Summative Assessment 1: System Design

Object Oriented Programming

Word count: 963

A Design Proposal of Software to Support Operation of a Driverless Car

The concept of a driverless car (DC) refers to a vehicle, capable of navigating on an open road from point A to point B and requiring minimal input from a human driver or functioning entirely without one (Leiss, 2023; Bathla et al., 2022). DCs gather information from a variety of environmental recognition sensors and cameras such as laser, ultrasound, radar, and LIDAR. These passive systems as shown in Figure 1 allow for distance measurement and the processing of other information related to the vehicle's surroundings, as well as communication with infrastructure and other vehicles (Leiss, 2023). The collected data is processed by computer and used to control the engine acceleration, brakes, and steering, or to intervene in these systems as well as to locate paths, obstacles, and pertinent signage. This mimics the complex behaviour that human operators perform when they scan the path and car while driving. Some recent advancements in this field include the patent of traffic light interpretation by Google and Tesla's latest autonomous vehicle, which modifies its speed according to street signs. (Parekh et al., 2022).

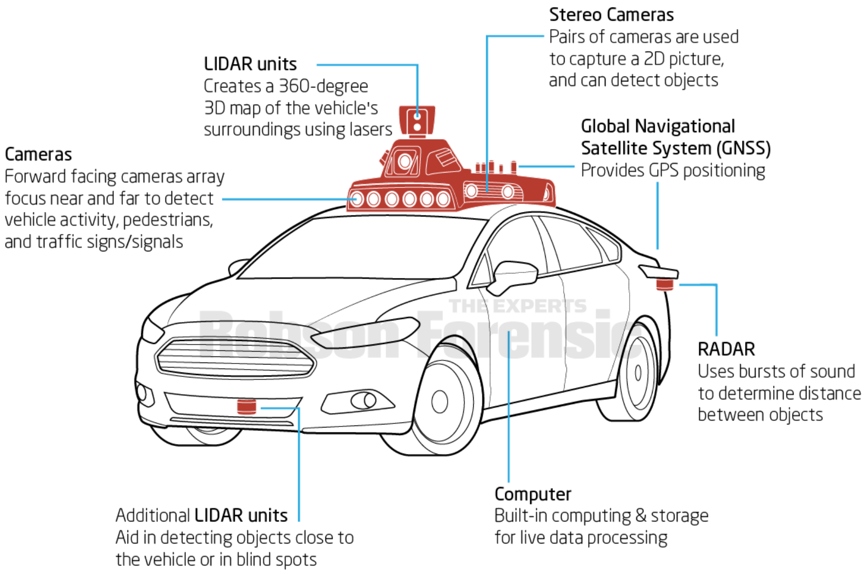

Figure 1. Basic components of driverless car (Leiss, 2023).

DCs may use various combinations of technologies to perform their autonomous functions. However, the functions outlined in Table 1 collaborate to enable a DC to operate securely and effectively in a range of driving circumstances. These functions, also known as the main functions of DC:

Table 1. Main functions of driverless car.

