload and deletion • music player
Quality Lead	Zijian Zeng	• Explore page Demo • Editor speed dial fab
Product Owner	Xiancheng Yang	• Redirect users to 404 page • Share portfolio functionality • Like portfolio functionality

Sprint 2 Review

Quick Review

Based on the sprint 2 plan, we fully finished the portfolio management part, in which users can create their own portfolio and view others'. In addition, some functions not scheduled in sprint 2 has been developed, which are sociality portfolios, such as like, comment, share, and follow.

Highlights

In this sprint, there are several things that we did well. our team has good communication, we increased our meeting frequency to 5-6 times per week. The front end and back end communicate together to solve current problems. In addition, each member did their part well and on time. We all attended every meeting and solve problems for each other. During the development, there were many debug sessions that we working together through zoom.

We finished our sprint 2 plan earlier than scheduled and did sprint 3 works. In terms of testing, the coverage rate has increased and every new functionality has been well tested before being merged into the master branch. We create branches to develop new functions and pull requests to facilitate the code review process.

Reflect on Roadblocks and Improvement

There are also some challenges we can improve in the coming sprints. The group members had some division about UI design. In addition, the GitHub sometimes has merge conflicts. That's because we did not communicate well when developing the website. More than one people doing different work on the same file. To solve these roadblocks, we will communicate more with group members.

Sprint 2 Scheduled

Function	change profile information	personal details delete the account	Completed
	homepage	Change the animation add the search bar Portfolio view by tags	Completed

	portfolio editor	Templates Add media font, size, colour, layout	Completed
	Search portfolio	Search by tags Search by names	Completed
	404 Page	Redirect to the not-found page if get an invalid URL	Completed
Non-Function	Homepage redesign	Remove animation	Completed
	Theme colour		Completed
	Back-end Testing	All function with testing	Completed

Not Scheduled

Like function	A user can like or cancel like portfolios after login	Completed
Comment Function	A user can leave comments under others portfolios The comment also can be commented or like	Partial Completed
Share function	A portfolio can be share by copy the URL to clipboard	Completed
Follow function	A portfolio can be followed hence once the portfolio updated, the follower will get email notification	Completed

Sprint 2 Retrospective Note

Shuyang Fan

Good

- Make good use of the Github Issue to keep track of bugs and enhancements.
- Everyone is devoting their private time for collaboration and debugging. It's very easy to arrange a meeting about a specific issue with responsible developers.

Not Good

- Insufficient communication regarding major UI/UX change
- Trello board is not updated frequently enough to reflect daily progress
- Unclear responsibility makes it difficult to turn to the right one for help

Improvement

- Break user stories into sub-tasks at Sprint Planning
- Major UI change requires a pull request before being merged into master
- Prepare an agenda at each meeting. Don't hold long meetings unless it is necessary.

Jiachen Zhou

Good

- Make good use of splitting branches to develop new functionality
- Make good use of SonarQube to fix security issues
- We have good and formal commit messages

Not Good

- Some codes are committed directly to master without reviewing
- Frequently changes of the response body

Improvement

- Should create two repositories with master and dev
- Need to split tasks on Trello board

Zijian Zeng

Good

- Having meeting frequently, catch up together almost every day.
- Introduce GitGuardian to scan credential in the codebase to prevent credential leaking

Not Good

- Front-end codebase sometimes is not consistent with our typescript coding style.

Improvement

- Should be familiar with typescript before contributing to front-end development.

Yunfei Jing

Good

- Set up branches for each main functionality and did professional code review
- Very fast feature implement cycle
- Can think of good design ideas about the front end (such as the explore page and the portfolio title)

Not Good

- Each member has different opinion about the UI/UX, did not communicate well before writing code
- Trello Board is not updated frequently enough

Improvement

- improve TypeScript coding style
- Assign each member his responsibilities clearly

Xiancheng Yang

Good

- Use Github “Pull Request” to review group members' code
- Group communication well, running debug session

Not Good

- Group member’s idea needs to discuss together to find the best one.
- Teamwork role not clear.

