Requirement Specification

Editor: Julius Kokko Ekholm

Version 1.1

Status

Reviewed	Erik Frisk	2018-11-29
Approved	Erik Frisk	2018-11-29

PROJECT IDENTITY

2018/HT, Cascar Group
Linköping University, Dept. of Electrical Engineering (ISY)

Project Group Members

Name	**Responsibility**	**Phone**	**Email**
Andreas Lundgren	Project Leader (PL)	070-022 56 44	andlu901@student.liu.se
Julius Kokko Ekholm	Documentation (DOC)	070-304 21 42	julek017@student.liu.se
Adam Ahlgren	Software (SW)	070-796 83 71	adaah731@student.liu.se
Karl Gudmundson	Testing (TST)	070-578 84 82	kargu357@student.liu.se
Anton Kullberg	Software (SW)	073-959 97 71	antku588@student.liu.se
Jonathan Hanses	Design (DES)	076-260 77 47	jonha594@student.liu.se
Andreas Larsson	Information (INF)	076-815 58 03	andla514@student.liu.se
Sten Magnusson	Hardware (HW)	073-766 99 43	stema100@student.liu.se

Email list for the whole group: cascargroup@gmail.com
Web site: *under development*

Customer: Lars Nielsen, lars.nielsen@liu.se
Customer contact: Erik Frisk, erik.frisk@liu.se
Course leader: Daniel Axehill, daniel.axehill@liu.se
Supervisor: Pavel Anistratov, pavel.anistratov@liu.se

Contents

Document history

Version	Date	Changes	Sign	Reviewed
1.1	2018-11-29	Minor adjustments to requirements 20, 40, 46, 47, 48, 68, 70, 71, 73.		
1.0	2018-09-26	First version.		
0.2	2018-09-20	Major revision.		
0.1	2018-09-17	First draft.		

List of Figures

List of Tables

1 Introduction

This requirement specification is a part of the project course TSRT10 CDIO Project (Automatic Control) at Linköping University, Sweden. Final year engineering students apply for a specific project related to automatic control and sensor informatics. In general, the projects of the course are closely linked to research in the fields or companies operating within these areas.

The project that this requirement specification is written for is related to both automatic control and sensor informatics. More specifically, the project treats the area of autonomous vehicles in complex scenarios. A detailed description of the project and its objectives will be presented in Sections 1.2 and 2. The included subsystems will be defined and the limitations and requirements of the project will be presented. Additionally, the reliability and maintainability aspects will be included.

1.1 Partners

The project has a customer, course examiner, tutor and a few other advisers listed in Table 1.

Table 1: Project advisers.

Role	Name	Email
Customer	Lars Nielsen	lars.nielsen@liu.se
Customer Contact	Erik Frisk	erik.frisk@liu.se
Course Leader	Daniel Axehill	daniel.axehill@liu.se
Supervisor	Pavel Anistratov	pavel.anistratov@liu.se
Expert	Björn Olofsson	bjorn.olofsson@liu.se
Laboratory Engineer	Tobias Lindell	tobias.lindell@liu.se

Furthermore, the project members can be seen in Table 2.

Table 2: Project members.

Responsibility	Name	Email
Project Leader	Andreas Lundgren	andlui901@student.liu.se
Documentation	Julius Kokko Ekholm	julek017@student.liu.se
Software	Adam Ahlgren	adaah731@student.liu.se
Testing	Karl Gudmundson	kargu357@student.liu.se
Software	Anton Kullberg	antku588@student.liu.se
Design	Jonathan Hanses	jonha594@student.liu.se
Information	Andreas Larsson	andla514@student.liu.se
Hardware	Sten Magnusson	stema100@student.liu.se

1.2 Goal and Purpose

The goal is to construct autonomous vehicles that can perform complex tasks. This will be made possible as the cars will be able to communicate with each other and react to the information received. The possible scenarios that will be examined are platooning, roundabouts, overtaking and four-way intersections. A more detailed description of these are found in Section 2.1. As these scenarios are so complex the focus is on delivering well functioning solutions for at least two of the given situation rather than decent solutions to all the situations.

As earlier mentioned, this project is in the final year for engineering students at Linköping University. Its purpose is to combine the knowledge from control system and programming courses to be able to take a project idea to a finished product.

1.3 Usage

This project will both be of use to examine the groups knowledge and be a starting point for a new control system course at Linköping University held by Erik Frisk and Björn Olofsson.

1.4 Background Information

There is currently a large interest from both the industrial and academical world for autonomous vehicles. The interest and development of more complex and intelligent autonomous vehicles have manifested itself in competitions, research initiatives and companies which aim to bring the transportation industry closer to being fully autonomous.

1.5 Definitions

Abbreviations and their definitions that are used throughout this requirement specification can be seen in Table 3.

Table 3: Abbreviations and definitions.

Abbreviation	Definition
ROS	Robot Operating System
RP	Raspberry Pi
VPS	Visionen Positioning System
SLAM	Simultaneous Localization and Mapping
GUI	Graphical User Interface
RC	Radio-Controlled
LIDAR	Light Detection and Ranging
BDM	Behavioural Decision Making

This requirement specification has three different priority levels of requirements:

Priority 1 Basic requirements. These requirements have to be fulfilled.

Priority 2 Extra requirements. If all requirements with priority 1 are fulfilled, and if the budget allows it, resources will be put to fulfill these requirements.

Priority 3 Long-term requirements. Requirements at this level can be seen as something the group recognizes could be beneficial for the product, but might not be feasible within the current budget. These requirements will only be worked on if all requirements at level 1 and level 2 are fulfilled.

The requirements are also specified as **Original** or **Revised**, in the latter case accompanied by a date. All original requirements are defined in version 1.0. If there are any re-negotiations regarding the original requirements later on in the project, these will be marked as revised with a date to confirm when the change was made.

Throughout this document there are some terms that are used that can easily be misinterpreted. To avoid this, each word is listed here and described.

Term	Description
Route	A route is a way which is to be travelled. It can either be determined by the user manually or by using computational algorithms. It is defined by two or more predefined waypoints. Also, a route is determined by the coordinates (x,y).
Trajectory	A trajectory is a calculated path which a body travels along on e.g. a route. It is hence determined by a specific route, but includes the time as well i.e. at what time point the body is at the coordinates (x,y). More explicitly, a trajectory is defined by (x(t),y(t)).
Static object	Anything that does not move is considered a static object.
Static environment	An environment where every object is static.
Dynamic object	Anything that moves is considered a dynamic object.
Dynamic environment	An environment where at least one object is dynamic.

