

Index

Sprint 1 Planning	2
Sprint 1 Review	2
Sprint 1 Retrospective	3
Sprint 2 Planning	3
Sprint 2 Review	4
Sprint 2 Retrospective	5
Sprint 3 Planning	5
Sprint 3 Review	6
Sprint 3 Retrospective	6

Sprint 1 Planning

Things that have to be done in sprint 1

1. Analyse requirements ✓
 - 1.1. motivational model
 - 1.2. user stories
2. Design a solution and seek agreement with the client ✓
 - 2.1. prototype + UI/UX
3. Build a product backlog ✓
 - 3.1. user stories
 - 3.2. prioritisation
 - 3.3. estimation
4. Decide on technology stacks ✓
 - 4.1. nodejs
 - 4.2. reactjs
 - 4.3. SQL database [on google cloud](#)
5. Design overall software architecture and components ✓
 - 5.1. page-page logic diagram
 - 5.2. class diagram
 - 5.3. er model
 - 5.4. *communication/state-machine diagram* [sequence diagram](#)
 - 5.5. [architectural model](#)
6. Decide on team roles ✓
 - 6.1. Product owner - Yunwei Wu
 - 6.2. Backend API - Haoyang Yu, Ruijie Pan
 - 6.3. Backend Database - Yifei Zhu, Danlei Mou
 - 6.4. Frontend - Yifei Zhu, Danlei Mou
 - 6.5. Documentation & diagrams/models - together
7. Build a baseline project with skeleton code structures & database
 - 7.1. decide product name
 - 7.2. initialize project
 - 7.3. backend APIs
 - 7.4. backend Database
 - 7.5. front end style sheet
 - 7.6. front end development

Sprint 1 Review

Below are things that have been completed in sprint 1:

1. Documentations and models
 - a. Axure prototype
 - b. Do/Be/Feel model
 - c. Motivation model
 - d. Product backlog → user stories + acceptance criteria

- e. Page-to-page logic diagram
- f. Class diagram
- g. Entity relationship model
- h. Sequence diagram
- i. Architecture model
- 2. Front-end development
 - a. Homepage
 - b. Login
 - c. Signup
 - d. User homepage
 - e. Others: Navigation bar, footer, redirections between pages
- 3. Back-end development
 - a. Database set up (Google Cloud + mySQL workbench)
 - b. User authentication
- 4. Others
 - a. Decided product name → Folihub

Sprint 1 Retrospective

What did we do well?	What could be done better?
<ul style="list-style-type: none"> - Prototype design to visualize and present ideas to clients, and set a standard for the product - Frontend and backend group working → learning from each other - Planning before acting → more structured - Works are distributed → everyone is contributing to the product - Decisions are made by everyone → voting on product color plan, product name etc. 	<ul style="list-style-type: none"> - Link between Trello and product backlog - Integration between groups → progress is not very unified - More prepared for the client meeting → powerpoints and rehearsal (we had this in sprint review meeting)

Sprint 2 Planning

2.1-2.5 Manage e-portfolio

frontend

- userHomepage → pop over to choose landscape/portrait
- userHomepage → eportfolio menu to delete this eportfolio (mainly about API)
- userHomepage → click a eportfolio to redirect to editing page
- EditingPage → save eportfolio as PDF (mainly about API)

backend

- Create new e-portfolio and add to database
- Delete an eportfolio
- Get all e-portfolios of particular user
- Get the information of an existing e-portfolio → cover page, last modified time
- Update the information of an existing e-portfolio
- Update the e-portfolio link (maybe?)

3.1-3.2 Edit e-portfolio

frontend

- e-portfolio editing page → quill.js, tool bar, visualizable change
- load the data uploaded → images

backend

- upload images → planning to save as data stream and store in the database
- edit text → add or update the text content in database
- retrieve the previous eportfolio data and send to frontend
- add new page → add new page to the database with the data send from frontend

Sprint 2 Review

Below are things that have been completed in sprint 2:

1. Documentations and models
 - a. Testing documentations
 - b. Deployment documentations
 - c. Update on user stories
 - d. Update on database er model (past models in the folder)
2. Front-end development
 - a. Edit folio template
 - b. Edit folio uploads
 - c. Edit folio buttons, add page
 - d. User homepage
 - e. Contact us
 - f. FAQ
3. Back-end development
 - a. get all eportfolios of a given user
 - b. rename a given eportfolio
 - c. get the content of a given eportfolio
 - d. create new page
 - e. save (update) page
 - f. delete page
 - g. get the content of a given page
 - h. create new eportfolio

Sprint 2 Retrospective

What did we do well?	What could be done better?
<ul style="list-style-type: none">- Code review process (with branch protections) to improve code quality- Start testing to ensure the product is robust and secure- Deployment to Heroku to allow access for external users- Changing documentation and Trello structures to make it clearer (according to the feedback from Sprint 1 and stand up meetings)	<ul style="list-style-type: none">- Apply the code review process for every requests, become more familiar with this process- Update the Trello board more frequently and create cards with all required information- Automated testing required for the product

Sprint 3 Planning

2.5 share e-portfolio

frontend

- generate new website
- send url to backend

backend

- receive url from frontend and update the database

2.6 export e-portfolio

frontend

- turns into pdf
- download

backend

- get all contents from given e-portfolio id

4.7 tutorial for using the app

frontend

- tutorial

Testing for existing and upcoming functionalities → see testing documentation at

https://docs.google.com/spreadsheets/d/1-wlu9MTsUhqct4dP7idWDdbHGg45FdmatDyInyoQG_c/edit#gid=0

Sprint 3 Review

Below are things that have been completed in sprint 3:

1. Documentations and models
 - a. Update on Testing documents
 - b. Update on Deployment documents for pipeline
 - c. Update on README
 - d. Presentation documents
 - e. Product report
 - f. Tutorial documentation of the app user guide
 - g. Uploading documents to github
2. Front-end development
 - a. Generate the new website
 - b. Send urls to backend
 - c. Download and conversion of PDF files
 - d. Layout modification for functionalities
 - e. Patch for bugs
3. Back-end development
 - a. Receive urls from front-end and update database
 - b. Handover contents that the front-end requests
 - c. Patch for bugs

Sprint 3 Retrospective

What did we do well?	What could be done better?
<ul style="list-style-type: none">- collaborations- communication with clients- user experience testing- more used to the code review process (learning)	<ul style="list-style-type: none">- more considerate when planning- more testing- automated pipeline for front-end