ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Project Report

CENG 407

Innovative System Design and Development I

201914

Autonomous Car

Büşra Nur BAHADIR
201511006
Zehra KASAP
201611678
Advisor: *Dr. Instructor Roya CHOUPANI*

Table of Contents

Abstract

Autonomous vehicles are the indispensable technology of the future, which arises from the combination of artificial intelligence and automotive. Along with many initiatives around the world, big automotive manufacturers are now carrying out serious work on autonomous vehicles. Autonomous vehicles produced by these companies have many features such as being able to go on a certain lane, self-parking, detecting objects and avoiding obstacles. However, these vehicles do not yet have the ability to change lanes in order to detect and give way to vehicles with higher priority such as ambulance and fire truck.

Our project aims to detect vehicles with higher priority in traffic by various sensors and cameras mounted on the autonomous vehicles and to recognize the emergency vehicle and clear their way, consequently to minimize the losses caused by traffic density in emergency cases.

Key words:

Autonomous Vehicle, Artificial Intelligence, Transition Superiority

Özet

Otonom araçlar, yapay zekâ ve otomotivin bir araya gelmesiyle ortaya çıkan geleceğin vazgeçilmez teknolojisidir. Dünya çapında birçok girişimin yanı sıra otomotiv devleri de artık otonom araçlar konusunda oldukça ciddi çalışmalar yürütmekteler. Bu firmaların ürettiği otonom araçlar belirli bir şeritte gidebilme, kendi kendine park edebilme, objeleri tespit etme ve engellerden kaçınma gibi pek çok özelliklere sahipler. Ancak bu araçlar henüz ambulans, itfaiye gibi geçiş üstünlüğüne sahip araçların tespiti ve onlara yol vermek adına şerit değiştirme özelliğine sahip değiller.

Projemiz trafikte geçiş üstünlüğüne sahip araçların, otonom araçlar tarafından çeşitli sensörler ve kamera ile tespit edilip geçiş önceliğinin tanınmasını ve sonuç olarak acil vakalarda trafik yoğunluğu yüzünden meydana gelen kayıpların en aza indirgenmesini amaçlamaktadır.

Anahtar Kelimeler:

Otonom Araç, Yapay Zekâ, Geçiş Üstünlüğü

1. Introduction

1.1 Problem Statement

Emergency vehicle priority is an important issue in the traffic. If we clear the way for emergency vehicles as quickly as possible, we can increase the survival rate of the patients or people on the emergency situations. Our project aiming to add vehicle priority awareness feature to autonomous cars. Of course, autonomous cars not common for today but in the future, everyone expected to use them.

1.2 Background or Related Work

There are no official researches by big companies on the same problem, but some engineers wrote blogs and articles about it. They offered wireless communication system between cars to be able to solve the problem.

1.3 Solution Statement

In order to solve this problem, we will use cameras, audio sensors. Firstly, our cars will check all lanes if one right side is available it will start driving from there to be able to balance the traffic on the all lanes with Image Data from the front camera. With audio sensor the vehicle will recognize sirens and with back camera it will check if emergency vehicle is behind of the car and not on the opposite side of the road after emergency vehicle move away the car will return its previous lane.

1.4 Motivation

In this project, we chose today's one of the most popular topic: Autonomous Vehicle. This project, which many engineers are curious about, was very suitable both for developing and keeping up with today's technology. We examined the studies on this subject. In order to keep up with this technology, we followed current developments and improved our knowledge on this field. Since we wanted to obtain more accurate results, we decided to use simulation environments rather than building the system with limited hardware.

2. Literature Search

2.1 Autonomous Cars

We can examine autonomous car’s features in five fundamental parts:

2.1.1 Localization

Localization is meaning of its own position known by the autonomous car. It’s being possible by using sensors as LIDAR and GPS. And data is being calculated with an algorithm which is called as Kalman Filter. The Vehicle’s exact position determined with the help of odometry. Odometry uses data from motion sensors to estimate change in position.

2.1.2 Perception

Perception is how cars perceive from their environment. In this part, computer vision and neural networks takes the role. Studies such as recognizing objects and location of objects are being developed by using deep learning algorithms.

2.1.3 Prediction

In this part, the autonomous car tries to predict the behaviour of the other cars like how fast it is or which direction in it and behaviours of the pedestrians in their environment to be able to decide its own action. For this purpose, Recursive neural networks (RNN) is being used.

2.1.4 Planning

For the route planning of the autonomous cars A* search (An algorithm for finding shortest path), Lattice Planning (An algorithm for basic and limited road networks) and Reinforcement Learning is being used.

2.1.5 Control

In the Control part the steering direction, speed of the car and braking are being set. The most commonly used method is called as Proportional Integral Derivative (PID) control. The steering direction changes according to output of the lane recognition system.