2 Overview of the System

The product consists of multiple autonomous vehicles that will be able to handle complex scenarios in a road network using route planning, behavioural planning, motion planning, control theory and communication. The vehicles should be able to handle these scenarios in a safe and efficient manner, making sure collisions are avoided at all costs. Most scenarios are ordinary in a standard road network, such as overtakes, roundabouts and intersections, but the vehicles should also be able to drive in a convoy, using communication to limit the distance between them. See Figure 1 for an overview of the general system architecture of one car.

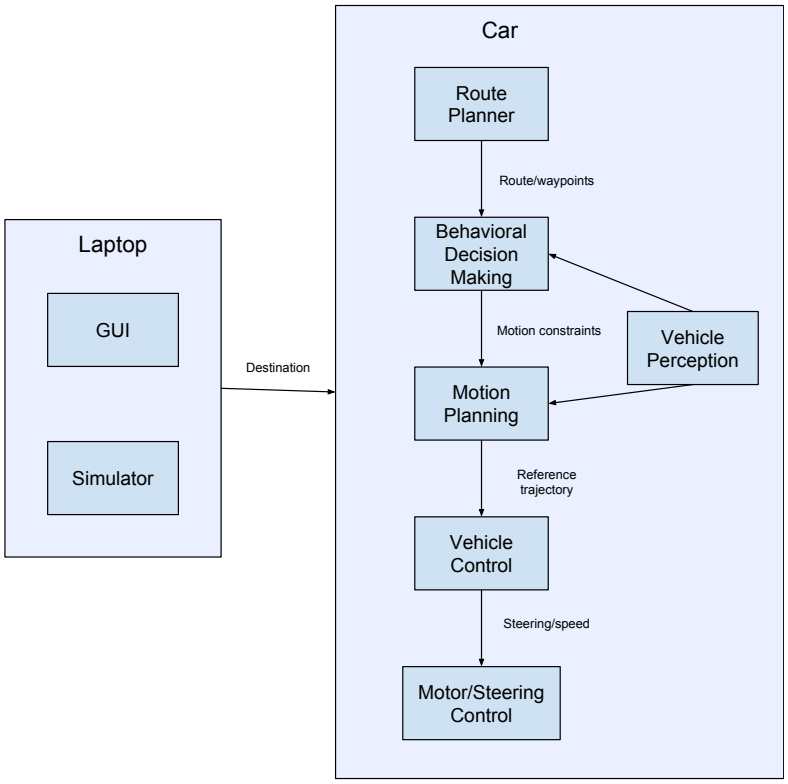

Figure 1: System architecture of one car.

2.1 Scenarios

The project will examine four different scenarios where the first two are prioritized over the others.

2.1.1 Platooning

Platooning refers to a lead car taking different actions and there being trailing cars that follow the lead car at a small distance so as to improve fuel efficiency and safety. The lead car makes all the decisions and propagates that information backwards so that if it needs to brake the trailing cars brake as well. A schematic overview of platooning can be seen in Figure 2.

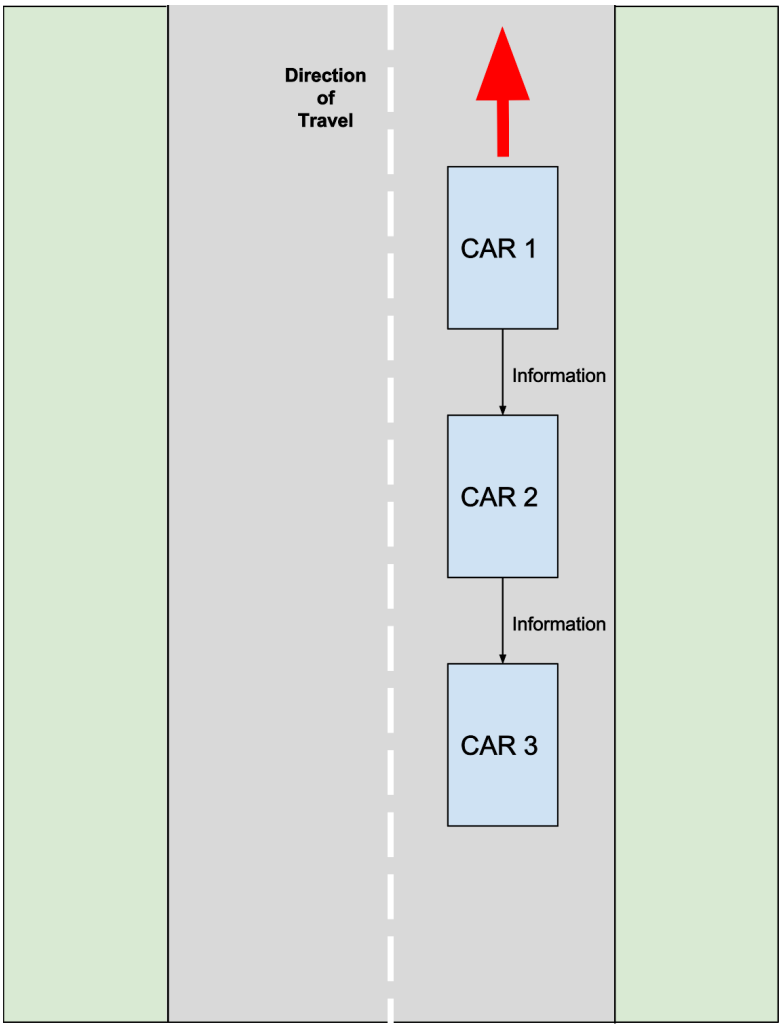

Figure 2: Overview of the platooning scenario.

In Figure 2 the cars travel in the direction of the red arrow. Information is being transmitted according to the black arrows. Car 1 is the so called platoon master, the car which leads the platoon.

2.1.2 Roundabout

A roundabout is a type of circular intersection where cars travel in a counter-clockwise direction according to the Swedish traffic regulations. When an autonomous car reaches a roundabout it needs to decide whether it will be able to enter the roundabout immediately or if it needs to wait. A schematic overview of a roundabout can be seen in Figure 3.

