CITY TRAFFIC SIMULATION

PROJECT PLAN

Fontys
University of Applied Sciences

VERSION HISTORY

The following table provides information about the development of the Project Plan, including the main changes in the Project Plan (adding and editing information) and the dates for the approval of this information:

VERSION #	Prepared by	Revision Date	Approved by	Reason
1.0	The team	11.02.2020		Project Plan Draft

CONTENTS

1.INTRODUCTION

1.1 PURPOSE OF PLAN

The SDM Project Plan will provide a definition of the project, including the project's goals and objectives. Additionally, the Plan will serve as an agreement between the following parties: Project Sponsor, Project Manager, Project Team, and other personnel associated with and/or affected by the project.

The Project Plan defines the following:

- Project purpose
- Project statement
- Business and project goals and objectives
- Management plan
- Risk assessment
- Roles and responsibilities
- Assumptions and constraints

2.PROJECT STATEMENT

2.1 FORMAL CLIENT

Formal client:

Name: Andrius Kuprys

Email: a.kuprys@fontys.nl

Telephone number: 0885076989

Tutor:

Name: Ragu Japaga

Email:

Telephone number:

2.2 TEAM

Team name: Hyena Crossing
Team members:

Name	Role	Occupation	Email
Yousef Abu Zahra	Team Leader	ICT & Software Engineering student at Fontys	y.abu.zahra@student.fontys.nl
Alina Baciu	Team Secretary	ICT & Software Engineering student at Fontys	a.baciu@student.fontys.nl
Mustafa Farah	Team member	ICT & Software Engineering student at Fontys	mustafa.farah@student.fontys.nl
Borislav Pavlov	Team member	ICT & Software Engineering student at Fontys	b.palvov@student.fontys.nl
Ghazi Abdul Fattah	Vice Leader	ICT & Software Engineering student at Fontys	g.fattah@student.fontys.nl

2.3 CURRENT SITUATION

Our client works for Sim Software Inc. which is a company for building simulation software. Currently there are numerous car accidents in the city because of badly made configuration of the roads and no traffic lights.

2.4 PROJECT JUSTIFICATION

In order to make the city a safer environment, Mr. Kuprys hired us to build a simulation software in which they can test out different scenarios to optimize the traffic.

2.5 PROJECT PRODUCT

The product of our project is going to be a software application that simulates real life traffic. This application is going to run in windows environment and will be built using visual studio as IDE and C# as a programming language.

2.6 PROJECT DELIVERABLES & NON-DELIVERABLES

Deliverables:

- A simulation application for real life traffic.
- Project plan.
- Test plan.
- URS (User Requirements Specification).
- DDS (Design Document Specification), which is going to include UML class diagrams.
- Manual
- Source code
- Working prototypes at the end of each iteration.
- Presentation.
- Process report.

Non-Deliverables:

- Training for the use of application.
- Any additional software that the application might need.

2.7 PROJECT CONSTRANTIS

These are the limitations that our project will have to abide by:

- **Time**
  We need to finish the project in a window of 18 weeks.

- **Programming language**
  The application will be written in C# mainly.

- **Version control software**
  We need to have a version control software in place.

- **Running environment**
  The application should run on Windows environment.