1. Environmental Perception	Environmental perception refers to developing a contextual understanding of the environment, such as where obstacles are located, detection of road signs/marking, and categorizing data by their semantic meaning (Pendleton et al., 2017). Primarily three types of sensors are in use today for data acquisition: 1. **LIDAR** – Light Detection and Ranging, is a remote sensing technology that leverages laser light to measure distances and construct maps of the environment. 2. **RADAR** - like LIDAR, is a technology that uses radio waves to detect the presence and location of objects 3. **Cameras** - capture images or video of the environment which can be used to detect and identify objects, such as pedestrians, other vehicles, traffic lights, and road signs.
2. Localization	The car determines its precise location and orientation on a map, often using GPS, inertial measurement units (IMUs), and other localization techniques. This module aims to localize the vehicle concerning its environment - know its position accurately enough to navigate to its target. It processes data from several of the sensors above and sometimes performs Simultaneous Localization and Mapping. GPS is often used to get a rough global estimate of the vehicle's position, and Kalman filters, particle filters etc. are used to fuse the information from the other sensors and get accurate localization (Thakurdesai & Aghav, 2020).
3. Path Planning	Based on the ***Environmental Perception*** and ***Localization*** data, the car plans a safe and efficient route to its destination, considering factors such as traffic conditions, speed limits, and road regulations. Path planning is a complex problem that deals with the physical constraints of DCs, other constraints from the environment, operational requirements, and above all, finding smooth paths (L'Afflitto et al., 2024).
4. Control and Navigation	The car's control system executes the planned path by controlling the steering, acceleration, and braking, while also navigating through traffic and following the planned route (Sjafrie, 2020).
5. Decision Making	The car's onboard computer system makes real-time decisions to handle complex driving scenarios, such as merging into traffic, changing lanes, and responding to unexpected events (L'Afflitto et al., 2024).
6. Communication	DCs may communicate with each other and with infrastructure (V2V and V2X communication) to share information about road conditions, traffic, and potential hazards (Bathla et al., 2022).
7. Passenger Interaction	Some DCs may provide interfaces for passengers to interact with the car's systems, monitor the route, and control certain functions (L'Afflitto et al., 2024).

UML Models

The UML diagram presented herein delineates the architecture and interactions of a driverless car system, encompassing crucial components responsible for autonomous vehicle operation and passenger interaction. This diagram specifically highlights the integration and collaboration of the four main modules: Environmental Perception, Localization, Control & Navigation, and Passenger Interaction. The purpose of the following UML diagrams is to offer a structured visualization of the manner in which these integral modules collaborate within the driverless car system, thereby demonstrating the flow of data and control necessary to ensure safe and efficient autonomous driving experiences.

1. Use case diagram

The use case diagram presented below depicts the three primary operations of Environmental Perception, Localization, Control and Navigation, as well as an additional operation for Passenger Interaction, all of which are included in the DC system.

