

Project Plan

RRS Hamster

Date	:	04/0302024
Version	:	0.1
State	:	Finished
Author	:	Nuno Dias

1. Version history

Version	Date	Author(s)	Changes	State
0.1	04/032024	Nuno Dias	First version	Finished

Distribution

Version	Date	Receivers

Contents

1. Project assignment.....	4
1.1 Context.....	4
1.2 Goal of the project.....	4
1.3 Scope and preconditions.....	4
1.4 Strategy.....	5
1.5 Research questions.....	5
1.6 End products.....	5
2. Project Organisation.....	6
2.1 Stakeholders and team members.....	6
2.2 Communication	6
3. Activities and time plan.....	7
3.1 Phases of the project.....	7
3.2 Time plan and milestones.....	7
4. Testing strategy and configuration management	8
4.1 Testing strategy.....	8
4.2 Test environment and required resources	8
4.3 Configuration management	8
5. Finances and Risk.....	9
5.1 Project budget.....	9
5.2 Risk and mitigation.....	9

2. Project assignment

2.1 Context

RSS Hamster is a Project made by an Indie developer for his own use and hopefully some monetization might come out of it if the project evolves well enough. Right now there are no clients or Stakeholders to satisfy except himself so the direction to head in is quite flexible.

This however does not mean the Project can go anywhere. To keep himself in track the developer has imposed some evaluation criteria and objectives that *must* be meet before their corresponding deadline.

The Project will have a desktop app mainly for administration, a Web App for users and databases to store data. The App will revolve around organizing RSS feeds, being customizable and even having metadata for each feed such as ratings. This will then be used to create a recommendation algorithm.

2.2 Goal of the project

The goal of this project is to have one place where one can keep updated with the internet. In this age of oversaturated content and algorithms recommending all the content you consume having your own RSS Feeds nicely organized for you in one place and all the updates about the people you want to follow in one place is extremely useful.

Ideally this project aims to create a situation where you can find new content but also keep up to date with people you have been following for a long time in one place, the internet is a big place.

The advantages of this are a centralized notification hub where one is sure they wouldn't miss or forget someone because "they forgot to login to Facebook this week".

The value this project aims to add can be described in two words **time** and **quality**. Reduce the time you spend making sure you are *up-to-date* and reduce the quantity of poor taste recommendations made by algorithms.

2.3 Scope and preconditions

Inside scope:	Outside scope:
1 Web App	1 Integration with other systems (API)
2 WinForms App	2 Any kind of manipulation/filtering of RSS feeds
3 Accounts	3 Universal support (the standard formats will be covered but no more)
4 Possibility to add RSS feeds	4 Long term support
5 Rating & comment functions	5 Monetization of the website (just no)
6 Recommendation Algorithm	6 Physical Infrastructure
7 SQL Server for saving and loading	7

The technologies being used have already been decided (ASP.net Razor Pages & WinForms in C#) and so if something cannot be implemented with these frameworks and their respective languages it will not be implemented.

There are a lot of features that were promised and due to lack of time I will not go above and beyond. The App will most likely lack some polish specially in the design department.

2.4 Strategy

When given the chance the agile method will be used simply so that adjustments to the project are made in a timely manner. As a solo developer there won't be issues related to dividing the workload, planning for integration, etc. But the possibility of my vision being different than the clients is still very real. The agile method will provide feedback on the direction the project is\should be going in.

Before the Agile method is learned the waterfall method will have to do.

2.5 Research questions

- **How do the current RSS feed viewers look/work? How can we improve on their design?**
- What is an efficient way of managing, storing and displaying these feeds?
- How will the algorithm provide meaningful recommendations?
- What will functionality will a user need?
- How will the Website and WinForms App work together?

2.6 End products

- Ideation Document
- Project Plan
- User Requirements Specification (URS)
- Test Plan & Reports
- SQL Database
- WinForms App
- Web App

3. Project organisation

3.1 Stakeholders and team members

Name	Contact	Role and functions	Availability
Sachin Bhardwaj (Fontys)	sachin.bhardwaj@fontys.nl	Advisor	Weekly
Marco Meulenbroeks (Fontys)	m.meulenbroeks@fontys.nl	Supervisor	Weekly
John Wijnen (Fontys)	john.wijnen@fontys.nl	Supervisor	Weekly
Customers	NA	Users	NA
Nuno Dias	540432@student.fontys.nl	User & Developer	Always

3.2 Communication

- In Person meetings – Semi-regular reports to supervisors on the progress of the project
- Canvas – Submission of incomplete application every couple weeks for evaluation
- Team – Obsidian – Timeline, tasks, deadlines, etc.

4. Activities and time plan

4.1 Phases of the project

The timeline of the project spans 16 weeks, the final product will be delivered before **June 14th 2024**.

- Project Plan & Ideation Document – 08.03.2024
- UML, App v1 – 22.03.2024
- App v2 – 29.03.2024
- URS, Database, Website live, App v3, Test Plan – 05.04.2024
- Unit test, App v4 – 26.04.2024
- App v5 – 10.05.2024
- Test Plan, App v6 – 17.05.2024
- Updated version of all deliverables, App Final version – **14.06.2024**

For an agile project describe how the artefacts are planned. E.g., length of sprint (with justification), organization of stand up, demo, retrospective.
>>

Phasing	Start date	Finish date
1 Initial Documentation	01.03.2024	22.03.2024
2 App	15.03.2024	14.06.2024
3 URS, Test Plan, (Unit) Tests	29.03.2024	17.05.2024
4 Database & Website	29.03.2024	14.06.2024

5. Testing strategy and configuration management

5.1 Testing strategy

Unit testing is essential for ensuring the correctness and reliability of individual code units. It helps detect and fix defects automatically. One of the most simple tests but also most important. *“A simple spell but quite effective” – A wise man*

Manual Integration testing to test that each function fits with the pre-existing code and doesn't break anything.

5.2 Test environment and required resources

The testing environment will be the tools provided by the “Visual Studio” IDE and possibly GitLab if it is time-efficient. There will be both automated testing and manual testing in debug mode to look for errors.

5.3 Configuration management

GitLab is used as the primary version control system and project management platform. Each feature will be developed in a separate branch and merged after passing the appropriate testing. As a solo developer no complicated measures will be implemented for merging or other actions.

6. Finances and risk

6.1 Project budget

NA (Not Applicable)

6.2 Risk and mitigation

Risk	Prevention activities	Mitigation activities
1 Sickness	Flexible timeline that accounts for extra time in case of sickness	None (Explain to supervisor)
2 Client Expectations different from result	Regular meetings and feedback from users	Deliverables clearly stated in Project Plan
3 Loss of Data	Git for version control	Be transparent with Client and Tutor + Recover as much as possible
4 Server Problems	Local Backups and Backup of SQL Db as script	Go to ISSD/Complain