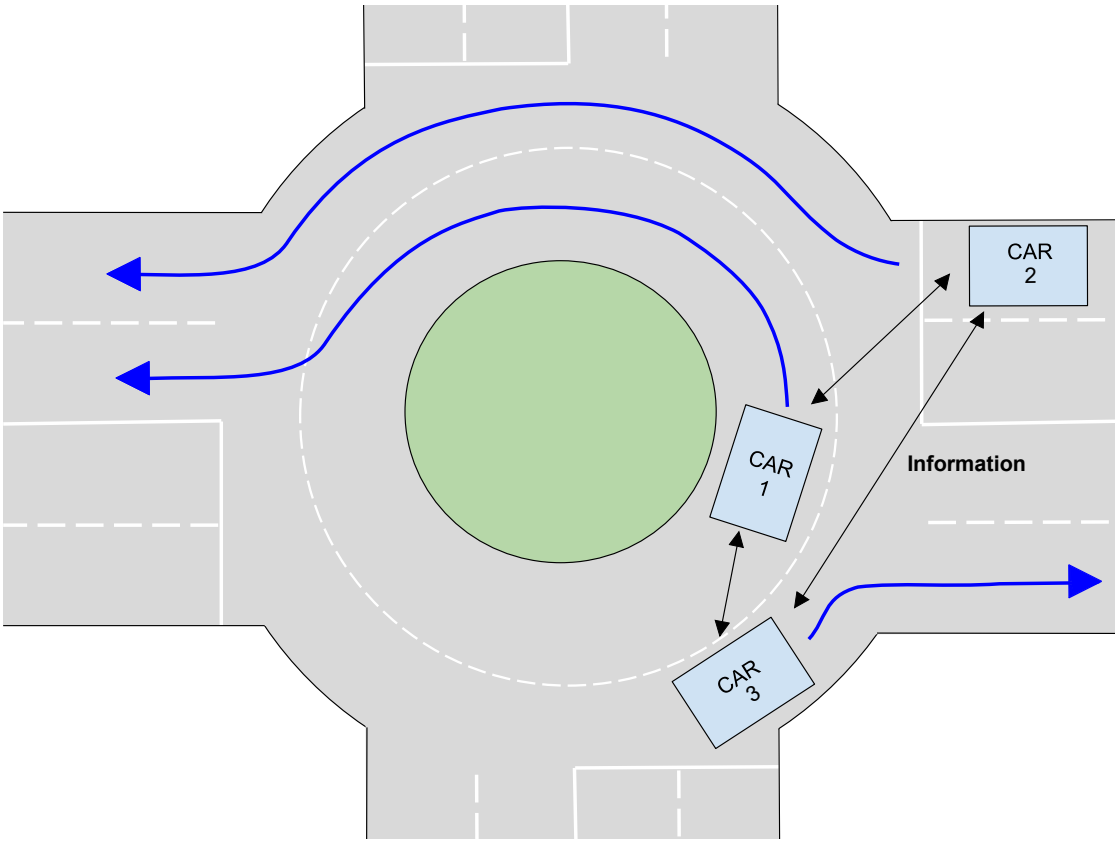

Figure 3: Overview of the roundabout scenario.

Three cars are present in Figure 3, two (car 1 and 3) in the lanes of the roundabout and one (car 2) about to enter. The blue arrows indicate which path each car intend to drive. Information is transmitted between the cars in order for the cars to complete driving the roundabout and reach their respective destination.

2.1.3 Highway with Overtaking

This scenario is basically a car traveling on a highway. It reaches another car traveling at a slower speed and it decides to overtake the other car. It must then make a decision if it will be able to overtake the other car based on a third car traveling in the other direction. A schematic overview of a highway overtaking situation can be seen in Figure 4.

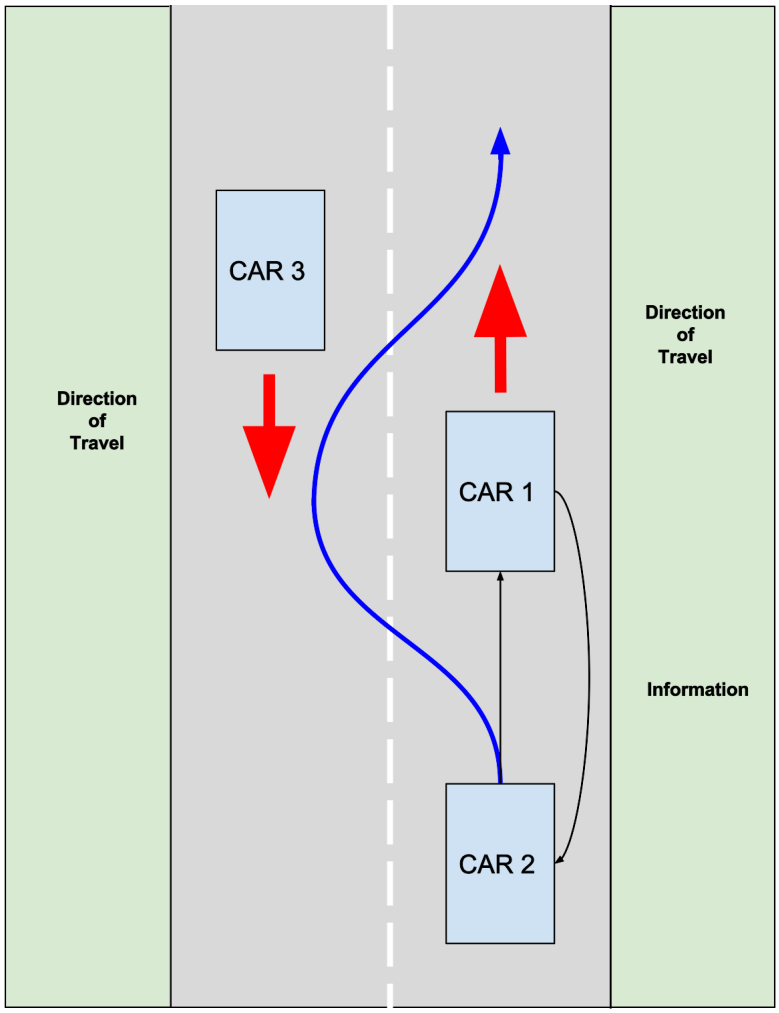

Figure 4: Overview of the car takeover scenario.

The highway takeover in Figure 4 displays three cars, of which car 2 intends to perform an overtake. The red arrows indicate the direction of travel. Information is sent among the three cars.

2.1.4 Four-way Intersection

A four-way intersection is a complex intersection often found in cities. The scenario is that the car reaches the four-way intersection knowing where it is supposed to go (left, right or straight on) and must interact with other cars currently in the intersection to avoid collision and to make the intersection and interaction as efficient as possible. A schematic overview of a four-way intersection can be seen in Figure 5.

