Project Plan

Vibe Check

Date	:	04/09/2024
Version	:	1.0
State	:	In Progress
Author	:	Nuno Dias

Version history

Version	Date	Author(s)	Changes	State
1.0	04/09/2024	Nuno Dias	First Draft	In Progress

Contents

1. Project assignment

1.1 Context

As a solo developer trying to learn more about programming and Web Apps, I decided to create an app similar to X (previously known as twitter). This will help me learn about Web Apps at scale, web sockets, media attachments, content moderation, user moderation, storing large amounts of small data (posts, likes, etc) and a lot more. This seemed to be a perfect learning project.

As previously mentioned this app will be similar to Twitter and be open to adding features modularly so I can squeeze in as much as I can with the time I have and learn a lot of different technologies. For example, if I have time, I would like to add some simple games in the style Facebook does it. This would be entirely optional and isolated in its own subsection.

1.2 Goal of the project

As briefly mentioned above the goal of this project is to help me learn new concepts, technologies and ways to implement certain features. I will also learn about APIs and isolating a projects layers (frontend & backend) even more than last year.

Ideally, by the end the project will result in a rudimentary twitter- like website with posts that you can add media to (images and music samples). As a user you will be able to post messages, like posts, save posts and reply to posts. All only while being logged into your account. I would also like to add more customization and settings this time around.

Finally, if time permits, I will add a subsection to my website that will host a couple simple games (similar to Facebook games). As well as exporting the project as a docker file for easy deployment anywhere.

The advantages of this project are that it covers a lot of different technologies. From data storage of complex objects with tree like structures (posts with various replies & replies to replies), to a user friendly UI, to auto-censoring posts and customizability (settings). It also has a lot of opportunities for exploring new concepts that aren't mandatory and so will not affect the project if I fail to implement them. These include P2P encryption, OAuth, linking & personalizing profiles, games and more.

1.3 Scope and preconditions

Inside scope:	**Outside scope:**
1 Website	1 Maintenance
2 Posts, Likes and Replies	2 API Keys for Music Databases
3 Accounts	3
4 Feed Algorithm	4
5 Basic Auto-Post Moderation	5
6 Adding Song samples to posts	6
7 Notifications	7

The Backend must be written in Java with the Spring boot framework.
Front end should be in HTML, CSS and React but the JS framework can be swapped out after deliberation.
Any CSS framework (or none) can be used.

1.4 Strategy

The first approach will be to use the Agile SCRUM methodology and work in sprints. This should allow flexibility and error correction as the project develops.

For the frontend we will stick to JS React, HTML and Tailwind CSS to speed up the page styling. The backend will be done in Java Spring boot.

Although there is a lot to do, there are also learning goals to check off for this semester so the first focus will be on getting the basic “twitter” functionality up and running. This includes accounts, posts, likes, replies and saves. Then implement an algorithm by trying to censor some posts and adding a report button, as well as 1 or 2 feeds.

Once checked with the “client” that all the requirements are met, Settings, Customization and upgrading the Interface are next. As well as trying to set 1+ APIs for encoding music samples when posting something.

The final step will be exploring the rest of the “Wishlist” by interest since at this point all tasks will be mainly done by self-interest. These include security, encryption, simple browser game hosting (similar to flash).

For planning and documentation Jira and Trello will be used to keep track of the backlog, sprints and goals, as well as word and google docs for documentation (Project Plan, URS, Testing, etc). For presentations at the end of sprints PowerPoint if necessary, but the main focus will be on practical demonstrations of the progress and improvements relative to the last sprint.

1.5 Research questions and methodology

<<
Describe the research questions that are most relevant to your project. For each research question, describe the approach and/or methodology. Use the Dot Framework to specify strategies and methods - see http://www.ictresearchmethods.nl for details.

Note that research is not only part of the initial phases (like analysis) of the project, but runs throughout the whole project. E.g., in the realization phases, you will probably do research in the Workshop and Lab context.

Realize that during the project your research questions may change, and that new ones will come up. That normal for any project, and is not a problem as long as you involve the right stakeholders, and keep your deliverables updated.
>>

(To Be Added)

1.6 End products

A. WebApp that includes a website for the frontend, and a backend to run all the logic and processing

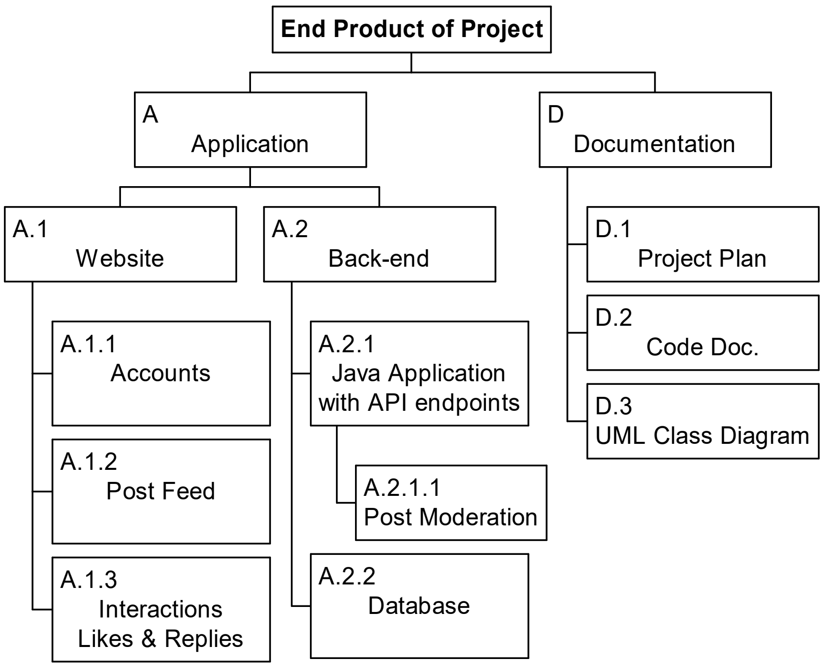
A.1. Front-end for WebApp that is user friendly. Written in HTML, CSS and JS React.
A.1.1. Accounts for user verification, moderation and personalization.
A.1.2. A feed for Users to browse through of recommended posts.
A.1.3. Interaction Actions for Users including liking, replying and saving posts