Improvement

- Assign works to each member clearly
- Add all member to vire code

Sprint 3

1. Sprint Planning
2. Stand-up Meeting Note

    a. Sprint 3-1 Stand-up
    b. Sprint 3-2 Stand-up
3. Sprint Review
4. Sprint Retrospective

Sprint 3 Planning

In sprint 3, most of the requirements from clients have finished. The development team will focus on refining code structure, debug, testing and adding some new features.

In sprint 3, the team aim to improve user experience, some little refines will be added, such as redirect to last viewing page after login.

The detail of the sprint 3 plan is shown in the following table,

Function	As a user, I can be redirected to my last viewing page after login, so that I will save time.
	As a user, I could click the links on the footer, so that I can view website policies.
	As a user, I could view the dashboard page, so that I can modify personal details.
	As a user, I could view the following user and followers on the dashboard page, so that I can find out how I am popular.
	As a user, I could explore new e-portfolios recommended by the system.
	As a user, I could upload background music for my e-portfolio.
	As a user, I could upload a cover image so that people can see this when clicking on my portfolio.
	As a user, I should be able to hide my portfolio from others or make it public to others.
	As a user, I could see my followers, so that I can know who are following me
	As a user, I could see who I am following users, so that I can manage my following
	As a user, I could change visibility (public/private), so that I can restrict access to private e-portfolio
Non-Function	Development Graph
	Screen Resolution (smartphone/ ipad suitable)
	New style of the Navigation bar (bases on clients' requirements)
	Add DNS record for eportfolio.tech
	Add footer content
	Deploy production environment
	Spring Boot Redis Cache for dev environment
	Deploy RabbitMQ for dev environment
	Deloy Redis on Redis manager
	Store Redis cache by Serialization
	Update Design Diagram

Sprint 3 Stand-Up Meeting

Sprint 3-1 Stand-Up Meeting Note

Date

2020-10-15

Participants

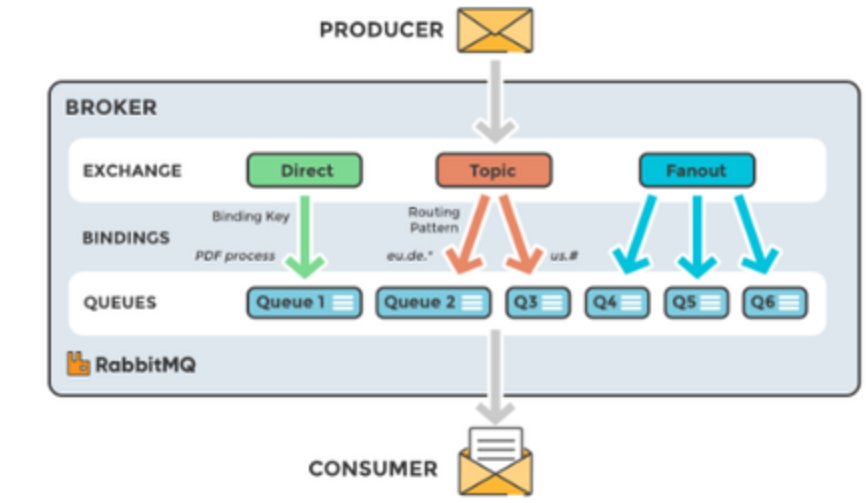
- @ Shuyang Fan
- @ Xiancheng Yang
- @ zijian Zeng
- @ Yunfei Jing
- @ Jiachen Zhou