2.2 Simulations for Autonomous Car

Using self-driving car simulators are good for saving money and time. And providing more accuracy with being able to test many traffic scenarios in it. Nvidia Drive Sim is most popular one among the driving simulator but it’s not open source and requires hardware like Drive AGX. Carla and Apollo Simulators are open source and provides many sensors, weather conditions etc. yet they don’t have environment voice which we need to use for simulating emergency car sirens. We contacted with companies of other simulators as Cognata and

Metamoto to requesting demo versions, but demo versions were not enough for demonstrating our project. At the end we decided to use Webots which is a simulator for scientific use and robots. They recently provide libraries for autonomous cars and has environment voice.

3. Summary

3.1 Technology Used

This software will be developed with Webots Simulator and its libraries for autonomous cars in Python. Webots is a professional mobile robot simulation software package. It offers a rapid prototyping environment, that allows the user to create 3D virtual worlds with physics properties such as mass, joints, friction coefficients, etc. [1].

We chose Webots because it is cross platform, open source, supports many languages including Python and has environment sound. It's also provides tools as OpenStreeMap to create our own world and SUMO (Simulation of Urban MObility) interface to simulate traffic and sensors as LIDAR, radar, GPS etc. It does not provide Emergency car models so we will try to create them in Blender and export to it. But modelling a car is requires modelling knowledge and skills so If we cannot make it, we will add sirens with speaker to normal car models which are the simulator provides. For voice assistant we will use Google Speech to Text API.

4. Software Requirements Specification

4.1 Introduction

4.1.1 Purpose

We aimed to add emergency vehicle priority awareness feature to autonomous cars. In our project, we plan to use Artificial Intelligence, Machine Learning, Image Processing methods and test the results in simulation environment. The Autonomous Vehicle Drive Simulator that we will use need to provide us to simulate sensors such as LIDAR, GPS, radar and gives potential sensor outputs, with these outputs and by trying out possible traffic scenarios we will improve the software that we will make.

When an emergency vehicle approaches, with audio sensors the vehicle will recognize sirens and with cameras it will check if emergency vehicle is behind of the car and not on the opposite side of the road, then will switch to an available line to clear emergency vehicle's way. This feature not only emptying based on one lane rule because emergency vehicle can approach from left lane, try to make an emergency corridor or can use shoulder of the road.

This Software Requirement Specification Report intended for software developers, software architects, testers, project managers and documentation writers. Before reading this report reading our Literature Review may help to understand working principles of sensors and algorithms. This report includes overall description of the product and requirement specification of the project.

4.1.2 Scope of Project

On this project we aimed to add emergency vehicle priority awareness feature to autonomous cars. The autonomous car that we simulate will be able to detect the emergency vehicles, and their location and direction. The autonomous car will use visual information; therefore, it is required to have multiple cameras mounted on the vehicle.

Our system will include:

- Lane detecting and following
- Object recognition and auto brake
- Virtual voice assistant
- Route Planning
- Emergency vehicle priority awareness

Current autonomous cars already have the first four of these features. We will add fifth one as a new feature to autonomous cars.

To be able to success on this project our car at least needs to follow the lane, not hit objects on the road and let the emergency vehicle pass by changing lanes. Other features may change according to the possibilities of the simulator.

4.1.3 Glossary

Table 1 Glossary of SRS

Terminology	***Definition***
User	A person who interacts with the system.
SRS	The report that provides an overview of all system components.

Autonomous Driving	A control mode which a vehicle doesn't need a driver attention.
Emergency Vehicle	An emergency vehicle is any vehicle that is designated and authorized to respond to an emergency in a life-threatening situation [2].
Vehicle Priority	When a vehicle with higher priority approaches, all other traffic must stop or move to the right side to allow the vehicles pass through.
System	System software is a type of computer program that is designed to run a computer's hardware and application programs [3].

4.1.4 References

- IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

4.1.5 Overview of Document

Rest of the documents includes development methodology we used, our project work plan, describing project’s functionalities, constraints and requirements.

4.2 Overall Description

4.2.1 Product Perspective

Self-Driving: Classification systems that monitor traffic signs, use cameras, monitor other systems, use radar and laser sensors.

LIDAR: Optical remote sensing technology to measure distance to target by illuminating with light.

GPS: Space-based satellite navigation system that provides time and location information anywhere.

Digital Maps: The process in which data collection is compiled and formatted in a virtual image.

Adaptive Cruise Control: Tracks distances to adjacent vehicles on the same lane. Detects objects in front of a vehicle at risk of emergency collision.

Lane Assist: Tracks the position of the vehicle in the lane.

Sound and Light Sensors: It is a system that will be used specially to identify cars with priority. It will also be used to detect surrounding objects, lanes, and other tools.

4.2.2 Development Methodology

In our project used Scrum which is an agile software development methodology. In the work plan everything is clearly described from start and finish dates to owner of the work and given workday for the job. After every documentation finished, we had sprints. At the total we made 4 sprints. If there are any tasks which are not completed on the time, 2 extra days, which are weekends, have been reserved for the purpose of completing these tasks.