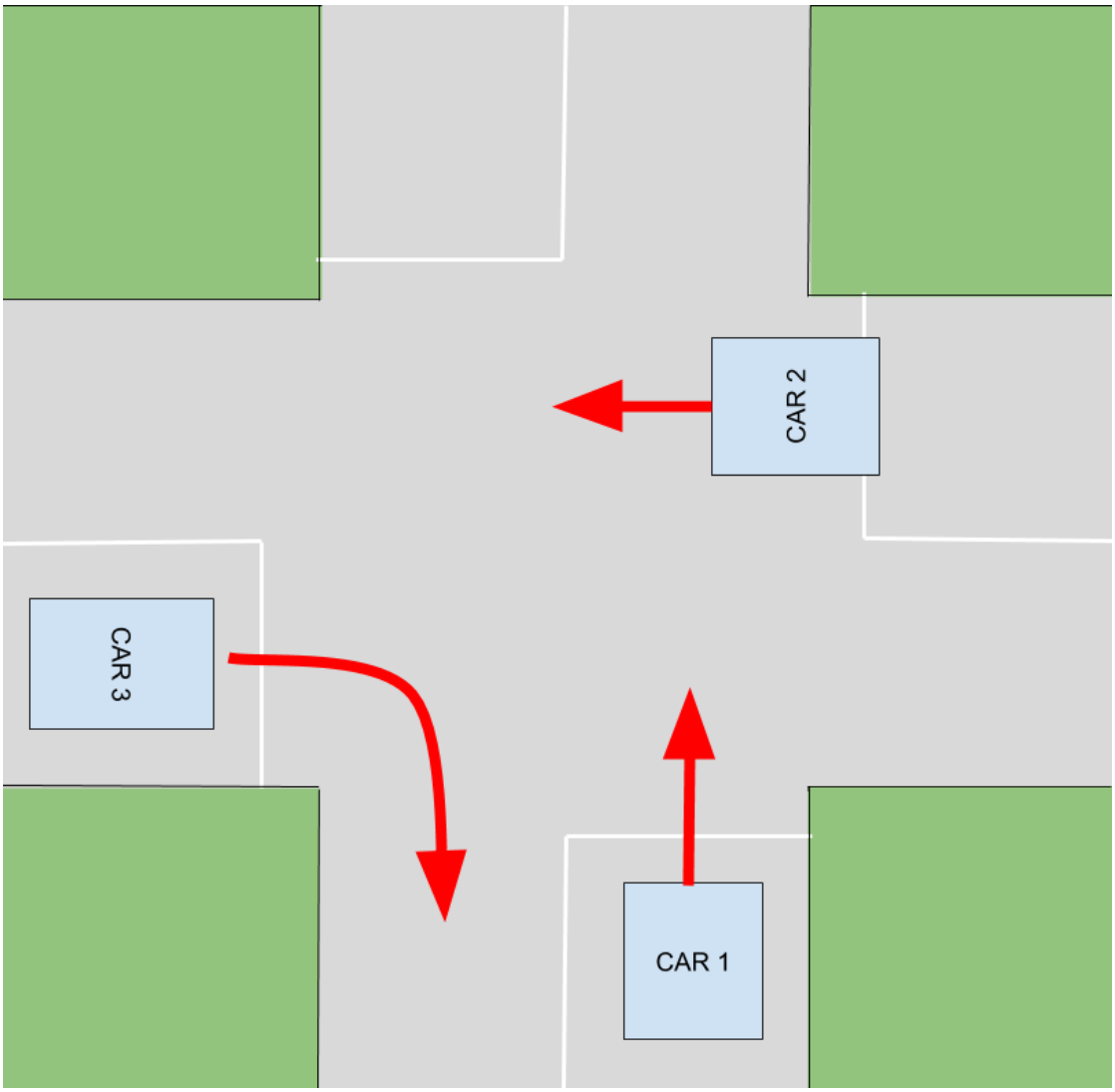

Figure 5: Overview of the four-way intersection scenario.

In the four-way intersection shown in Figure 5, three cars are driving in the intersection. Car 3 is clear to drive since no other car is crossing its route. Car 1 needs to wait on car 2 because of traffic regulations, car 1 also need to know the position of car 2 to know when it is free to drive.

2.2 Product Components

The components in the product consist of three rebuilt RC-cars with sensors, each having one RP and a microcomputer, Arduino. Furthermore access to one large arena,

Visionen, will be used. The arena provides a positioning system.

2.3 Dependencies on other Systems

The product is dependent on some other systems.

1. **VPS:** The Visionen positioning system (VPS) simulates a basic GPS system for the cars. It is a visual tracking system that uses a set of cameras to track the position of marked objects inside a confined three dimensional space.

   In the first version of the system, this is required for the functionality outlined in this document. A possible improvement, or rather generalization, is to let the cars use a type of SLAM system for positioning.

2. **ROS:** The system is dependent on the Robot Operating System to function.

2.4 Subsystems

The system will be split into a number of subsystems. Each subsystem is intended to function independently of the others and will be developed separately and integrated as they are functional. Most of the subsystems will be run on the cars but some will need to be run on an external computer, such as the user interface.

2.5 Delimitations

The project group has decided to focus on two scenarios for the cars: platoon traveling and roundabout driving. Only after these objectives have been completed, overtaking and a four-way intersection will be investigated. ROS has been chosen to be used as much as possible when designing the subsystems to complete the two driving scenarios. The idea is to have a modular setup for the cars in which various functions can be added in ROS.

2.6 Design Philosophy

As the whole idea of the project is to mimic real world scenarios with autonomous vehicles the design philosophy and goal is to deliver a product that is very robust and stable. As a autonomous vehicle malfunctioning in the real world could have catastrophic consequences.

2.7 General Requirements for the System

There are some general requirements of the system outlined here.

Req. 1	**Original**	The system shall be built using the ROS architecture.	**1**
Req. 2	**Original**	The system shall be functional inside the VPS.	**1**
Req. 3	**Original**	The system shall be modular where each component should function independently.	**1**
Req. 4	**Original**	The system shall follow Swedish traffic regulations.	**1**
Req. 5	**Original**	The cars shall be able to be controlled both manually (with a joystick) and drive autonomously in the given scenarios.	**1**
Req. 6	**Original**	The system should be able to function outside of VPS.	**3**

3 Vehicle

The complete system consists of three autonomous cars. Each car consists of a custom RC-car fitted with an RP, lidar, two odometers, an Arduino and a camera. Two of the cars are also equipped with IMUs connected to the RP. Figure 6 shows an overview of a single car and what hardware it consists of.