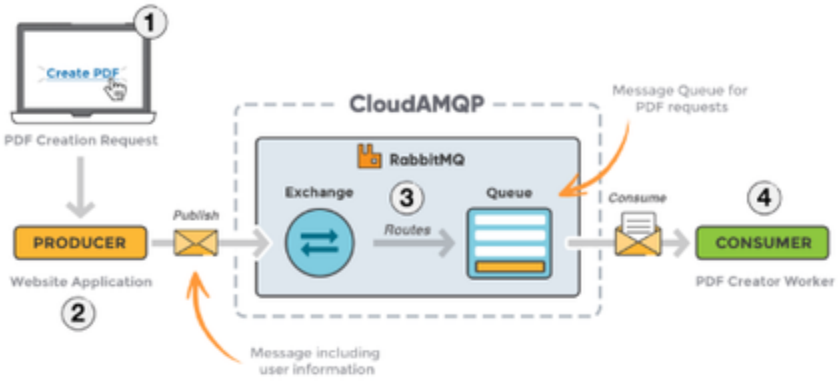
Host

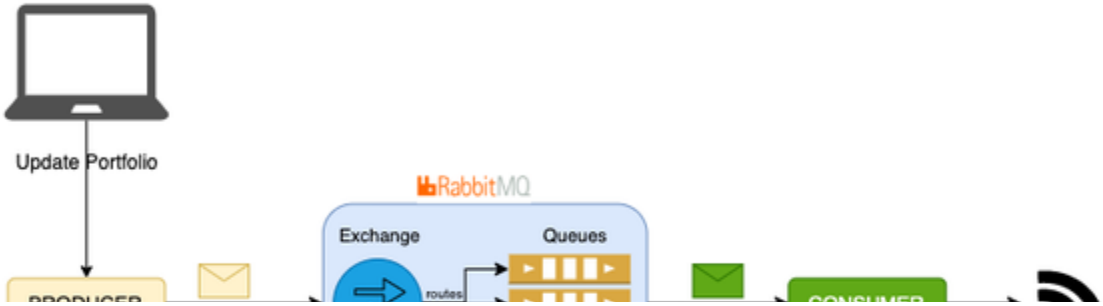
@ Shuyang Fan

Message Queue

Overview

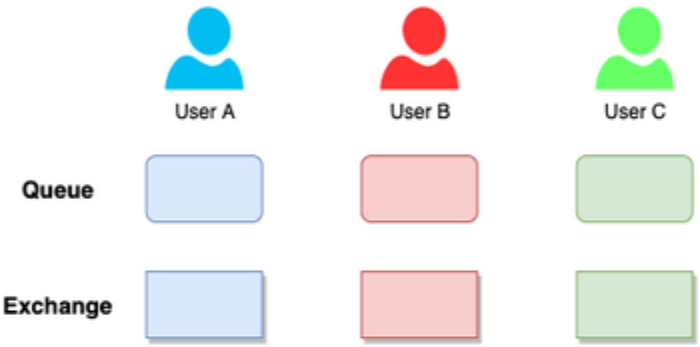






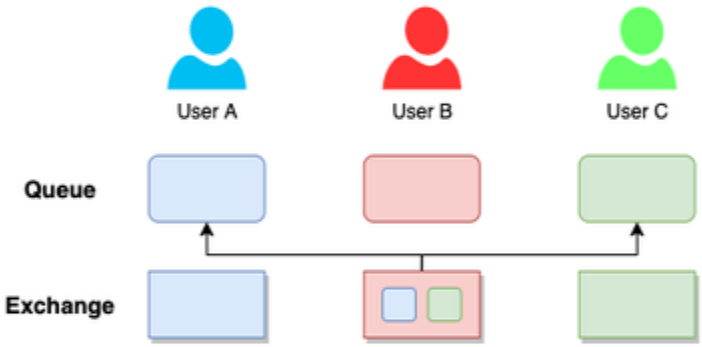
Signing up

When a new user is created, a new message queue and new exchange will be created for the user.
Both message queues and exchanges are initially empty.



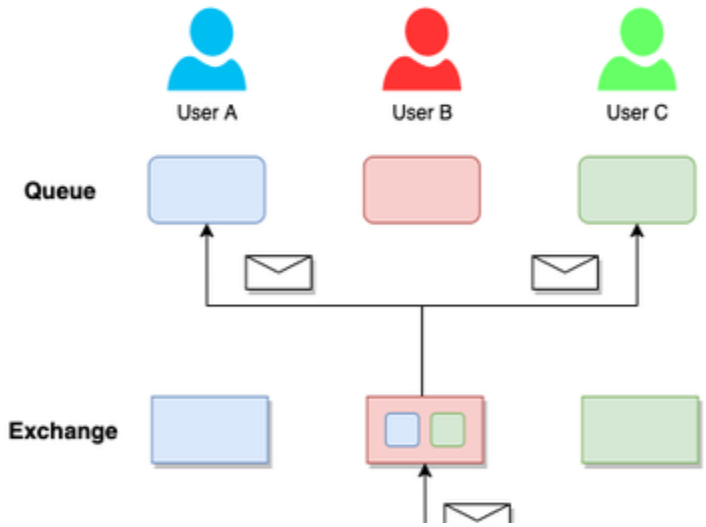
Follow

When user A follows user B, we bind B's queue with A's exchange.
This binding tell Rabbit MQ to route messages to A's queue for every message send to B's exchange.
Similarly, we create a binding when C starts following B.