	A	B	C	D	E	F	G	H	I	J
1	Task	Dead line for School	Start Date		End Date		Job Description	Owner	DAY LEFT	Workdays given for the job
2			14.10.2019	Mo	18.10.2019	Fr	Research of 2016-2017 papers	Zehra	Over	4
3	Literature Review Document	1.11.2019	14.10.2019	Mo	18.10.2019	Fr	Research of 2018-2019 papers	Büşra	Over	4
4			18.10.2019	Fr	20.10.2019	Sa	Finishing raport of the research	Team	Over	2
5	Sprint 1		20.10.2019	Su	20.10.2019	Su			Over	
6			21.10.2019	Mo	05.11.2019	Tu	Remaining sections on Introduction	Büşra	Over	15
7	SRS	22.11.2019	21.10.2019	Mo	08.11.2019	Fr	Overall Description	Zehra	Over	18
8			21.10.2019	Mo	5.11.2019	Tu	Requirements Specification	Büşra	Over	15
9	Project Webpage	29.11.2019	6.11.2019	We	8.11.2019	Fr	Project Webpage	Büşra	Over	2
10	Sprint 2		9.11.2019	Sa	9.11.2019	Sa			Over	
11			11.11.2019	Mo	19.11.2019	Tu	Introduction Review	Zehra	Over	8
12	SDD	13.12.2019	20.11.2019	We	6.12.2019	Fr	Architecture Design	Büşra	Over	16
13			20.11.2019	We	6.12.2019	Fr	Usecase Realization+Environment	Büşra	Over	16
14	Sprint 3		7.12.2019	Sa	7.12.2019	Sa			Over	
15	Project Report / Project Tracking Form	27.12.2019	7.12.2019	Sa	8.12.2019	Su	Project Report / Project Tracking Form	Team	0	1
16	Powerpoint Presentation	14.01.2020	9.12.2019	Mo	13.12.2019	Fri	Presentation	Team	5	4
17	Sprint 4		14.12.2019	Sa	14.12.2019	Sa			6	

Figure 1 Work Plan of our project.

4.2.3 Product Functions

4.2.3.1 Traffic Lane Detection

The system detects highway lane lines. Distinguishes dashed lines and straight lines. Provides warning in case of loss of lanes. Image analysis techniques are used to define lines.

4.2.3.2 Tracking Environment

Tracks objects around the vehicle using the scanner and algorithms. Sensors monitor the position of objects as they move within the scanning range. So, the system behaves according to objects.

4.2.3.3 Detection of Traffic Signs

Using image processing techniques and various algorithms, studies are done to classify traffic signs. Recognize traffic signs. The vehicle behaves according to the colours of the traffic lights.

4.2.3.4 Vehicle Detection and Tracking

The system performs vehicle detection and tracking events. It adjusts the speed and position according to the behaviour of the vehicles around it during highway driving. Sudden braking performs events such as lane change. Keeps track distance always constant with a vehicle in front.

4.2.3.5 Apply Braking

When a pedestrian step in front of the vehicle and a collision is possible, sudden braking has been performed. When the obstacles, pedestrians, and vehicles on the road are lifted, the vehicle accelerates to regain its speed.

4.2.3.6 Road Planning

This project aims to is to create a road planner that can create safe trajectories for the vehicle to follow. On the highway track, there are other vehicles, all at different speeds. The car transmits its position along with sensor fusion data that estimates the position of all vehicles on the same side of the road.

4.2.3.7 Priority of vehicles

Recognizes vehicles such as fire trucks or ambulances using sound sensors and image processing techniques. The system changes lanes to give way to these vehicles.

4.2.4 User Characteristics

The vehicles that will be equipped in the system used in our project will be autonomous. The user interaction with the system is expected to be minimal. We assume that the user has little or no knowledge of the system. User interact the system by giving order to voice assistant.

4.2.5 Constraints

This section covers the constraints for the system, which also includes descriptions of safety-critical properties for the system.

The system is designed to help the user operate the vehicle, and the system should always be operating correctly. There are some situations where the system requires restraint. In the absence of lane markings (an unmarked lane, dirt road, bad weather, etc.), the system becomes ineffective and must give the user an audible or visual warning. This will continue until road conditions with clearly defined lane markings are met.

If there is a problem with the cameras used to detect these strips, the system will not work again. This could also be a malfunction of cameras and sound sensors that see priority vehicles. In this case, the system switches itself off again and should display a warning to the user again. This problem is valid until the camera or sound sensors are repaired. Sensors will still be used to obtain distance information. The sensor data will be used with the camera for objects around the vehicle. It requires stable and fast internet connection when locating objects around. Otherwise it cannot communicate quickly with other tools.

The system must be configured to respond to commands in 500 msec. If there is an obstacle on the road during driving, the response time and brake release time must always be the same to achieve the desired deceleration. In all scenarios, the vehicle speed must be constant.

4.2.6 Assumptions and Dependencies

Various assumptions and dependencies went into the creation of project in order to ensure that the system works safely and efficiently.

- The main assumption in this project is that the system works well when there are no environmental factors. (Bad weather, holes, slope, etc.)
- Lane markings are assumed to be distinct.
- In our system, it is assumed that all traffic signs and the presence of all objects around the vehicle can be clearly and seen.
- Priority vehicles such as ambulances and fire trucks are clearly recognized by their voice and image recognition.
- If the vehicle is running, it is assumed that the system is always on and scanned.
- It is assumed that all system elements are operating properly and there are no abnormal conditions.