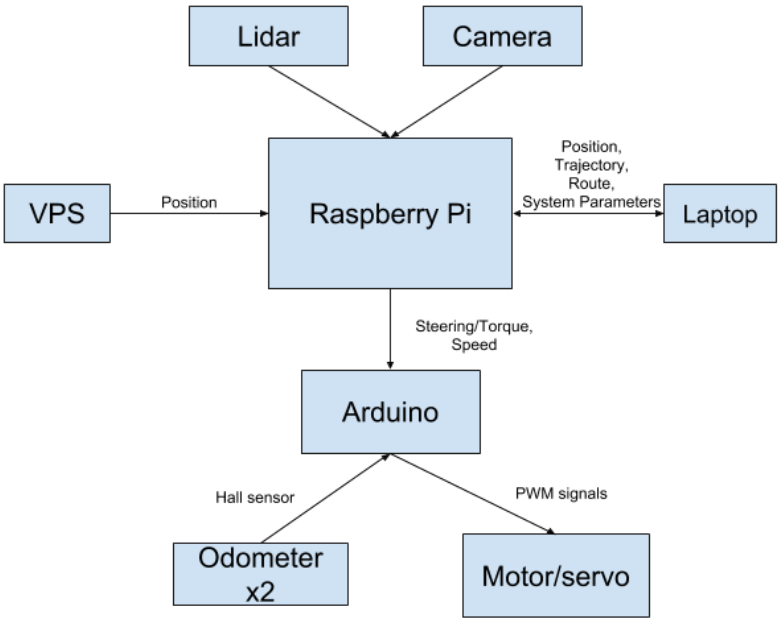

Figure 6: Overview of one separate vehicle and its communication.

Req. 7	**Original**	The system shall work with the provided hardware.	**1**

4 Route Planning

The Route Planning subsystem is responsible for the planing the general route the car should traverse. It solves an optimization problem of what the best way is to get from point A to point B. The start point, goal point, and a map containing the road map is chosen in the GUI. The coordinates is chosen in the VPS coordinate system. The subsystem as a whole has a low priority and therefore most functional requirements are priority two.

4.1 Interfaces

For the subsystem to function it has to communicate with the GUI to receive instructions. The GUI will receive start position, goal position and map as input from the user. The positions and map will then be sent to the Route Planning subsystem. Once the route has been planned the route is sent to the Behavioral Decision Making

subsystem. How the subsystem interacts with other subsystems can be visualized in Figure 7.

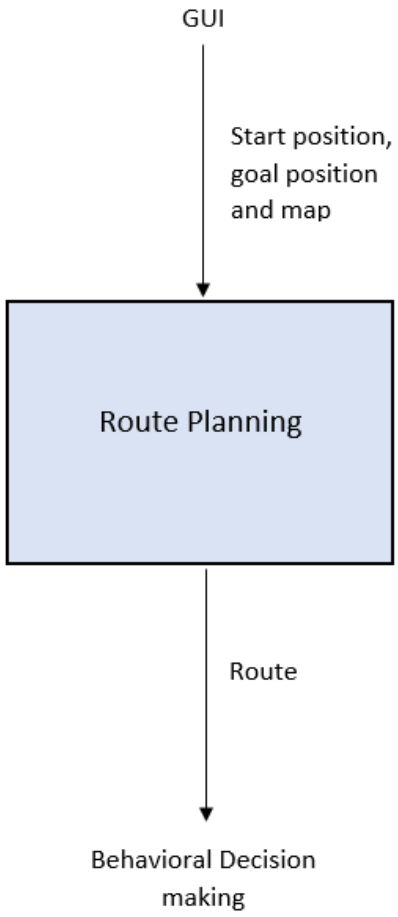

Figure 7: Interface for the Route Planning subsystem.

4.2 Design Requirements

This subsection describes the design requirements for the Route Planning subsystem.

Req. 8	**Original**	The subsystem shall be usable both in the physical car as well as the simulator.	**1**
Req. 9	**Original**	The subsystem shall be implemented in ROS.	**1**

4.3 Functional Requirements

The functional requirements for the Route Planning set by the group are presented in this subsection.

Req. 10	**Original**	The subsystem shall output a route given a set of points.	**1**
Req. 11	**Original**	The subsystem should output a route given a start end end point.	**2**
Req. 12	**Original**	The subsystem should receive the start position, goal position and map from the GUI.	**2**
Req. 13	**Original**	The start position, goal position and map should be sent to the subsystem in global positions.	**2**
Req. 14	**Original**	The subsystem should send a kinetically feasible route to the vehicle control and modeling subsystem, e.g. no impossibly sharp u-turns are allowed.	**2**
Req. 15	**Original**	The subsystem should output a route given in coordinates local to the car.	**2**
Req. 16	**Original**	The subsystem plans the route itself based on a received goal position and a map.	**2**

5 Vehicle Perception

The Vehicle Perception subsystem is in charge of handling everything the cars are able to percept. The cars can currently percept through ROS communication and their LIDAR sensors. (The VPS information is sent through ROS communication). By having a standalone subsystem at each car to handle the perceived data the whole system becomes more modular and new sensors can easily be added to the system.

5.1 Interfaces

Every car has its own ROS-node handling the perception for that specific vehicle. The data is analyzed and processed and then sent to the behavioral decision making subsystem and the motion planning subsystem. In Figure 8 the interface for the system is visualized.

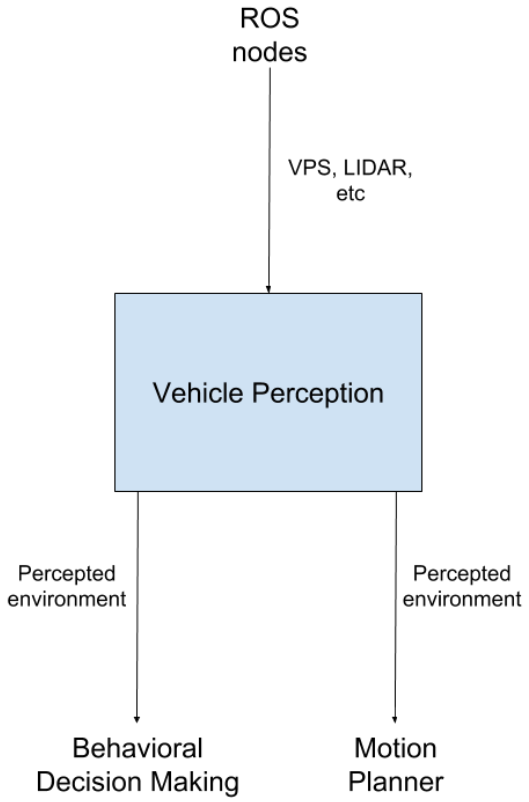

Figure 8: Interface for the Vehicle Perception subsystem.