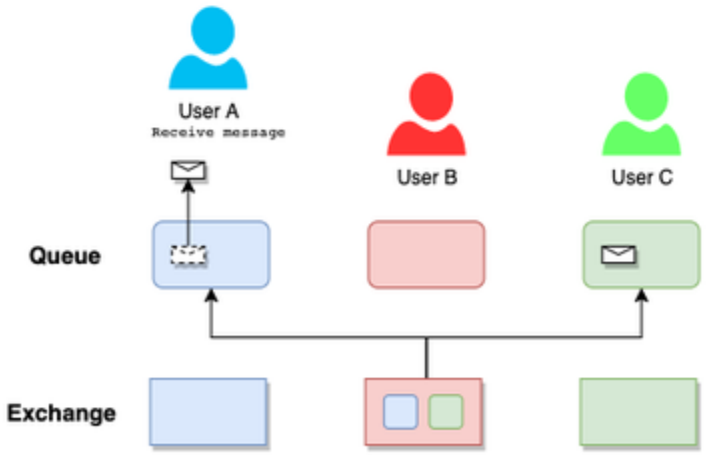
Produce Message

When user B updates his/her eportfolio, we send a message contains information of this update to B's exchange.
The exchange routes the message to user A's queue and user C's queue as they follows A.



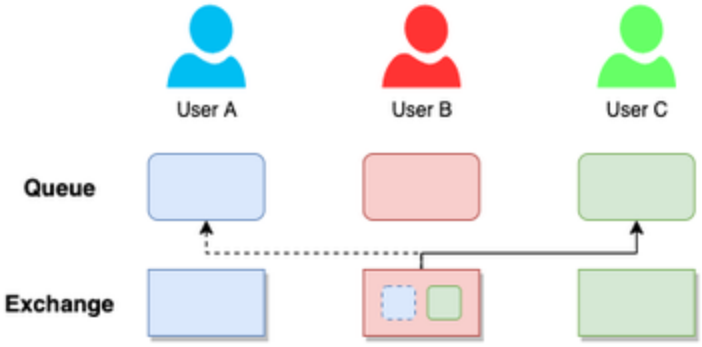
Consume Message

When User A access his/her feed on our website, we retrieve the message from message queue and "consume" it. Then we will gather related information from database based on the message and return it to the front end.