4.3 Requirements Specification

4.3.1 External Interface Requirements

4.3.1.1 User interfaces

User can start and turn-off the system. And may can see informations about car from dashboard like the speedometer, tachometer, odometer, engine coolant temperature gauge, and fuel gauge, turn indicators, gearshift position indicator, seat belt warning light, parking-brake warning light, and engine-malfunction lights. And can give order like “speed up to 70 km/h”, “change the left lane” with virtual assistance.

4.3.1.2 Hardware interfaces

Hardware interface may include sensors and necessarily control unit (e.g. DRIVE AGX Kit) but for now we're trying to make and test everything on simulation environment so, there won't be any external hardware used except our computers.

4.3.1.3 Software interfaces

Since its on simulation environment there are no external software interface requirements. But if we can build a prototype, Software Interfaces will change with respect to the hardware which we will be using.

4.3.1.4 Communications interfaces

There are no external communications interface requirements.

4.3.2 Functional Requirements

4.3.2.1 Lane Detecting and Following

System needs to detect the lines and follow the selected lane until user wants to change it or emergency vehicle approaches on the lane.

4.3.2.2 Object Recognition and Auto Brake

System need to recognize objects as human, vehicle, traffic light and trigger auto brake in case of appearance of an object in front of the car.

4.3.2.3 Virtual Voice Assistant

User can give basic order to system by using virtual voice assistant. (ex: turn left, turn light, change lane etc.)

4.3.2.4 Route Planning

User can give a destination location to system by using virtual voice assistant. And system should plan a route to destination.

4.3.2.5 Emergency vehicle priority awareness

System need to detect the emergency vehicle, and their location and direction and change its lane if it is on the same lane with the vehicle.

4.3.3 Performance Requirements

Performance Requirement	Description
Response Time	This system work in real-time. So, response time (T_response=T_actuation-T_event) must be 500 msecs.
Error Handling	When an unpredictable failure occurs, system need to recover briefly.
Workload	System should be able to handle many inputs from its environment in different challenging weather and traffic conditions.
Scalability	Sensors and other used hardware tools effective on it but we're working on simulation there is no scalability requirement.

4.3.4 Design constraints

The hardware, software and technology used should have following specifications:

- Processor with speed of 2GHz
- Continuous power supply
- Ability to use camera, microphone and other services of the system
- 1GB memory or more

The software being developed will be tested on Webots which an open source and cross platform simulator is.

4.3.5 Software system attributes

Quality Attribute	***Description***
Correctness	The system needs to recognize objects, follow the traffic rules and avoid accidents.
Reliability	Every functionality on the code must be able to work smoothly without failure

Quality Attribute	***Description***
	under given normal conditions.
Learnability	The system needs to be simple enough to learn by users.
Robustness	The autonomous car must work properly under given abnormal traffic or weather conditions.
Maintainability	When an unpredictable failure occurs, system needs to correct defects their cause, repair it and report the failure.
Extensibility	Ability to extend the system is limitless. New functionalities can be added to system anytime.
Testability	Every functionality on the code must be able to work smoothly in simulation environment.
Efficiency	The system should work with maximum performance with minimum used energy and fuel.
Portability	The system should work on Linux and Windows.

5. Software Design Description

5.1 Introduction

This Software Design Description Report provides detailed information about the requirements of the AutoCar system software. This document includes the working principles of the proposed methods, and designs of the simulation.

5.1.1 Purpose

Our main purpose is "Increasing the survival rate of the patients or people on the emergency situations by clearing the emergency vehicle's way with adding emergency vehicle priority awareness feature to autonomous cars". We may cannot apply this project to our lives since generally people do not use autonomous cars today, but it may help to humans in the future when all cars on the roads become autonomous.

5.1.2 Glossary

Table 2 Glossary of SDD

Terminology	***Definition***
User	A person who interacts with the system.
SDD	The report that provides an overview of system's design.
Autonomous Driving	A control mode which a vehicle doesn't need a driver attention.
Emergency Vehicle	An emergency vehicle is any vehicle that is designated and authorized to respond to an emergency in a life-threatening situation [4].
Vehicle Priority	When a vehicle with higher priority approaches, all other traffic must stop or move to the right side to allow the vehicles pass through.
System	System software is a type of computer program that is designed to run a computer's hardware and application programs [5].
Control Unit	The control unit (CU) is a component of a computer's central processing unit (CPU) that directs the operation of the processor [6].
ADAS	Advanced driver-assistance systems.
LIDAR	Light Detection and Ranging, is a remote sensing method that uses light in the form of a pulsed laser to measure ranges (variable distances) to the Earth.[7]
GPS	The Global Positioning System (GPS), originally NAVSTAR GPS, is a satellite-based radio navigation system owned by the United States government. [8]

5.1.3 Overview of document

The remaining sections and their contents are listed below. The Architectural Design that defines the stage of project development. The subheading of this topic includes the identification and explanation of the problem and the technologies used. The architect details the general diagrams of the simulator program as a subtitle of the Design section. Sequence Diagram, Activity Diagram and Class Diagram. The continuation of section includes the elaboration and detailed explanation of the project plan. In Environment section, we have detailed and demonstrated the tools in the simulation environment.