5.2 Design Requirements

This subsection presents the design requirements for the Vehicle Perception subsystem.

Req. 17	**Original**	The subsystem shall be usable both in the physical car as well as the simulator.	**1**
Req. 18	**Original**	The subsystem shall be implemented in ROS.	**1**

5.3 Functional Requirements

The functional requirements for the Vehicle Perception subsystem are presented in this subsection.

Req. 19	**Original**	The subsystem shall receive data from VPS.	**1**
Req. 20	**Revised**	The subsystem shall receive data from LIDAR.	**2**
Req. 21	**Original**	The subsystem shall receive data from the other cars.	**1**
Req. 22	**Original**	The subsystem shall process the data received.	**1**
Req. 23	**Original**	The subsystem shall send the processed data to the rest of the subsystems on the car.	**1**

6 Behavioral Decision Making

The Behavioral Decision Making subsystem (BDM) is responsible for estimating and predicting the behavior of other traffic participants and selecting a driving behaviour based on those estimates and the complete traffic situation. For an overview of the system, see Figure 9.

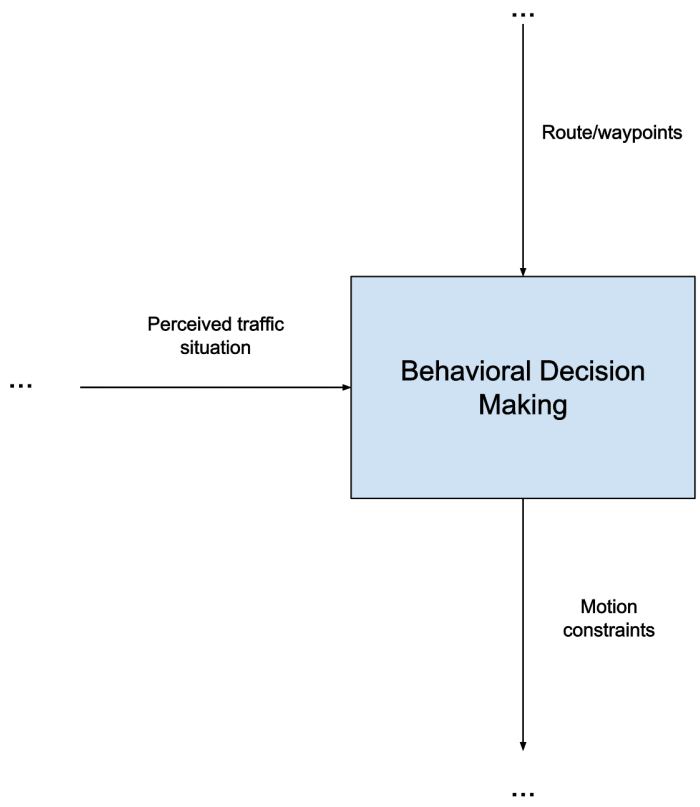

Figure 9: Behavioral Decision Making subsystem.

6.1 Interfaces

The BDM subsystem actively communicates with the motion planning subsystem described in Section 7, the route planning subsystem described in Section 4 as well as the vehicle perception subsystem described in Section 5. BDM outputs new driving behaviours if the environment changes and the current decisions would result in a dangerous traffic situation (i.e. a collision).

6.2 Design requirements

This subsection presents the design requirements for the Behavioural Decision Making subsystem.

Req. 24	**Original**	The subsystem shall be usable both in the physical car as well as the simulator.	**1**
Req. 25	**Original**	The subsystem shall be implemented in ROS.	**1**

6.3 Functional Requirements

The functional requirements for the Behavioural Decision Making subsystem are presented in this subsection.

Req. 26	**Original**	The subsystem shall not set any unrealistic constraints with respect to the vehicles kinematics for the motion planner, e.g. infinite distance to the car ahead.	**1**
Req. 27	**Original**	The subsystem shall output constraints given in coordinates local to the car.	**1**
Req. 28	**Original**	The subsystem shall be able to make autonomous decisions based on the current traffic situation given that the traffic situation is one of the two predefined scenarios.	**1**
Req. 29	**Original**	The subsystem shall be able to receive intentions from other traffic participants.	**1**
Req. 30	**Original**	The subsystem shall base its decisions on the intentions of other traffic participants.	**1**
Req. 31	**Original**	The subsystem shall output the decision to the motion planner.	**1**
Req. 32	**Original**	The subsystem should be constructed generally to use the same decision structure for the two predefined traffic situations.	**2**
Req. 33	**Original**	The subsystem should be able to predict the behavior of other traffic participants.	**2**
Req. 34	**Original**	The subsystem should output the decision to the GUI.	**2**
Req. 35	**Original**	The subsystem should be constructed generally to use the same decision structure for an arbitrary traffic situation.	**3**

7 Motion Planning

The Motion Planning subsystem will be responsible for computing a safe, feasible trajectory from the vehicle's state to a goal state. What the goal state is may vary, e.g., a point further on in the current lane, a stop line at the upcoming intersection or a point on a new road we would like to turn on to. The motion planning subsystem will receive data about static and dynamic objects in the surroundings of the vehicle and generate a collision-free trajectory based on constraints of the vehicle dynamics and the behavioral decision making layer. The output of the motion planner is sent to the vehicle control layer which will be responsible for following the trajectory. The overview of the motion planner can be seen in Figure 10.

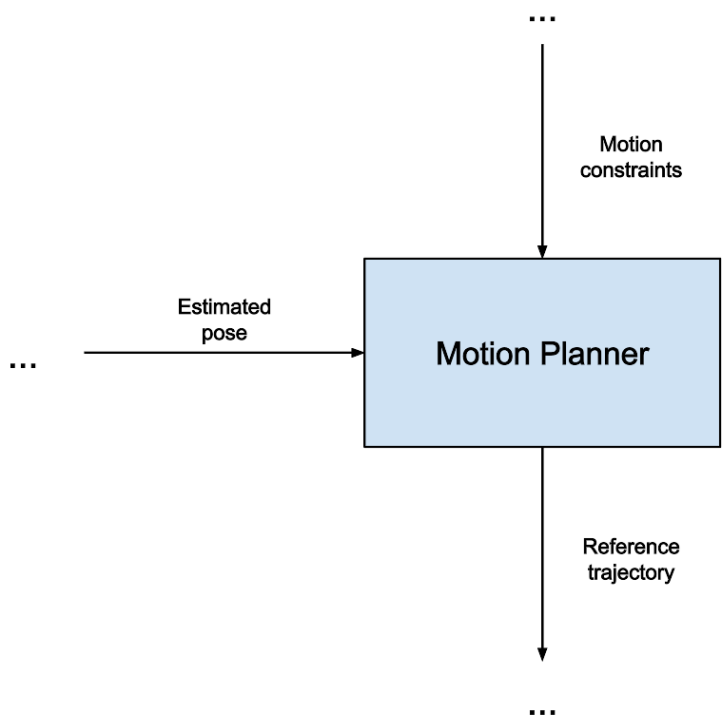

Figure 10: Motion Planner subsystem.

