

Project Plan

TRIP SPLITTER/MERGER AND ENRICHER

CROSSYN

Date	:	24/09/2021
Version	:	0.1
State	:	Unreleased
Author	:	Dobri Trifonov, Joran van de Moosdijk, Robert Enuta, Stanislav Petkov, Tobias Halomoan

Version history

Version	Date	Author(s)	Changes	State
0.1	24/09/2021	Dobri Trifonov, Joran van de Moosdijk, Robert Enuta, Stanislav Petkov, Tobias Halomoan	Early draft of project plan.	Draft
0.2	09/11/2021	Dobri Trifonov, Joran van de Moosdijk, Robert Enuta, Stanislav Petkov, Tobias Halomoan	Update of the whole Document.	Draft

Distribution

Version	Date	Receivers
0.1	08/10/2021	Gupta, Roopali R.
0.2	09/11/2021	Gupta, Roopali R. , Bram van Herwijnen

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	4
1.5 Research questions and methodology	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.....	8
4. Testing strategy and configuration management	9
4.1 Testing strategy	9
4.2 Test environment and required resources.....	9
4.3 Configuration management	9
5. Risk.....	10
5.1 Risk and mitigation	10

1. Project assignment

1.1 Context

Crosssyn is a company that made a trip creator, it makes trips based on data. With the project we want to make sure the trips are based on the correct data so that users can see the correct start and end points and routes they drove. This will allow them to use it in case of an accident or will help for companies to see how much an employee drove with a car. The application will display this in a simple way that is easy to use.

1.2 Goal of the project

The goal of our application is to create trips from provided datasets by the Crosssyn company.

We have the goal of visualizing and enriching the datasets through a software solution.

Through linear interpolation we will be able to automatically remove data deviation without human input. Using the location data, it would be possible to visualize the data on a graphical interface for a simple end-user-like experience.

The lightweight application we will develop will keep track and link trip data to the vehicle's driver account.

This will allow Crosssyn to keep track of recorded information on a per account basis.

The main function of our application will be to ease up the workflow the recorded vehicle datasets.

1.3 Scope and preconditions

Inside scope:	Outside scope:
1 An application that will take the dataset and processes it into trips	1 The ability to create trip grades from processed data
2 Quality assurance and testing for the application (unit testing, CI/CD, test plan and test report)	2 The processing of data in the user case of an accident on the road
3 User support through a manual	3 Further usage of processed trip data for business or research purposes
4 Documentation that justifies choices made during development (research docs, project plan, architecture document)	4 Working with streams of data
5 Creating and designing a database	

1.4 Strategy

For this project we are using the agile and iterative Scrum methodology of development. Our team consists of 5 people and each sprint a new scrum master is assigned to make sure that all agile practices are followed correctly by the developers. The development of the product is divided into 5 sprints, each lasting 3 weeks. At the start of the project a product backlog is set up by our team with all the project-goals in mind. In the beginning of each sprint a sprint planning meeting is held, where every team member participates in setting the goals for the sprint (creating a sprint backlog), then the sprint goals and priorities are discussed with our client and adjusted according to his feedback. During each week of the sprint our team has 3 stand-up meetings, where current progress and goals are discussed. At the end of each sprint, a sprint review meeting with our client is held to review the progress made during the sprint. After the review meeting, the goals that have not been met are redirected to the product backlog again and a sprint retrospective meeting is held, where everyone reflects on the sprint and can express their opinion on what the team should start, stop, and continue doing. Then the process begins all over again until the completion of the project.

We choose the Scrum framework, because it allows us to make changes on the fly as not everything is predefined, therefore we can consider and implement the client's feedback immediately. Testing is done constantly as well and not left at the very end of the project, which allows the team to identify and prevent potential issues much earlier on. The daily meetings allow for the current progress to be easily tracked as well. Lastly, using this methodology allows the team to put out new releases very frequently, which keeps the end-users happy, as they see constant progress and give feedback as well.

1.5 Research questions and methodology

1. Which database architecture is the most suitable for our project?
 - 1.1. What are the structural differences?
 - 1.2. What are the advantages and disadvantages of each database type?
 - 1.3. How scalable is each type?
- The information on this topic will be collected through community research and literature study.

2. Project organisation

2.1 Stakeholders and team members

Name	Abbreviation	Role and functions	Availability
<i>Bram van Herwijnen</i>	<i>BvH</i>	<i>Product Owner: define goals and relay functional requirements for project. Evaluates progress after each iteration.</i>	<i>In person: during product demonstration In correspondence: during workdays</i>
<i>Roopali Gupta</i>	<i>RG</i>	<i>Project Consultant/Teacher: Oversees project development and scrum. Gives instructive feedback during and after every iteration.</i>	<i>In person: On every Wednesday or through appointment. In correspondence: Through appointment</i>
<i>Tobias Halomoan</i>	<i>TH</i>	<i>Developer.</i>	<i>Available on every weekday through appointment / in class.</i>
<i>Dobri Trifonov</i>	<i>DT</i>	<i>Developer</i>	
<i>Stanislav Petkov</i>	<i>SP</i>	<i>Developer</i>	
<i>Robert Enuță</i>	<i>RE</i>	<i>Developer</i>	
<i>Joran van de Moosdijk</i>	<i>JvdM</i>	<i>Developer</i>	

2.2 Communication

Communication between the product owner (BvH) and the development team are mainly done through email (bram.van.herwijnen@crossyn.com), with the scrum master as the main communicator for the group's progress or if any inquiries are needed. Weekly meetings will also be held with the teacher (RG) as to show progress and receive feedback/ask questions regarding the scrum methodology and project. A Jira board is implemented for communicating project and sprint backlogs, as well as assigning specific tasks/issues/stories to each developer for every sprint. Stand-up meetings for the group are assigned three times during the week (Tuesday, Wednesday, Friday) as to stay updated on the development progress and discuss solutions when issues arise. A Discord group chat is also formed for the group for easy online communication.