5.2 Deployment

We will deploy our code with docker.

5.3 Architecture Design

5.3.1 Sequence Diagram

Figure shows sequence diagram of our project. The user gives order to voice assistant to choose the route and start it. After it starts Main controller handles everything by obtaining data from sensors and sends them to each sub modules with multithreading, takes necessarily actions in a loop.

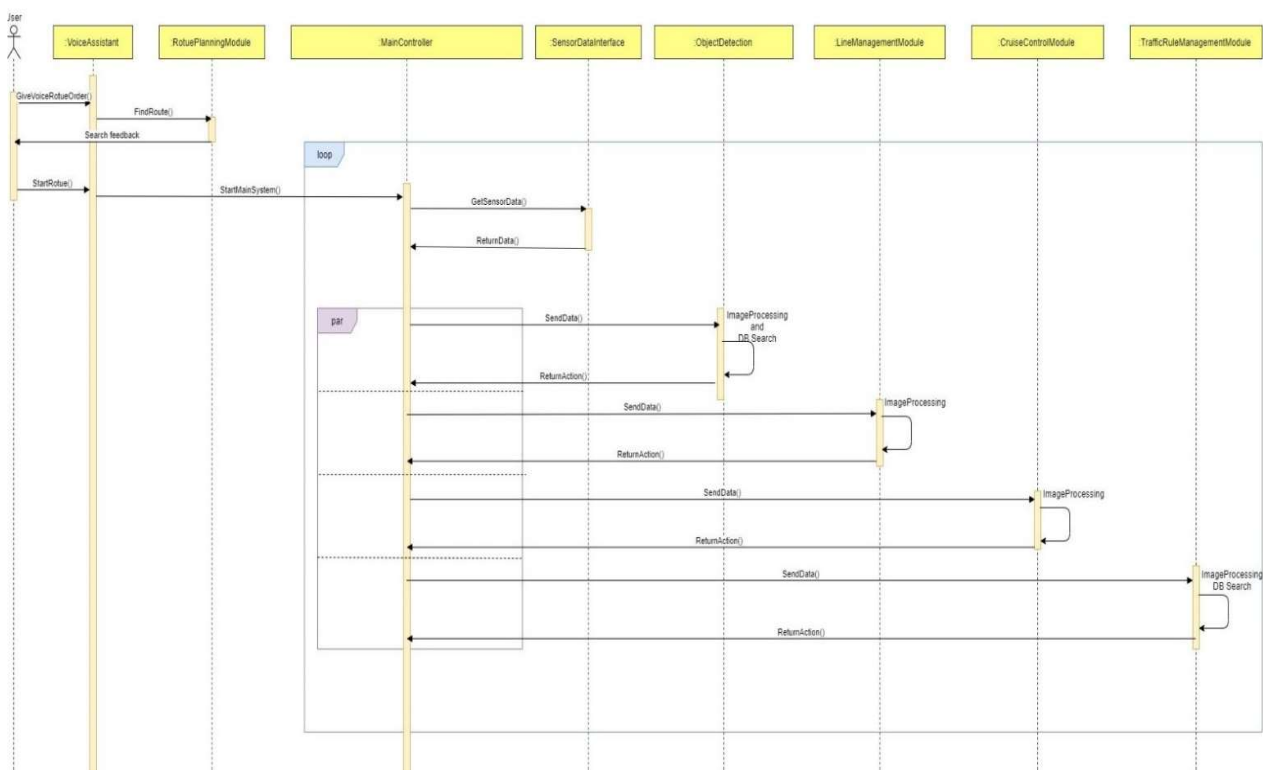

Figure 2 Sequence Diagram of our project.