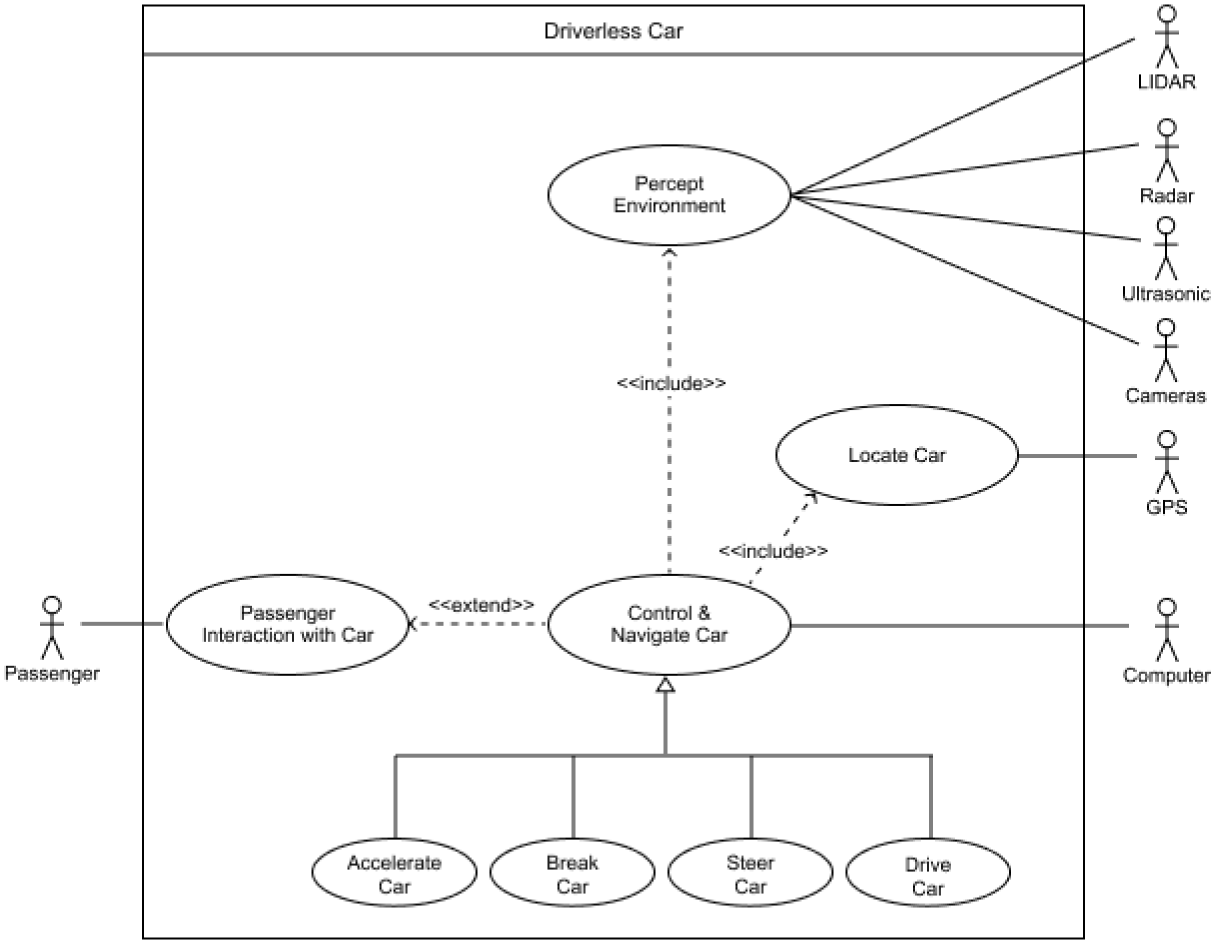

Figure 2. Use case diagram of DC.

2. Activity diagram

The activity diagram for the DC to drive to the destination, and navigate through any possible obstacles is illustrated in the figure shown below. It can be observed, that the following actions run parallelly: Generate coordinates values for the current environment, and Control and Generate local maps. Simple open-headed arrows depict the transition of control from one action to the next. Tiny squares with arrows and tiny squares joined by open-headed arrows (which are semantically equivalent) represent the movement of things, which can be data. Actions that include both incoming object and control flows are executed when both the control and the object are present.