3. Activities and time plan

3.1 Phases of the project

Using the Scrum methodology, we are going to split the project into 5 sprints, each lasting 3 weeks. During the first sprint we are analyzing the problem, planning the project, creating base documentation, and implementing the chosen functionalities for this sprint. At the end of the sprint the MVP is released, and a review of the progress is made.

Then there are 4 more sprints, where the process of development is as follows: plan – implement – test – release – review.

After all the sprints are completed, a final evaluation/reflection is made, a final release is scheduled, the project is wrapped up and the product is handed over to the client.

All the progress, goals and work division is managed and tracked via the online agile management environment “Jira”.

In the table below, you can see in detail what are the activities each sprint consists of.

Phasing	Effort
1 Sprint Planning	-Items, that should be achieved during this sprint are chosen from the backlog and put onto the sprint backlog as goals for the sprint - The workload is properly divided among the developers
2 Stand up meetings	-Each week three stand up meetings are held by the developers on Tuesday, Wednesday, and Friday, where the problems, progress made and the goals are discussed among the group members.
3 Iteration	-Implementation of unfinished features from the previous sprint is continued. -Implementation of the planned features for the sprint is made.
4 Testing	-Unit testing is performed on the newly implemented features -The whole system is tested after the new features have been integrated
5 Demo Release	-A demo with the newest stable version of the application is provided to the client -The client provides feedback based on the newest release
6 Sprint Review	-The progress made during this sprint is inspected and the backlog is reviewed to check which tasks have been completed and which are still unfinished
7 Sprint Retrospective	-A final meeting is held at the end of the sprint, where the team reflects on the sprint and discusses what everyone should start, stop and continue doing for the next one.

3.2 Time plan and milestones

Phasing	Effort	Start date	Finish date
Sprint #1	First draft of the project plan completed Online agile management environment set up 80%+ of the product backlog filled in Acceptance criteria for the project architecture met MVP released Planning for the second sprint finalized	18.09.2021	08.10.2021
Sprint #2	Impact on society document created Initial test plan and report documents created Project plan updated First draft of architecture document completed New functional requirements implemented Planning for the third sprint completed	09.10.2021	05.11.2021
Sprint #3	Test plan and report updated Architecture document updated Project plan approved by PO Cultural awareness report completed New functional requirements implemented Planning for the fourth sprint completed	06.11.2021	26.11.2021
Sprint #4	Test plan and report updated Architecture document updated Project plan updated New functional requirements implemented Planning for the final sprint completed	27.11.2021	17.12.2021
Sprint #5	Test plan and report finalized Architecture document finalized Project plan finalized Application finalized All source code zipped and sent to PO Every other relevant document submitted	18.12.2021	14.01.2022

4. Testing strategy and configuration management

4.1 Testing strategy

For our Unit Testing we plan to make a per use case. We think that having a large amount of single use case will simplify the testing process and improve the codes future debugging and maintainability. Tests that validate several things may become too complex and hard to maintain in the future. To be able to have maintainable code we will stick with having one (or if needed two) assertions per test.

Every test will be self-contained and not interact with other components to verify validity.

Taking the route of having many isolated, single case unit test will be very time consuming so our group will only promise 60% code coverage with the potential to expand it to 80% in the future.

We plan to base our tests on user stories that will give us an idea of what to focus on.

4.2 Test environment and required resources

For testing we have set up a CI/CD pipeline for automated testing and easier bug fixing. Later, we will make use of a docker container to further speed up testing and assure the quality of the project. Classes and methods will be implemented alongside unit tests to assure as little errors are left in the code as possible.

User acceptance tests will also be a part of the testing process. The goal of the project is confirming that the system is comparable and upholds the initial requirements agreed upon. After the projects reaches a stated deemed acceptable for user testing, a small number of people, preferably with different backgrounds and interests, will be asked to use the product and give feedback on state of the application.

For conducting these tests, we will use a computer running Windows OS. No additional tools or resources are required at this time, but if any necessities arise, we will discuss about what tools are needed and the results will be talked about with the client.

4.3 Configuration management

For our Git repository we plan to use the Branch Per Feature strategy.

Every basic feature will get its own branch. All these branches will merge into a

Development branch. After throughout testing and review on how the branches interact with each other the development branch will be deemed stable. The Development branch will then be merged into a separate, Stable branch on which we will have a stable version of our application.

On the Stable branch only bug fixes should be made.

When it is time for a client review and feedback, the stable version will be merged into main and represent the current working release of our app.

5. Risk

5.1 Risk and mitigation

Risk	Prevention activities	Mitigation activities
1 Time risk	Overestimating the time, we must spend on our tasks and creating a project schedule in the form of a timeline for example	Setting up a clear schedule and setting a time limit on meetings could help. .
2 Clarity	Make sure someone takes the lead in some regards if no one does and have clear tasks for everyone.	If there isn't any clarity on when things should be done or who does what, plan a meeting, sort everything out and document it in some sort of way if needed.