5.3.2 Activity Diagram

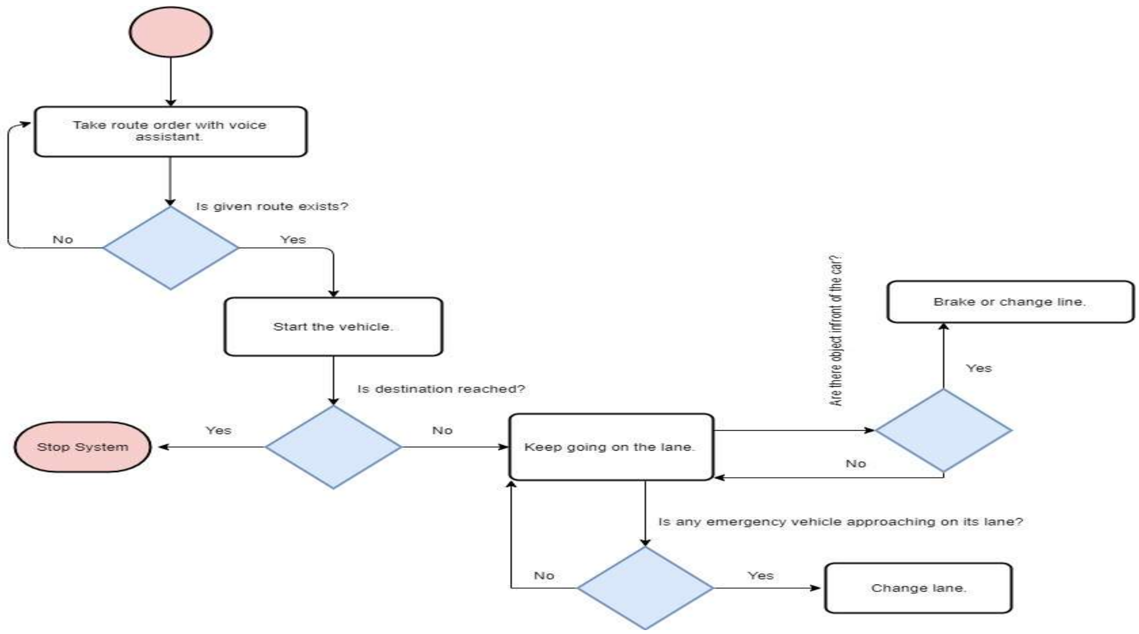

Figure 3 Activate Diagram of our project.

Figure shows activity flows in our program. User gives order to system by voice assistant after system starts to run it tries to reach destination by itself.

5.3.3 Class Diagram

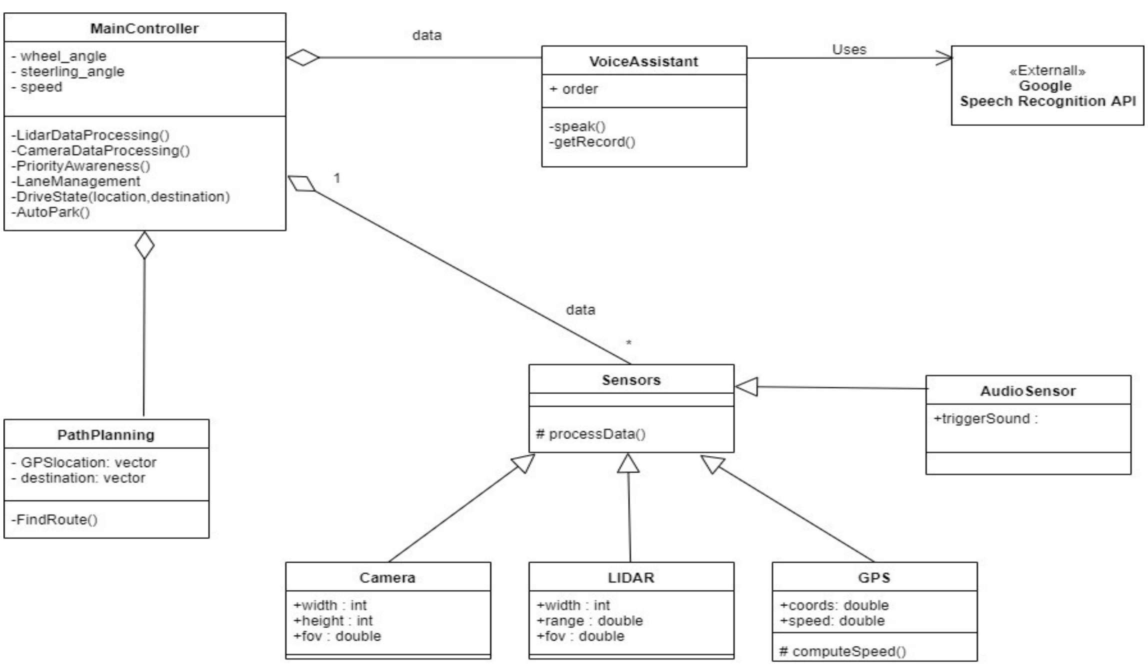

Figure 4 Class Diagram of our project.

5.3.4 Architecture Design of AutoCare

5.3.4.1 Create Route

Summary: The user can select his/her location, determine the target position. With giving order by voice assistant.

Actor: User.

Precondition: The car must be connected to the Internet and keep the location settings open.

Basic Sequence:

1. The user calls voice assistant to start it.
2. The user tells desired destination location.
3. The user can select when to start travel.
4. The user can cancel the route.

Exception:

- Location error due to cellular data or another external factor.
- Incorrect orientation if map information is out of date.

Post Conditions: User must tell destination location information.

Priority: Medium.

5.3.4.2 Driving State

Summary: The car will try to reach the destination.

Actor: None.

Precondition: The car must be connected to the Internet and keep the location settings open.

Basic Sequence:

1. Start engine.
2. Follow the path to the destination.
3. Avoid collisions.
4. Consider traffic rules.
5. Clear the way if emergency vehicle approaches.

Exception: None.

Post Conditions: Sensors need to work well.

Priority: High.

5.3.4.3 Change Direction

Summary: The user can change the path with giving order by voice assistant.

Actor: User.

Precondition: The car must be connected to the Internet and keep the location settings open.
Basic Sequence:

1. The user calls voice assistant to start it.
2. The user tells desired direction or lane.
3. The route will recalculate by system.
4. The car follows the new path.