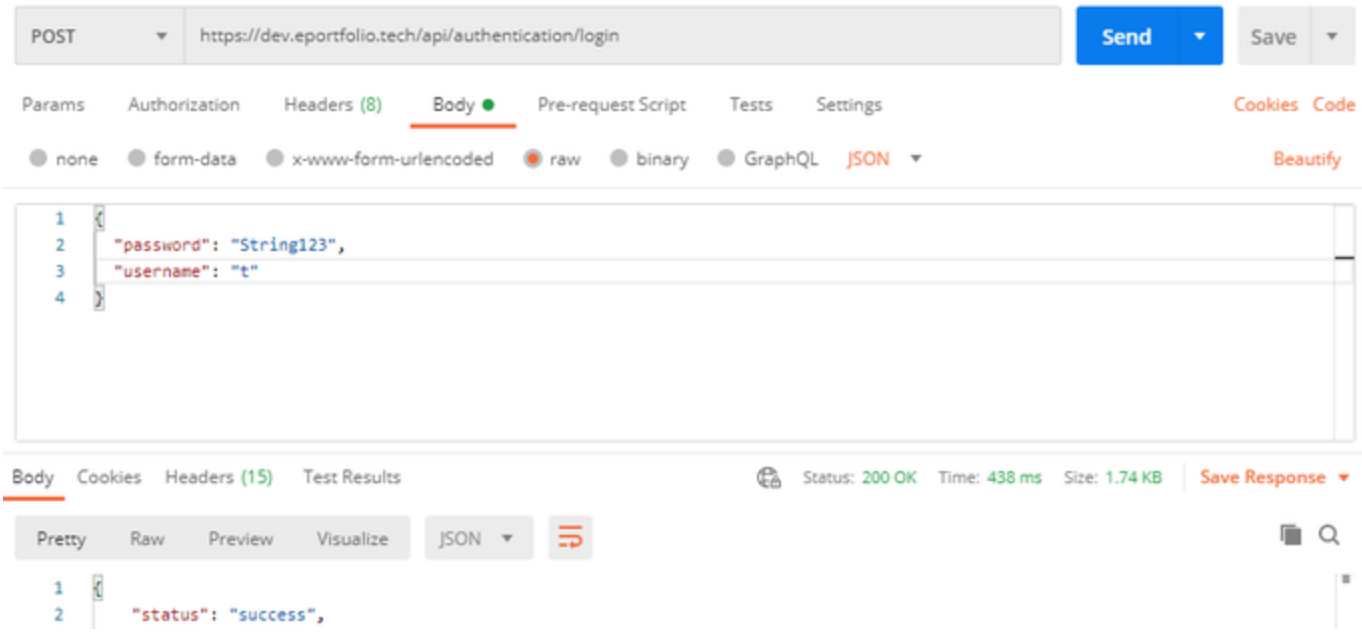
Unfollow

When user A unfollows user B, his queue will be unbound from user B's exchange. Therefore, user A will no longer receive message from User B in the future.

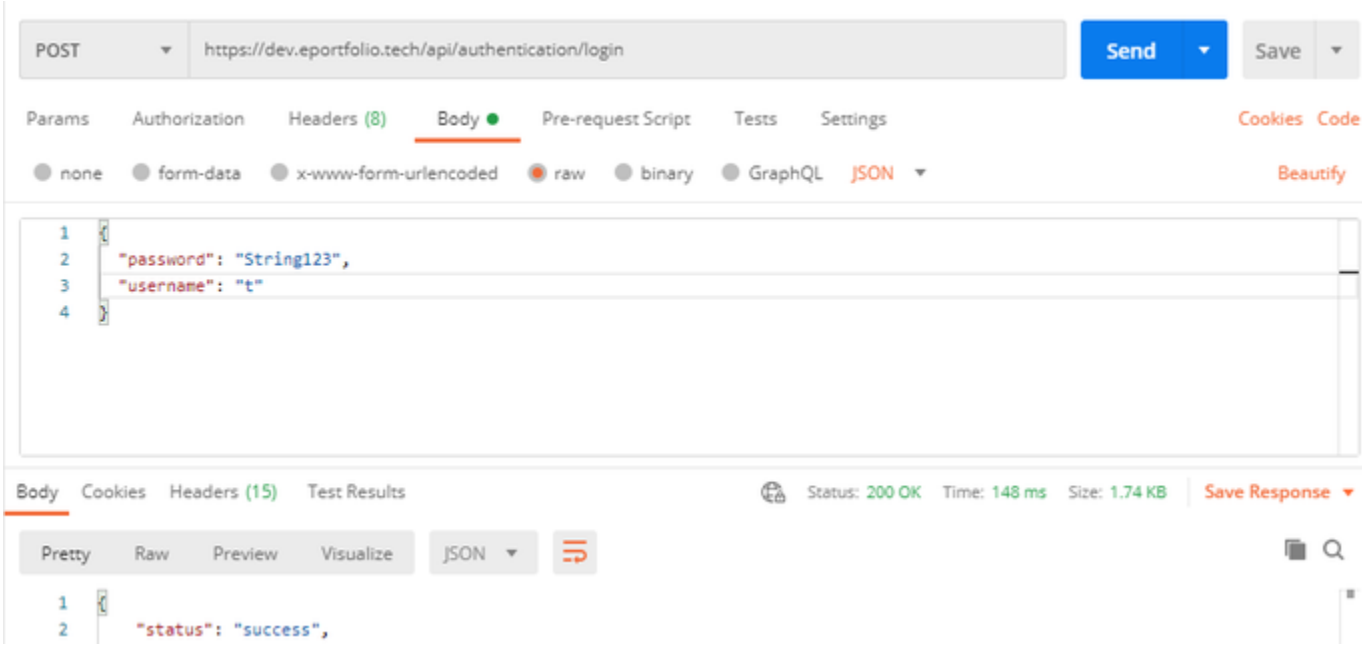


Redis Cache

> *"Before Caching"*



> *"After Caching"*



Progress

Presenter	Notes
Shuyang Fan	• Use RabbitMQ to implement activity functionality (i.e. send a notification to one's follower upon portfolio update) • Change Travis CI to build dev and prod docker image
Xiancheng Yang	• Follow function • Redirect to the previous page after login
Zijian Zeng	• Develop the Dashboard page • Following && Followers features