2.8 RISK ASSESSMENT

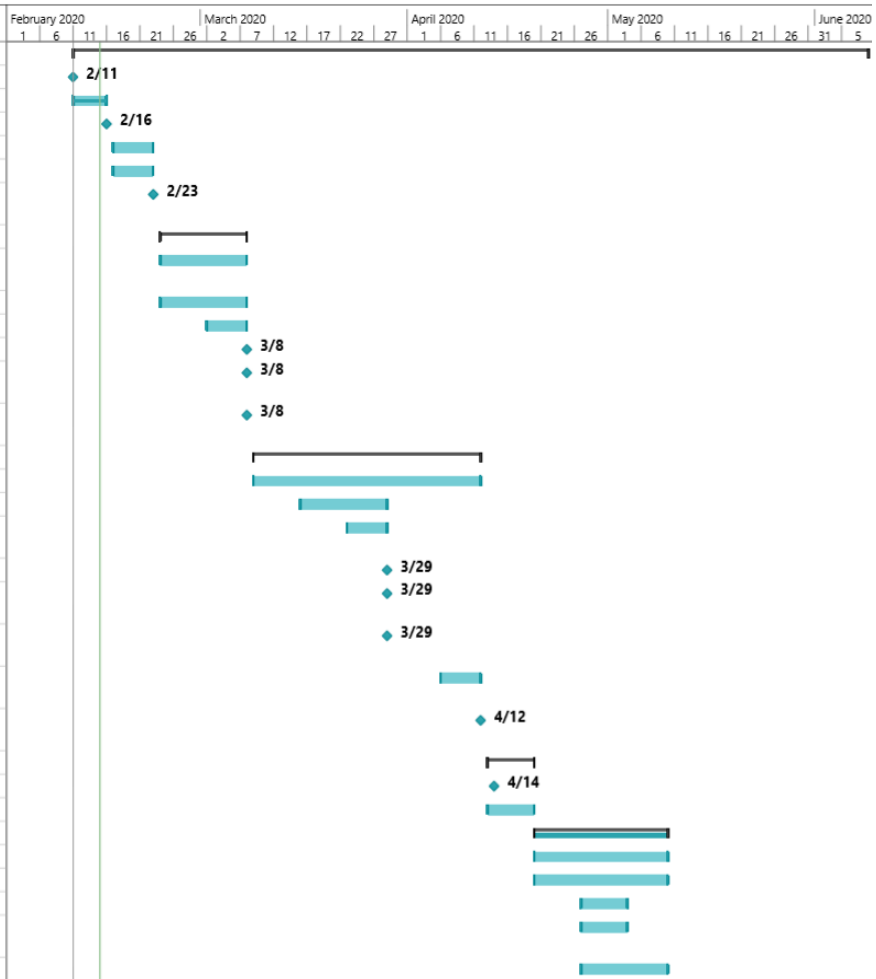
Risks
In every project, there always will be risks involved. Here's a list of the risks that we think our project might face:

RISK	DESCRIPTION	PROBABILITY	IMPACT	PREVENTIVE MEASURES
Falling behind on deadlines/milestones	Activities taking longer than expected	**Medium**	**High**	Having frequent meetings to keep up with each other and the client. Also having milestones in place which have a deliverable for each one of them.
Communication problems	Conflict within team members. Absence and other disciplinary issues.	**Low**	**Medium**	The team leader has to make sure that everyone is on the same page.
File loss/corruption	Losing some or all of the files, or having them corrupted.	**Low**	**High**	Using version control software to make sure that everything is backed up.

Final product not meeting all promised features	The final containing wrong requirements or not all of the requirements we promised to deliver.	**Medium**	**Medium**	Keeping track of all the requirements in our documents. Also getting feedback from our client after each iteration and during periodic meetings.
Final product lacking quality	Final product is buggy or inaccurate	**Medium**	**High**	Performing periodic testing during and after each iteration.
Lack of experience within our team	Our team lacking the background knowledge needed to finish the project, whether software or real life knowledge.	**Medium**	**Medium**	Doing research before and during the project.

3.PROJECT PHASING

	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		**Traffic Simulation Project**	**85 days**	**Tue 2/11/20**	**Tue 6/9/20**	
2		Kick off	0 days	Tue 2/11/20	Tue 2/11/20	
3		Project Plan v1	5 days	Tue 2/11/20	Sat 2/15/20	
4		Deviler Draft Project Plan	0 days	Sun 2/16/20	Sun 2/16/20	
5		Project Plan v2	6 days	Mon 2/17/20	Sat 2/22/20	
6		URS for iteration 1	6 days	Mon 2/17/20	Sat 2/22/20	
7		Deliver Project Plan Final Version	0 days	Sun 2/23/20	Sun 2/23/20	
8		**Initial Phase**	**10 days**	**Mon 2/24/20**	**Sun 3/8/20**	
9		Update URS for iteration 1	11 days	Mon 2/24/20	Sat 3/7/20	
10		Plan for Iteration 1	11 days	Mon 2/24/20	Sat 3/7/20	
11		Devision Report	6 days	Mon 3/2/20	Sat 3/7/20	
12		Deliver Plan For Iteration	0 days	Sun 3/8/20	Sun 3/8/20	
13		Deliver Updated Version of URS	0 days	Sun 3/8/20	Sun 3/8/20	
14		Deliver Work Division Report	0 days	Sun 3/8/20	Sun 3/8/20	
15		**Iteration 1**	**25 days**	**Mon 3/9/20**	**Sun 4/12/20**	
16		Code/Debug	26 days	Mon 3/9/20	Sat 4/11/20	
17		Plan for iteration 2	11 days	Mon 3/16/20	Sat 3/28/20	
18		Update Work Devision Report	6 days	Mon 3/23/20	Sat 3/28/20	
19		Deliver URS for iteration 1	0 days	Sun 3/29/20	Sun 3/29/20	
20		Deliver Plan for iteration 2	0 days	Sun 3/29/20	Sun 3/29/20	
21		Deliver Updated Work Devision Report	0 days	Sun 3/29/20	Sun 3/29/20	
22		Create UML Class diagram(s)	6 days	Mon 4/6/20	Sat 4/11/20	
23		Deliver UML Class Digram(s)	0 days	Sun 4/12/20	Sun 4/12/20	
24		**Calibration Session**	**5 days**	**Mon 4/13/20**	**Sun 4/19/20**	
25		Deliver Proof Of Concept	0 days	Tue 4/14/20	Tue 4/14/20	
26		Create Deisgn Document	6 days	Mon 4/13/20	Sun 4/19/20	
27		**Iteration 2**	**16 days**	**Mon 4/20/20**	**Sat 5/9/20**	
28		Code/Debug	16 days	Mon 4/20/20	Sat 5/9/20	
29		Unit Tests	16 days	Mon 4/20/20	Sat 5/9/20	
30		Plan for iteration 3	6 days	Mon 4/27/20	Sun 5/3/20	
31		Update Work Devision Report	6 days	Mon 4/27/20	Sun 5/3/20	
32		Update Design Document	11 days	Mon 4/27/20	Sat 5/9/20	