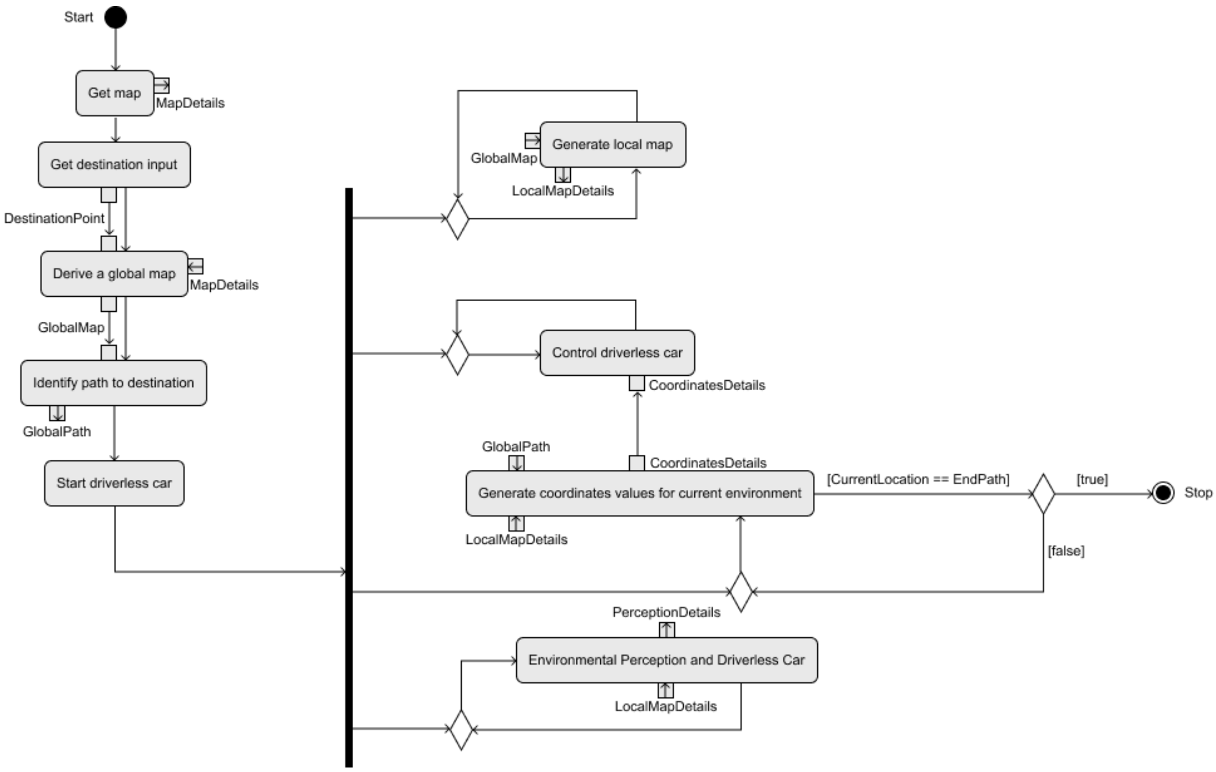

Figure 3. Activity diagram of full DC drive.

3. Class diagram

Figure 4 shows a UML class diagram that previews DC software with the relevant processes, including the required classes, subclasses, and relationships for implementation chosen operations (Environmental Perception, Localization, Control and Navigation, and additional operation of Passenger Interaction).