Yunfei Jing	• separate dev and production env at the front end • refine editor page
Jiachen Zhou	• Implement Redis Cache to store users and portfolios with TTL set up • Introduce Cache Service layer

Sprint 3-2 Stand-Up Meeting

Date

2020-10-22

Participants

- @ Shuyang Fan
- @ Xiancheng Yang
- @ zijian Zeng
- @ Yunfei Jing
- @ Jiachen Zhou

Host

@ Jiachen Zhou

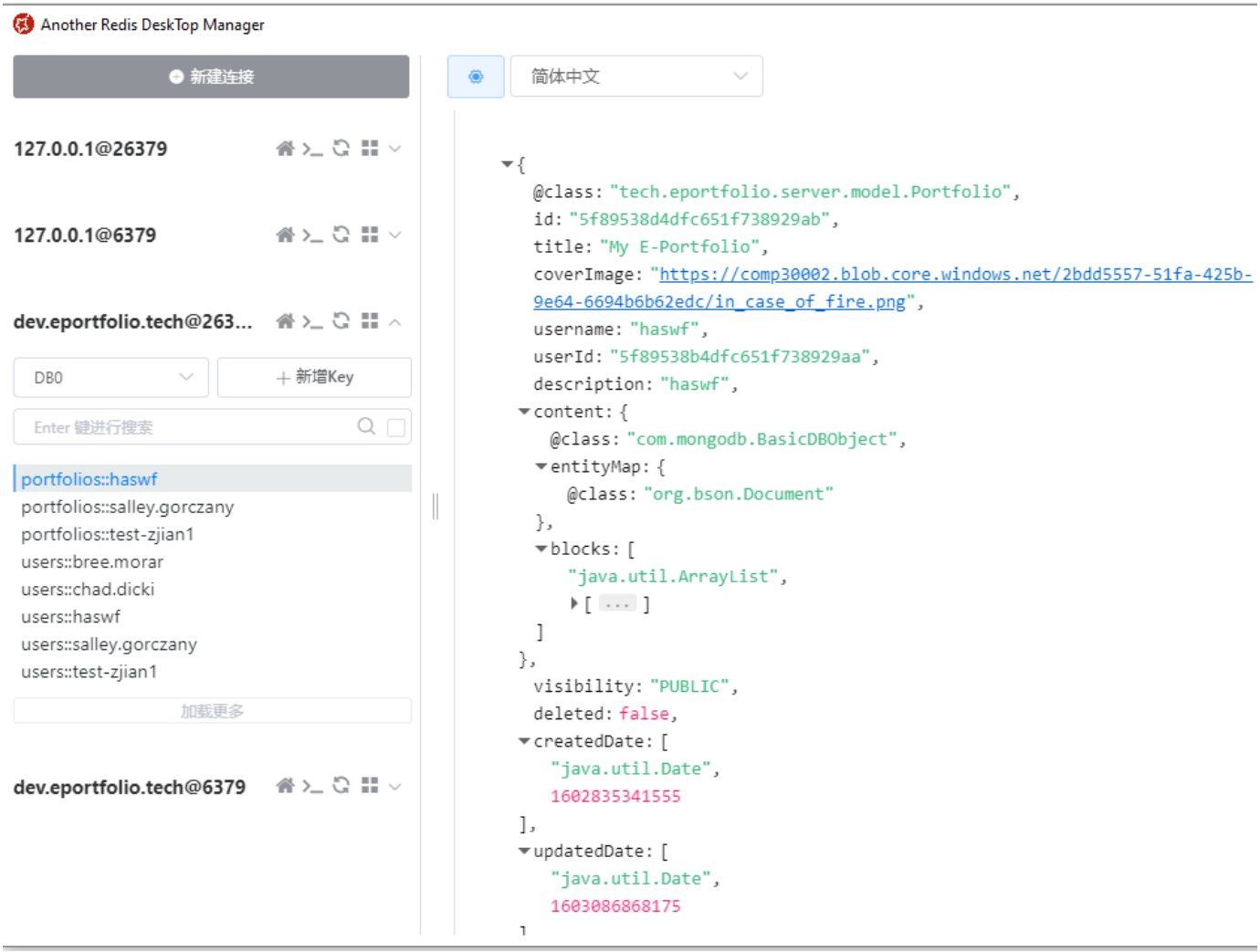
Work has been done in the last week

- Implementing last new feature: FEED
- Fix bugs (Cache, follow service)
- Run all tests to ensure everything works well
- Deploy from dev to prod
- Prepare for presentation

Store JSON in Redis Cache

The following figure shows how a service call is cached in Redis.

- Serializer model to change cache file from binary to JSON.
- Only users (TTL=2 hrs) and portfolios (TTL=1 hrs) will be cached.



Fix Potential Memory Leak

```
        List<User> sourceUsers = userService.findByIdIn(userFollows.stream().map(UserFollow::getSourceUserId).collect(Collec

        Map<String, Object> map = sourceUsers.stream().collect(Collectors.toMap(User::getUsername,
                e -> new HashMap<String, Object>() {
                    {
                        put("firstName", e.getFirstName());
                        put("lastName", e.getLastName());
                        put("avatarUrl", e.getAvatarUrl());
                    }
                }));
```

Use another way to initialize this instance. See Rule — 21 hours ago — L62

Bug — Minor — Open — Not assigned — 5min effort — Comment — leak

```
        List<Object> result = userFollows.stream()
                .map(userFollow -> new HashMap<String, Object>() {
                    {
                        put("user_follow", userFollow);
                        put("source_user", map.get(userFollow.getSourceUsername()));