34		Deliver Plan for iteration 3	0 days	Sat 5/9/20	Sat 5/9/20
35		Deliver Deisgn Document	0 days	Sat 5/9/20	Sat 5/9/20
36		Deliver Updated Work Devision Report	0 days	Sat 5/9/20	Sat 5/9/20
37		Deliver Prototype	0 days	Sat 5/9/20	Sat 5/9/20
38		**Iteration 3**	**16 days**	**Mon 5/11/2(**	**Sun 5/31/20**
39		Code/Debug	16 days	Mon 5/11/2(	Sun 5/31/20
40		Unit Tests	16 days	Mon 5/11/2(	Sun 5/31/20
41		Process Report	6 days	Mon 5/18/2(	Mon 5/25/20
42		Update Work Devision Report	6 days	Mon 5/11/20	Mon 5/18/20
43		Update Design Document	11 days	Mon 5/11/2(	Mon 5/25/20
44		Deliver Final URS	0 days	Sun 5/31/20	Sun 5/31/20
45		Deliver Process Report	0 days	Sun 5/31/20	Sun 5/31/20
46		Deliver Deisgn Document	0 days	Sun 5/31/20	Sun 5/31/20
47		Deliver Updated Work Devision Report	0 days	Sun 5/31/20	Sun 5/31/20
48		Deliver Final Product	0 days	Sun 5/31/20	Sun 5/31/20
49		Prepare Presentation	6 days	Mon 6/1/20	Mon 6/8/20
50		Presentation	0 days	Tue 6/9/20	Tue 6/9/20

3.1 SKILLS

In order to complete our project, these are the skills that our team needs to have:

- Project Management
- Programming skills in C#
- Testing skills
- Risk Management
- Database Administration
- User interface design skills.
- Quality Management
- Administrative / Secretarial Skill
- Communication skills