7.1 Interfaces

The motion planning subsystem actively communicates with two other subsystems, the BDM subsystem described in Section 6 and the vehicle control subsystem described in

Section 8. It also receives positioning information from VPS. It will communicate over Wi-Fi with the ROS messaging system.

7.2 Design Requirements

This subsection describes the design requirements on the Motion Planning subsystem.

Req. 36	**Original**	The subsystem shall be usable both in the physical car as well as the simulator.	**1**
Req. 37	**Original**	The subsystem shall be implemented in ROS.	**1**

7.3 Functional Requirements

The functional requirements for the Motion Planning subsystem are presented in this subsection.

Req. 38	**Original**	The subsystem shall always output a trajectory or path given a specific situation.	**1**
Req. 39	**Original**	The subsystem shall output the trajectory to the vehicle control subsystem.	**1**
Req. 40	**Revised**	The trajectory output shall be collision free given a known static environment.	**1**
Req. 41	**Original**	The subsystem should always output an optimized trajectory given a specific situation.	**2**
Req. 42	**Original**	The trajectory output should be collision free given a dynamic environment.	**2**
Req. 43	**Original**	The subsystem should output the trajectory to the GUI.	**2**

8 Vehicle Control and Modeling

The Vehicle Control subsystem is responsible for the motion control of the car and following a defined path. For the car to be available to drive in a road network on its own it needs to be able to follow a defined path. The performance of the path following is critical for different maneuvers but also so the car does not run in to obstacles, other cars or run off the road.

To achieve good path following, a model of the car's motion is needed. The accuracy of the motion model depends on how good the path following needs to be.

For the car to perform some of the maneuvers e.g. passing another car or driving in a convoy the speed of the car needs to be controlled. The control of the speed is important to not lose control of the car when accelerating or braking.

8.1 Interfaces

The Vehicle Control subsystem will communicate with the motion planning subsystem described in Section 7 and receive new trajectories to follow. The subsystem will also receive its position from the the vehicle perception subsystem described in Section 5.

Messages to an Arduino which controls the motors of a car and servos will be sent from the subsystem. The messages will control wheel speed and steering angle. An overview of the subsystem can be seen in Figure 11.

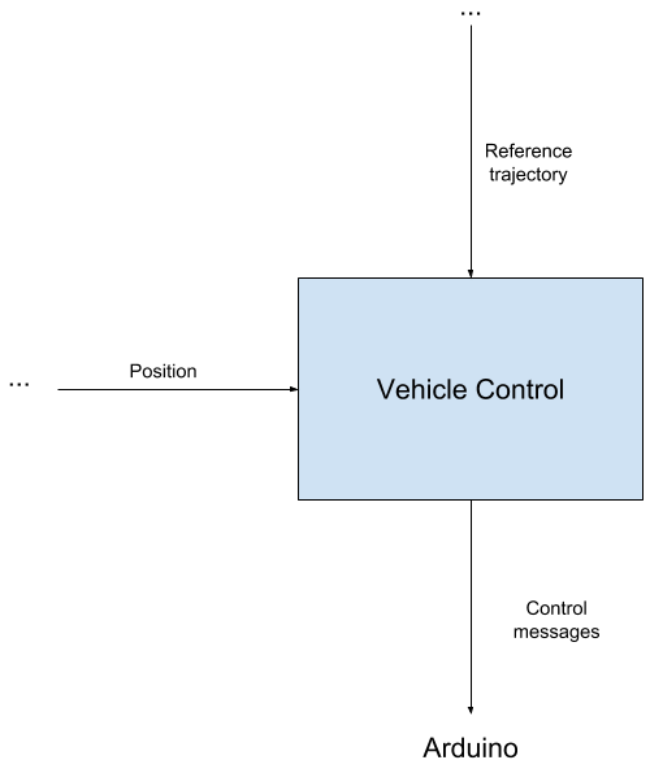

Figure 11: Vehicle Control subsystem.

8.2 Design Requirements

This subsection describes the design requirements on the Vehicle Control subsystem.

Req. 44	**Original**	The subsystem shall be usable both in the physical car as well as the simulator.	**1**
Req. 45	**Original**	The subsystem shall be implemented in ROS.	**1**

8.3 Functional Requirements

This subsection describes the functional requirements on the Vehicle Control subsystem.

Req. 46	**Revised**	The car shall be able to follow a defined path. The center of the car shall not deviate more than 10 cm in either direction of the path.	**1**
Req. 47	**Revised**	A motion model of the car with enough accuracy shall exist. The accuracy of the model shall be high enough to satisfy the requirement to not deviate more than 10 cm from the path.	**1**
Req. 48	**Revised**	The speed of the car shall be controlled in a manner that the car still follows its path and not deviate more than 10 cm in either direction of the path.	**1**
Req. 49	**Original**	The subsystem shall not make assumptions if it does not have a current trajectory. If it has not received a trajectory from the motion planner it should stand still. Also if it reaches the end of the trajectory without receiving a new one it shall stop.	**1**
Req. 50	**Original**	The subsystem should use single-wheel odometry.	**1**
Req. 51	**Original**	The subsystem should use two-wheel odometry.	**2**

9 Simulator

A simulator for the car is necessary in order to minimize the need for real world testing whilst retrieving the same quality of test feedback about the system. The obvious desire of the simulator is that it should mimic the behaviour of the car in the real world, even though all results are computed. After completion of the project, the system will be used in other courses for student education, which results in strict requirements for a simple interface in the simulator.

9.1 Simulator Interface

The simulator should preferably have a graphical interface to easily set up and start the simulation. The interface should also show the results from the simulator to the user.

In order to simulate the behaviour of the car, interactions with system functionality is necessary. The ROS environment makes implemented functionality easily accessible by sending out inputs for the different nodes and register their response. Figure 12 shows a sketch of the overview of the simulator subsystem.