                    }
                }).collect(Collectors.toList());
```

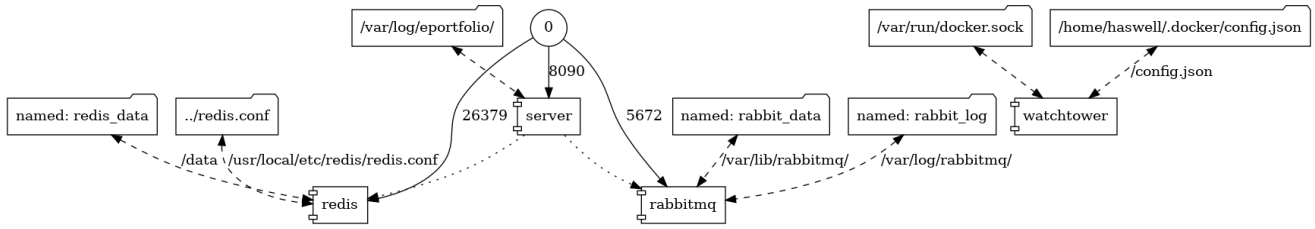
Use another way to initialize this instance. See Rule — 21 hours ago — L70

Bug — Minor — Open — Not assigned — 5min effort — Comment — leak

Docker-compose Diagram

The following diagram illustrates the services defined in docker-compose.yml for the development environment.

- The package symbol represents service.
- The folder symbol stands for volumes, which is the preferred mechanism for persisting data generated by and used by Docker containers.
- The dashed line represents volume mapping between service and volume. For example, volume `redis_data` is mapped into `/data` directory in `redis` container.
- The Filled line illustrates the dependency between services. In the following service, `server` depends on `rabbitmq` and `rediq`. Therefore, `redis` and `rabbitmq` will be initialised before spinning up the `server`.



Progress

Presenter	Notes
Shuyang Fan	• Deploy everything using Docker • Move sensitive configuration to .env • Tweak Nginx to cache SSL handshake • Implement Feed API
Xiancheng Yang	• Sprint 3 Documentation • Write & Add Footer Content • Working on Powerpoint
Zijian Zeng	• Implement Discovery page • Will recommend Portfolio, Tag and Following user's update
Yunfei Jing	• Add cover image in editor • Add change visibility in editor
Jiachen Zhou	• Store Redis cache by Serialization (.json) • Fix codes that may cause a memory leak • Refactor follow functionality

Sprint 3 Review Note

Quick Review

Based on the sprint 3 plan, we fully finished the user interaction part, in which users can like, comment, share, follow their own portfolio and view others'. In addition, we implement many non-functionality features, such as splitting the development and production environment.