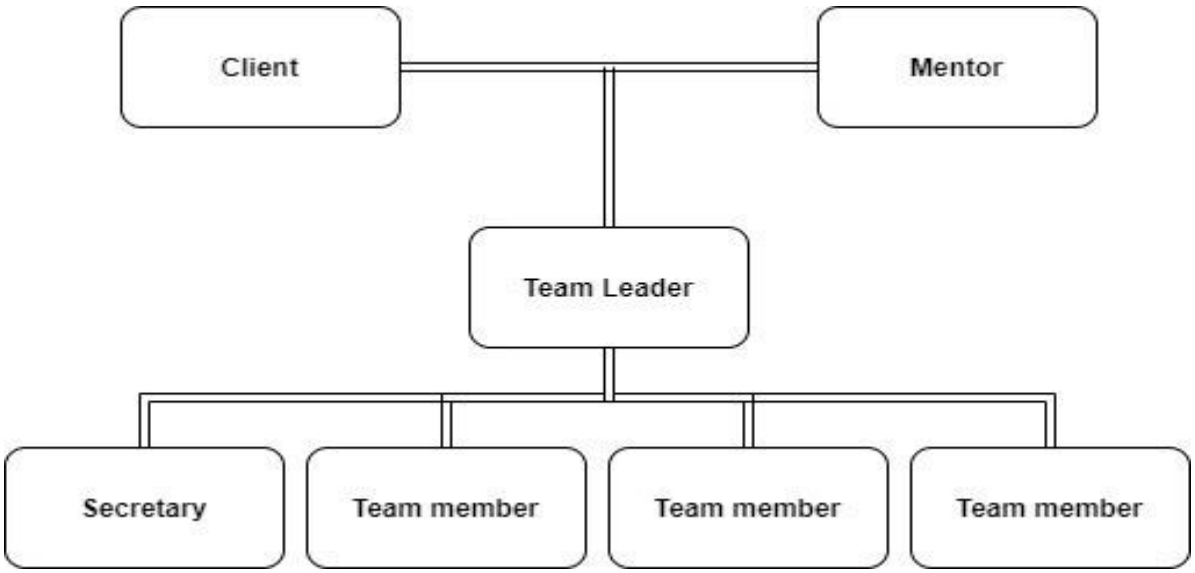
3.2 QUALITY

Our software application must be able to fulfill all the requirements specified by our client. It also needs to be delivered on time.
The interface of our application should be interactive and easy to use. It must implement a form of assistance to help guiding new users.
After each iteration, we need to have a working prototype that adheres to our client's and test users' expectations and feedback.

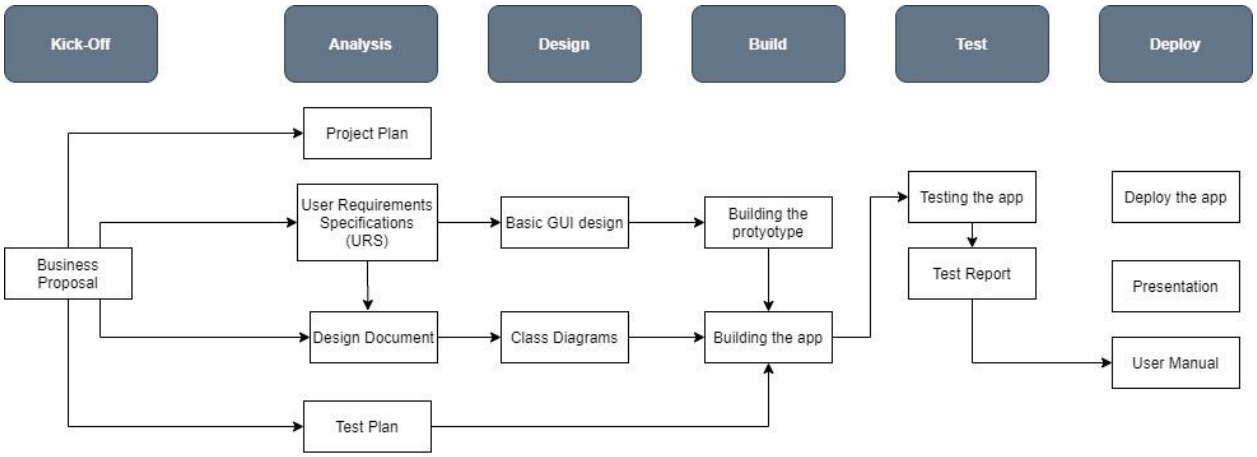
Our team is going to ensure that any bugs we might face during development will be dealt with, and that no errors will be displayed during runtime.
Alongside all of this, the documentation that we will provide should be clear and easy to follow.

ROLE	RESPONSIBILITIES	PARTICIPANT(S)
Team leader	• Make some project decisions • Communicating with the client • Resolve conflict within team • The same responsibilities as a team member	Yousef Abu Zahra
Vice Leader	• Replaces the leader in case of absence/sickness • The same responsibilities as a team member	Ghazi Abdul Fattah
Secretary	• Keep track of the meeting minutes • Update the process document • The same responsibilities as a team member	Alina Baciu
Team member	• Create applications • Design interfaces • Documentation	Mustafa Farah Borislav Pavlov
Tutor	• Give feedback • Track the project progress	Andrius Kuprys
Client	• Approve project decisions • Provide the requirements • Gives feedback	Ragu Japaga

3.3 ORGANIZATION



3.4 PHASES & MILESTONES



Milestones are the benchmarks for our project. They indicate that we are on the planned track to finish the project. Each milestone results in at least one deliverable.