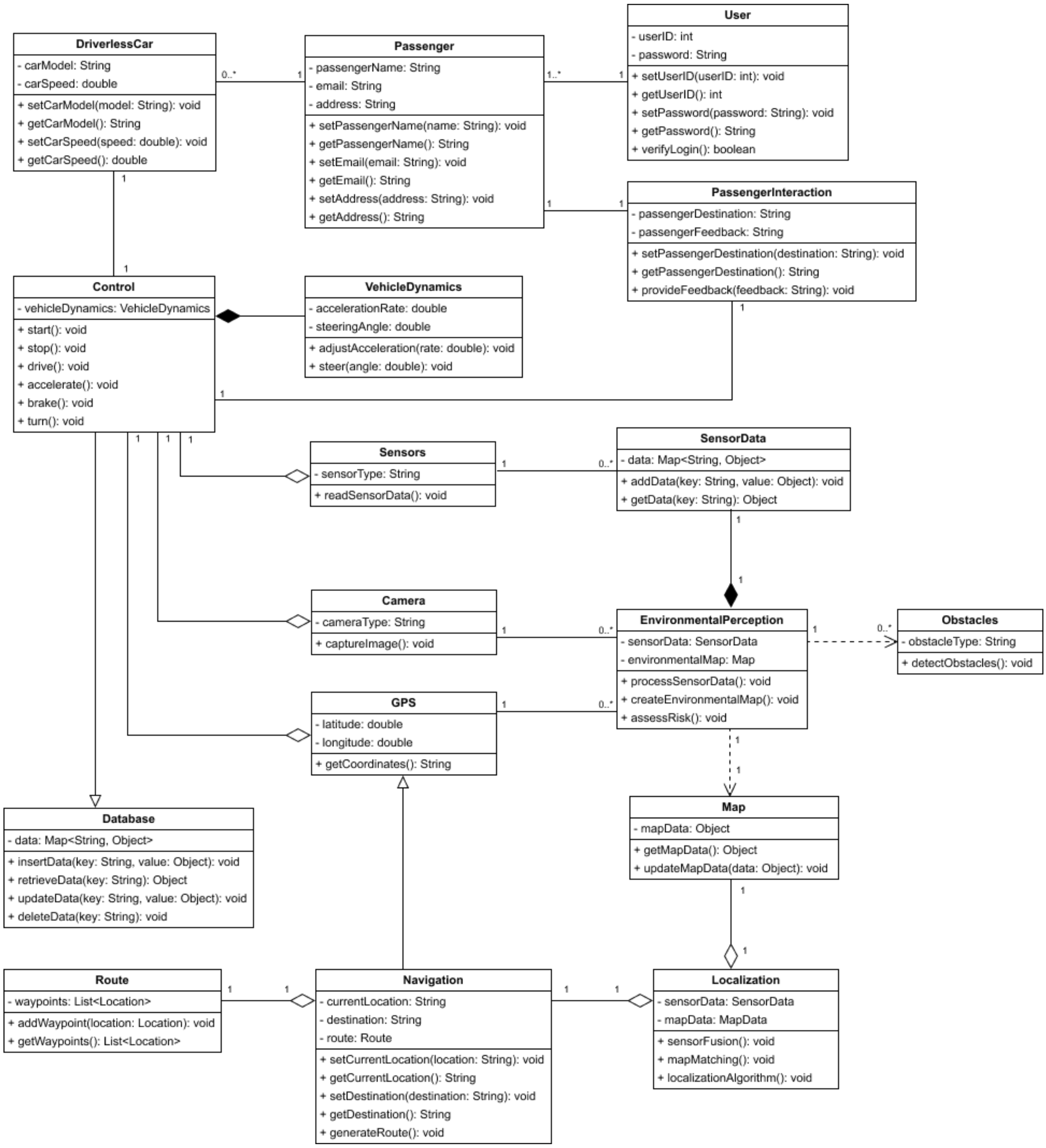

Figure 4. Class diagram

4. Sequence diagram

The sequence of interactions by setting the destination between the passenger and interactions of the DC system with other subsystems are shown in the sequence diagram in Figure 5. And Figure 6 represents sequence diagram of adjusting direction interacting between Controller, Database, VehicleDynamic and Sensosrs.