Exception: If there is not such a lane (ex: one right lane when there is not exists.) or direction (ex: when user orders to turn left but there is no way.).
Post Conditions: Sensors need to work well.
Priority: Medium.

5.4 Data structure design

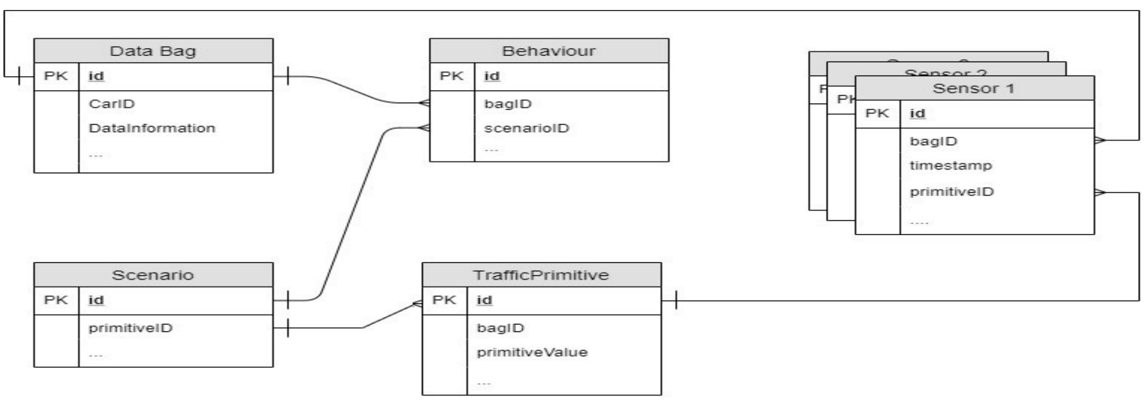

Figure 5 Database Diagram of our project.

In our Database we have ‘DataBag’, ‘Scenario’, ‘TrafficPrimitive’ and ‘Behavior’ tables to relate all the driving data. We don’t need prior knowledge of driving scenarios and primitives, but we need manually defined several hypo-theorical scenario conditions.

5.5 Use case realizations

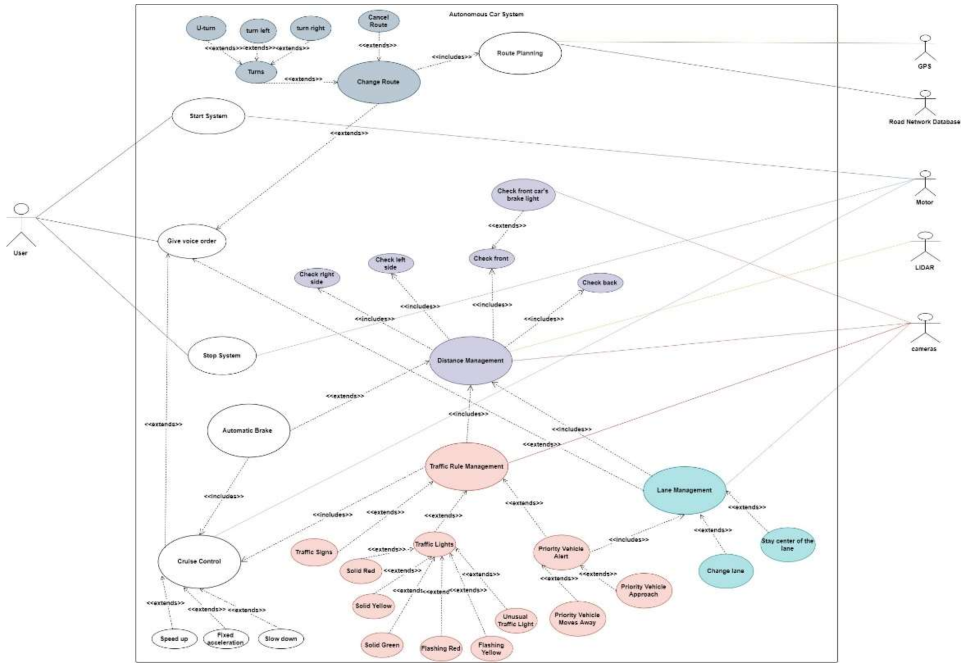

Figure 6 Use Case Diagram of our project.

5.5.1 Use Case: Start System

Description: User turns on the system to use the vehicle.

Functional Response: The systems will turn on the sensors and checks other components. Waits for destination input.

5.5.2 Use Case: Give Voice Order

Description: User can give orders by voice to change the destination or acceleration.

Functional Response: With the system's virtual assistant system will get the order and based on it Cruise Control Module or Lane Management Module or Change Route Modules will be activated.

5.5.3 Use Case: Route Planning

Description: System takes destination input from user.

Functional Response: System gets destination input. If can't find destination on GPS or road network database displays error message. If else, it calculates shortest path to destination and shows route.

5.5.4 Use Case: Change Route

Description: User can change route with voice order or can withdraw it.