Milestone 1: Kick off phase (week 1 - 2)

Activities:

- Team forming
- Research on topic

- Preparation for meeting with client

Deliverables:
- Project plan
- Proposal for an application
- Concept version URS Document

Milestone 2: Initial phase (week 3 - 4)

Activities:
- Discuss project plan
- Create concept version of plan for iteration 1
- Updated version of project plan & URS

Deliverables:
- Updated version of URS
- Final version of plan Iteration 1
- Work division report

Milestone 3: Iteration 1 (week 5 - 9)

Activities:
- Code

Deliverables
- Final URS for iteration 1
- Final version of plan for iteration 2
- Source code of proof of concept
- Proof of concept
- Updated version of work division report
- UML Class diagram(s) & non-trivial sequence diagram(s) of proof of concept

Milestone 4: Calibration Session (week 10)

Activity:
- Create design document
- Calibration session on how to proceed
- Update URS
- Create design document
- Test report

Milestone 5: Iteration 2 (week 11 - 13)

Activities:

- Code
- Finalize design document iteration 2
- Creation of a test plan for iteration 2
- Work on the second iteration of the product
- Updating work division report
- Updating user requirements specification

Deliverables:

- Final URS & design document
- Final version of plan for iteration 3
- Source code of prototype
- Unit tests of prototype
- Prototype
- Updated version of work division report
- Test report

Milestone 6: Iteration 3 (week 14- 16)

Activities:

- Code
- Creation of the design of the final iteration
- Creation of a test plan for iteration 3
- Work on the final iteration of the product
- Updating work division report
- Updating user requirements specification
- Create a user manual

Deliverables:

- Final URS & design document
- Source code of final product
- Final product
- Final version of work division report
- Process report (including work division)
- User-Manual
- Test report

Milestone 7: End phase (week 17-18)

Activities:

- Create a presentation for our product
- Meeting for the final grade

Deliverables:

- Presentation about the application.

3.5 PROJECT APPROACH

In this section, we are going to specify which approach we are going to take for both the development process and managing the project:

3.6 DEVELOPMENT APPROACH

For this project we will be following the Agile methodology to develop the software. We will be working in iterations and deliver a working prototype at the end of each iteration.

3.7 OVERALL PROJECT APPROACH

The overall project approach will be the scrum methodology. The work will be completed in short cycles and the team will meet weekly to discuss tasks and roadblocks that need clearing. All of the work required will be stored in the product backlog or on the scrum board.
The work will be done in an incremental, iterative approach.

FUNCTIONALITY	PROGRESS	START DATE	END DATE
ITERATION 1			
Create road elements			
Remove road elements			
Add intersections			
Have a default number of cars automatically generated			

Create custom layout

Start simulation

Show real-life traffic

Adjust the green time of the traffic light

Stop simulation

Pause simulation

ITERATION 2

Save as file

Open file

Add pedestrians

Remove pedestrians

Control the number of cars generated

ITERATION 3

Simulate traffic flow based on road sensors

Store custom layout in database

Open custom data from database

Increase/Decrease pedestrian traffic

Change the speed of the simulation

User can sign up for an account

3.8 MOSCOW

Our project is going to contain many features, these features carry different weights: some of them are a must have, while others would make a nice addition but would not cause our project

to fail if they were to be left out. And some others will not be implemented at all due to the limitations/scope of the project.
In the following table, you can see the list of the features our project is going to contain, and how important each one of them is to our project' success.

Legend for MoSCoW Table: **M** = must have **C** = could have
S = should have **W** = won't have

FEATURE	M	S	C	W
Show real-life traffic movement	✓			
Adjust the green time of the traffic light	✓			
Start simulation	✓			
Stop simulation	✓			
Pause simulation	✓			
Create custom layout	✓			
Create road elements	✓			
Remove road elements	✓			
Add intersections	✓			
Have a default number of cars automatically generated	✓			
Clear the screen of all intersections and elements	✓			
Save as file		✓		
Open file		✓		
Add pedestrians		✓		
Remove pedestrians		✓		

Control the number of cars generated		✓		
Simulate traffic flow based on road sensors			✓	
Store custom layout in database			✓	
Open custom layout from database			✓	
Increase/Decrease pedestrian traffic			✓	
Change the speed of the simulation			✓	
User can sign-up for an account			✓	
User can log-in/out			✓	
Generate report with information about current traffic			✓	
Have help section			✓	
Pre-defined environments for traffic			✓	
Record the simulation				✓
Change truck density				✓
Inserting different variety of vehicle				✓
Extra traffic signs				✓