Highlights

In this sprint, there are several things that we did well. our team has good communication, we increased our meeting frequency to 5-6 times per week. The front end and back end communicate together to solve current problems. In addition, each member did their part well and on time. We all attended every meeting and solve problems for each other. During the development, there were many debug sessions that we working together through zoom.

We finished all the scheduled works and did many other backend works to make our website more professional, such as implement cache to increase response time. In terms of testing, the coverage rate has increased and every new functionality has been well tested before being merged into the master branch. We create branches to develop new functions and pull requests to facilitate the code review process.

Reflect on Roadblocks and Improvement

There are also some challenges we can improve in the coming sprints. After we split dev and prod environment, the website was not stable and sometimes Travis was not working when merging dev branch to prod.

Sprint 3 Scheduled

Function	As a user, I can be redirected to my last viewing page after login, so that I will save time.	Completed
	As a user, I could click the links on the footer, so that I can view website policies.	Completed
	As a user, I could view the dashboard page, so that I can modify personal details.	Completed
	As a user, I could view the following user and followers on the dashboard page, so that I can find out how I am popular.	Completed
	As a user, I could explore new e-portfolios recommended by the system.	Completed
	As a user, I could upload background music for my e-portfolio.	Completed
	As a user, I could upload a cover image so that people can see this when clicking on my portfolio.	Completed
	As a user, I should be able to hide my portfolio from others or make it public to others.	Completed
	As a user, I could see my followers, so that I can know who are following me	Completed
	As a user, I could see who I am following users, so that I can manage my following	Completed
	As a user, I could change visibility (public/private), so that I can restrict access to private e-portfolio	Completed
Non-Function	Development Graph	Completed
	Screen Resolution (smartphone/ ipad suitable)	Completed
	New style of the Navigation bar (bases on clients' requirements)	Completed
	Add DNS record for eportfolio.tech	Completed
	Add footer content	Completed
	Deploy production environment	Completed
	Spring Boot Redis Cache for dev environment	Completed
	Deploy RabbitMQ for dev environment	Completed
	Deloy Redis on Redis manager	Completed
	Store Redis cache by Serialization	Completed
	Update Design Diagram	Completed

Sprint 3 Retrospective Note

Shuyang Fan

Good

- The team remains highly committed during the assessment season.

Jiachen Zhou

Good

- Keep introducing new features during the assessment weeks

- Secure system by putting sensitive information into property files
- Make websites highly adaptive to mobile devices

Not Good

- The randomly created images (for portfolios) may be sensitive or scaring
- Users may not be able to see new explores after swiping all existing portfolios once (No portfolio will appear twice)

Improvement

- Implement Spring cloud microservices to simplify development, deployment, and maintenance.
- Try to read more documentation to avoid making existing wheels
- Replace Random Recommender System with a better one such as content-based or NLP content-based

Zijian Zeng

Good

- set up Master branch for the production and Dev branch for development, improved quality control
- make good use of Git issues to keep tracks of bugs and suggestions

Not Good

- The searching feature doesn't work well sometimes.
- Some functionalities on the explore page would not be accessible if the screen height is too short.

Improvement

- Update our searching algorithm (elastic search).
- Try a better CSS screen resolution for the explore page.

Yunfei Jing

Good

- Made good use of Github Organization to separate different logic into different repositories
- Fixed all bugs we have encountered before the submission of the product
- Improved the user-interaction and made the interface more understandable

Not Good

- Some parts of the product are not fitted well with responsive design
- Templates are insufficient

Improvement

- Could use environment variables for deployment
- Could use more of the pull request and do the code review for front end
- Separate logic in Editor file since it's too complicated now

Xiancheng Yang

Good

- Testing functionalities large-scale
- Improve UI/UX for final refinement

Not Good

- Travis not stable
- Dev to Prod merge not professional

Improvement

- Make a clear note when merging from dev to prod environment