Functional Response: Voice order will be recognized by system's virtual assistant. Based on the user's order new shortest path will be calculated or the vehicle will be parked on highway shoulder or parking areas until new order comes.

5.5.5 Use Case: Distance Management

Description: Checks vehicle's environment and finds estimate distance to objects with the data comes from cameras and LIDAR.

Functional Response: The computer vision data is sent to the machine learning model for classification. The model classifies the selected area of the environment is clear or not. And tries to balance distance between objects and the vehicle.

5.5.6 Use Case: Automatic Brake

Description: Stops the car when an object appears in front of the car.

Functional Response: When an object suddenly appears in front of the car, system automatically stops the car. The car remains stationary so long as the obstacle is present. Once cleared, it continues its destination.

5.5.7 Use Case: Lane Management

Description: Manages the lane operations.

Functional Response: In the lane it's balances the vehicle to stay in the center of the lane with the computer vision data comes from cameras. If there is a request from user to change the line uses Distance Management to check environment and changes the line based of the request.

5.5.8 Use Case: Cruise Control

Description: It controls the speed of the motor.

Functional Response: It changes the speed of the motor based on user's order or responds comes from Distance Management module.

5.5.9 Use Case: Priority Vehicle Alert

Description: Displays alert on the dash board and takes necessarily actions when an Emergency Vehicle approaches or moves away.

Functional Response: When an emergency vehicle approaches, with audio sensors the vehicle will recognize sirens and with cameras and sensors it will check if emergency vehicle is behind of the car and not on the opposite side of the road, then will switch to an available line to clear emergency vehicle's way. This module not only emptying based on one lane rule because emergency vehicle can approach from left lane, try to make an emergency corridor or can use shoulder of the road.

5.5.10 Use Case: Solid Red

Description: When there is a traffic light in front of the car with a solid red light. It means stop the car.

Functional Response: The car will slow down and stop until next signal input from computer vision.

5.5.11 Use Case: Solid Yellow

Description: When there is a traffic light in front of the car with a solid yellow light. It means get ready.

Functional Response: The car will slow down until see the next input. If it's red Solid Red Use case will be on, if it's green Solid Green Use case will be on.

5.5.12 Use Case: Solid Green

Description: When there is a traffic light in front of the car with a solid yellow light. It means go.

Functional Response: The car accelerates straight if there is no obstacle on the road.

5.5.13 Use Case: Flashing Red

Description: When there is a traffic light in front of the car with a flashing red light. It means the same as a stop sign. After stopping, proceed when safe and observe the right-of-way rules.

Functional Response: The car will slow down and stop until next signal input from computer vision

5.5.14 Use Case: Flashing Yellow

Description: When there is a traffic light in front of the car with a flashing yellow light. It warns to be careful. Slow down and be especially alert.

Functional Response: The vehicle will slow down and after checking environment with Distance Management. The car will continue its destination.

5.5.15 Use Case: Unusual Traffic Light

Description: When there is a traffic light in front of the car with no lights on or more than one lights on.

Functional Response: The vehicle will slow down and after checking environment with Distance Management. The car will continue its destination.

5.5.16 Use Case: Traffic Signs

Description: When there is a Traffic Sign in front of the car.

Functional Response: The computer vision data is sent to the machine learning model for classification. The model classifies the sign and acts based on it.

5.5.17 Use Case: Stop System

Description: Turns off the system.

Functional Response: When user wants to turn off the system, vehicle parks automatically in a safe area and stops the system.

6. Conclusions

In this semester we did researches about Self Driving Car algorithms and systems for 'Literature Review' document. We find out that Autonomous cars don’t have vehicle priority awareness feature and decided to add this feature as an improvement on this field in our project. We decided our software needs, did researches and prepared our 'SRS' document with the information we obtained. Once the requirements are identified, the design of the developing product is prepared and described in an SDD document. During this period, we did research and decided about the simulator environment to be used. We created the product architecture and talked about the deficiencies in our project.

The simulated autonomous car that we will code, will be able to detect emergency vehicles, their location and direction with the sensors and cameras which are mounted on it. When the autonomous car recognises the emergency vehicles, it will change the lane to clear the emergency vehicle's way. Our autonomous vehicle will have features such as lane detection and tracking, object recognition and automatic braking, voice assistant, route planning and emergency vehicle priority awareness. The current autonomous cars already have the first four of these features. We will add fifth feature as an innovation to autonomous cars.

To be able to successful in this project, our car must at least follow its lane, not hit objects on the road and allow the emergency vehicle to change lane. Other features may change depending on the possibilities of the simulator.

7. Acknowledgement

We are grateful for guidance we have received from our advisor Dr Instructor Roya CHOUPANI.

8. References

[1] https://cyberbotics.com/doc/guide/introduction-to-webots

[2],[4] https://en.wikipedia.org/wiki/Emergency_vehicle

[3],[5] https://whatis.techtarget.com/definition/system-software

[6] https://en.wikipedia.org/wiki/Control_unit

[7] https://oceanservice.noaa.gov/facts/lidar.html

[8] https://en.wikipedia.org/wiki/Global_Positioning_System