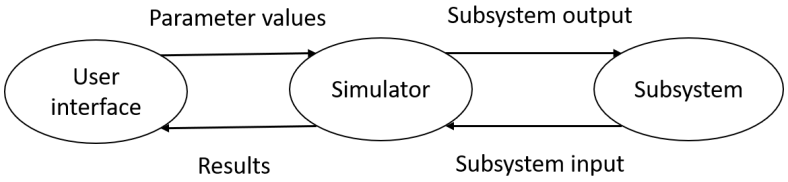

Figure 12: Simple overview of the Simulator subsystem.

9.2 Design Requirements

This subsection describes the design requirements on the Simulator subsystem.

Req. 52	**Original**	The simulator subsystem shall be implemented in ROS.	**1**
Req. 53	**Original**	The simulator shall have a graphical interface.	**1**

9.3 Functional Requirements

For the simulator, functional requirements were set according to the information in this subsection.

Req. 54	**Original**	Parameters for the simulated subsystems shall be able to be put into the simulator and applied to the subsystem.	**1**
Req. 55	**Original**	The simulator shall be able to simulate the vehicle motion and position given output from Route Planning, Behavioral Decision Making, Motion Planning and Vehicle Control subsystems.	**1**
Req. 56	**Original**	Results are shown in real-time throughout the simulation.	**2**
Req. 57	**Original**	The simulator should not take more than 10 seconds to start up.	**2**

10 Graphical User Interface

A graphical user interface (GUI) is convenient in order to interact and communicate with the car from a laptop. The GUI is supposed to give information about the system state and decisions to the user and sometimes receive user input to change or tweak the behaviour of the car.

10.1 Functional Requirements

The requirements concerning the GUI are presented in this subsection.

Req. 58	**Original**	The GUI shall be able to show the positions of all currently active cars.	**1**
Req. 59	**Original**	The GUI shall allow the user to input a route and send to a car by inputting two or more way points.	**1**
Req. 60	**Original**	The GUI shall allow the user to alter the parameters of a car.	**1**
Req. 61	**Original**	The GUI shall show the current decision that the car has taken.	**1**
Req. 62	**Original**	The GUI should be able to draw the planned trajectories of all currently active cars.	**2**
Req. 63	**Original**	The GUI should give information about the decisions of all currently active cars.	**2**
Req. 64	**Original**	The GUI should allow the user to input separate routes for all active cars.	**2**
Req. 65	**Original**	The GUI should be able to show a video feed from any car.	**2**
Req. 66	**Original**	The GUI should be able to show all three video feeds simultaneously.	**3**

11 Further Development

Since the system will be built in ROS there are many possibilities of further development when the project is finished. Potentially each subsystem could be improved to allow for better performance, reliability and safety. Furthermore the system could be expanded to handle even more complex traffic scenarios. A big area of development will probably be the behavioral decision making and motion planning subsystems. This is an ever-expanding field of research that could be exploited more thoroughly to allow for more general decision making structures and better motion planning algorithms.

12 Reliability

Since the system will be used for education it is important that it is reliable and that the described situations work as expected.

Req. 67	**Original**	The system should be able to handle all implemented scenarios in a safe and efficient manner 9 out of 10 times.	**1**
Req. 68	**Revised**	If a subsystem crashes when driving, the system should come to a complete stop so as to avoid collision.	**2**

13 Economy

The project group has access to three modified RC-cars, one laptop and the VPS. All equipment is provided by LiU.

Req. 69	**Original**	The work hours for each project member shall not exceed 240 hours.	**1**

14 Safety Requirements

The system will first and foremost be used in the Visionen arena and is thus in a confined space. Therefore the system is of minimal danger to anyone or anything but itself. There are however some requirements posed on the system so as to avoid damage to it or its surroundings.

Req. 70	**Revised**	The active cars shall try to avoid collision with known static objects.	**1**
Req. 71	**Revised**	The active autonomous cars shall try to avoid collision with each other.	**1**
Req. 72	**Original**	The active cars shall handle all implemented situations with a decent safety margin.	**1**
Req. 73	**Original**	The active cars should never collide with other visible dynamic objects.	**2**

15 Performance Requirements

In order to make sure that the project group are able to complete the project objectives, performance requirements have been set. This section displays these requirements set by the group.

Req. 74	**Original**	The active cars shall be able to complete the roundabout scenario.	**1**
Req. 75	**Original**	The active cars shall be able to complete the platooning scenario.	**1**
Req. 76	**Original**	The different scenarios shall only have one lane in each direction.	**1**
Req. 77	**Original**	The different scenarios should have two lanes in each direction.	**2**
Req. 78	**Original**	The active cars should be able to complete the four-way intersection scenario.	**2**
Req. 79	**Original**	The active cars should be able to complete the overtaking scenario.	**2**
Req. 80	**Original**	The active cars should be able to travel faster than 1m/s.	**2**

16 Requirements on Delivery

The project is set under the fall semester of 2018. Its time span is from September 3 2018 - December 17 2018. There are several deadlines throughout the course, the most important one being the project conference on December 17 2018. Prior to this day, all other subtasks need to be completed. Also, it is most important that the project is approved by the customer contact before presenting it to the customer.

17 Documentation

In Table 4 the documentation that will be written during the project is listed.

Table 4: Documentation parts of the project.

Document	Language	Purpose	Target Group	Format
Requirement Specification	English	Requirements of the project	Customer	PDF
Project plan	English	Defines How the project will be conducted.	Customer	PDF
Timetable	English	Schedule for the project	Customer	PDF
Status report	Swedish	Weekly update of the course progress	Customer	Orally and PDF
Technical documentation	English	Technical description of the system	User, Customer	PDF
User manual	English	Manual how to use the system	User	PDF
Design Specification	English	How the design ideas will be implemented	Customer	PDF
Website	English	Explain the project	Customer	Website

18 Maintainability

Since the system is intended to be used in future courses for educational purposes as well as possibly being a platform for further development, it is necessary for the system to be maintainable and as easy to use as possible. As outlined in each section describing the subsystems, each subsystem will be constructed as a ROS node and as modular as possible to allow for easy replacement and maintenance. The software written will follow the Google code standards [1] and be reviewed by the two project members responsible for the software before they are included in the main project. Additionally, it is important to be consistent in terms of file naming and uploading when using Git. A Git style guide such as **this one** [2] is recommended to use for all group members. The goal is to leave a system which is easy to get in to and expand, maintain and further develop.

Req. 81	**Original**	The code written in the project shall be written with the Google code standard [1]	**1**

References

[1] "Google Style Guides",
Internet: `https://github.com/google/styleguide` [2018-09-24]

[2] "Git style guide",
Internet: `https://github.com/agis/git-style-guide` [2018-09-24]