A.2. Back-end for WebApp that handles logic, the algorithm, data storage and more. Written in Java Spring boot.
A.2.1. Java application that handles all the logic of the program from user actions to data management to moderation algorithms.

A.2.1.1. A moderation algorithm that checks the posts text and possibly images for banned content
A.2.2. Database with all the information the App needs to run. Provides data permanence.

D. Documentation on project, the development process and how it works. Helpful for expansion and maintenance.
D.1. This Document. An overview of the development plan.
D.2. Documentation on how the code works. Includes Code comments and a possible helper document.
D.3. An UML Class Diagram that represents how the Classes, Packages and project works in general from a technical but high point of view. Useful for developers.

Project Breakdown Structure (PBS)



2. Project organisation

2.1 Communication

Ideally there will be 2 meetings a week for feedback on progress and alignment of objectives, so the project doesn’t derail. However these will be quick meetings and only brief overviews of progress.
So on top of these there will be end-of-sprint meetings every 3 weeks. These will be longer format, more organized and formal. They will cover all the progress of the past sprint and will be more effective at seeing if the direction the project is going is the right one.

Finally, these will be followed by an internal team meeting that will aim to process and apply the feedback received in the end-of-sprint meeting.

3. Activities and time plan

3.1 Phases of the project

1 Initial Planning

Reading of any documents provided by client and composition of first Project Plan draft. Interview of client asking questions created the in previous steps. Project Plan second draft as well as other documents (Requirements, MOSCOW, other) and more concrete planning for first sprint. Finalize plan by checking with client.

2 Sprints (3 weeks)

Every sprint will start with a short retrospective of the previous sprint followed by planning the next sprint and adding the corresponding tasks to the sprint backlog. Then tasks are mostly determined for the next 3 weeks but they can still be slightly changed depending on feedback.

3 End of Sprint

The last week is dedicated to finishing ongoing tasks, polishing the code a little for presentation, optionally creating some extra material for presentation such as a PowerPoint and a practice run of the demo to avoid crashes while presenting.

4 Final Presentation

Similar to the End of Sprint phase but more expansive. The last sprint will be dedicated to finishing all lose ends, testing and bug fixing to that the end product is stable as well as preparing a presentation going over all deliverables and functions in the end product. This presentation will also require more practice and thought put into it.

5 Wrap Up

After the final presentation all documentation will be updated and products exported into a final package to be delivered to the client.

3.2 Time plan and milestones

Phasing	Start date	Finish date
1 Sprint A	01-09-2024	11-10-2024
2 Sprint B	11-10-2024	08-11-2024
3 Sprint C	08-11-2024	29-11-2024
4 Sprint D	29-11-2024	20-12-2024
5 Sprint E	20-12-2024	17-01-2025

4. Testing strategy and configuration management

4.1 Testing strategy

<<Which testing strategy do you envision? E.g., on which levels will testing take place? Consider that you could choose unit, component, integration, system, or acceptance testing.

Justify your strategy, and also set goals where relevant. E.g., percentage code coverage for the relevant unit tests. For each of the planned tests, indicate what will be automated and what not.

Also think of quality testing setups like, e.g., Sonarqube.
>>

4.2 Test environment and required resources

<< Describe the test environment. E.g., do you envision a DTAP (Development, Testing, Acceptance, Production) environment. Can you make use of a CI/CD environment or will you develop your own?

It often helps to use a picture to visualize the test environment.

If you already know, describe which resources are required for realization and testing. Think of hardware, cloud environments and specific tooling required for development and testing.
>>

4.3 Configuration management

<< Describe the project approach with respect to version management (e.g. your GIT repository). This might include things like tooling, branching strategy, promotion-, release- and baseline strategy.

Also, when relevant, think of a mechanism to deal with change requests and problem reports.>>

(To be Added)

5. Finances and risk

5.1 Risk and mitigation

Risk	Prevention activities	Mitigation activities
1 Loss of contact with Teacher	Frequent updates so time between loss of contact is minimal. Discuss short but also long-term plans with Teacher	Updates over email and with an alternative guide (teacher)
2 Sickness	Add an extra week of work for delays	Online work and redistribution of tasks in planning
3 Data Loss	Version Management with Git and cloud backup with Gitlab on Fontys Servers	Checking for latest code version on other computers and re-merging
4 Expectations different from result	Frequent updates and sprint-end meetings to align/re-align goals and expectations with client	Reorganizing priorities and planning
5 Scope Creep	Frequent internal meetings (retrospective on sprints) to assess the feasibility of the project, timelines and goals MOSCOW & Phasing that prioritizes critical parts of project first	Reducing the scope on less critical parts of the project
6 Time Loss	Phasing that prioritizes critical parts of project first Thorough planning with some breathing room for bug fixing, testing and other problems that can be relied on if needed	Reorganizing priorities and planning