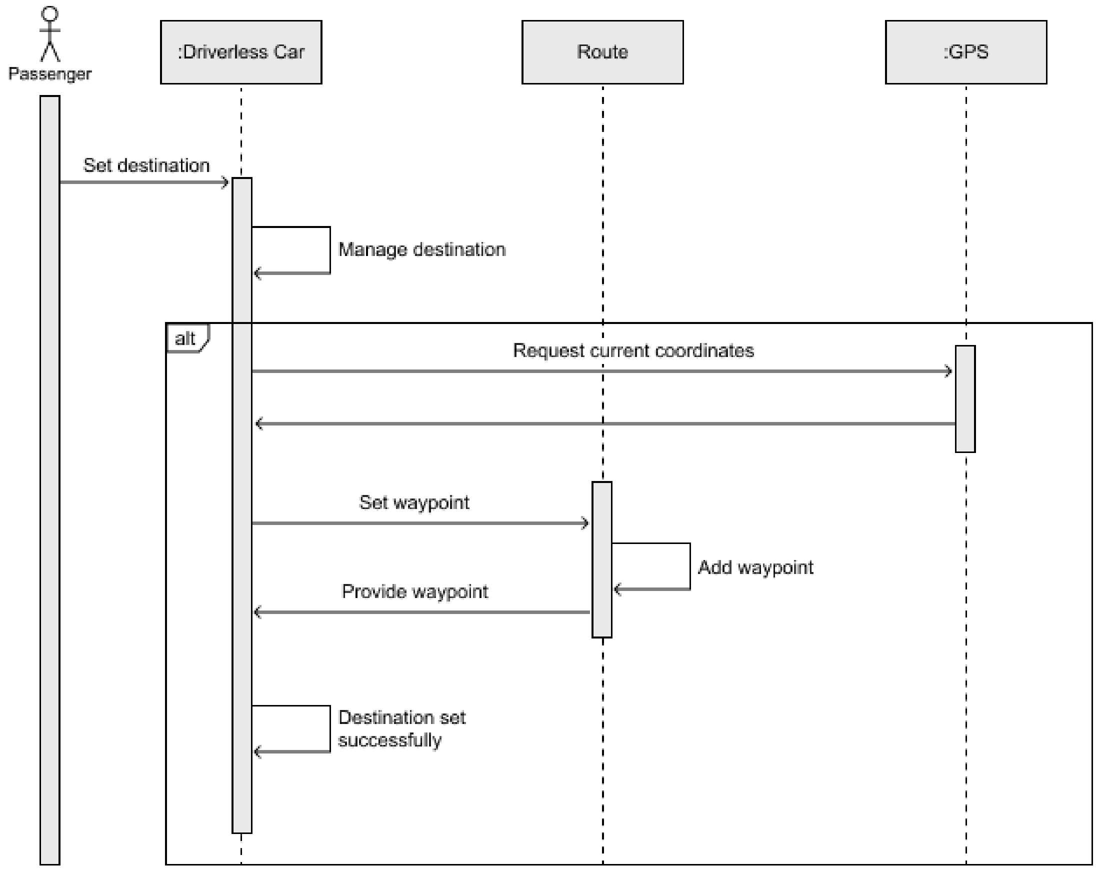

Figure 5. Set destination sequence diagram

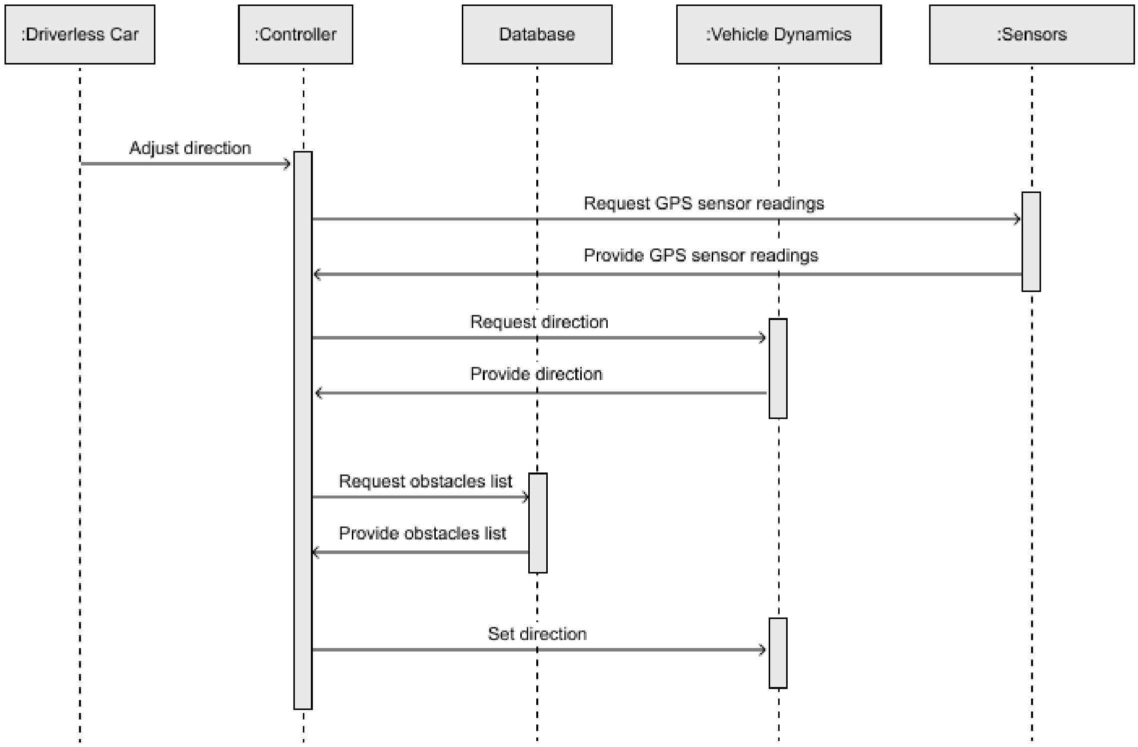

Figure 6. Adjust direction sequence diagram

5. State Transition Diagram

Figure 7 illustrates the route updates that occur when the path predicted by the algorithm differs from the actual route. If no data structure holds for the routes, every route must be checked to determine whether it is still possible according to the user's last action. These structures allow for the removal of previously established routes when several possible paths are synchronized with the past movement of the user.

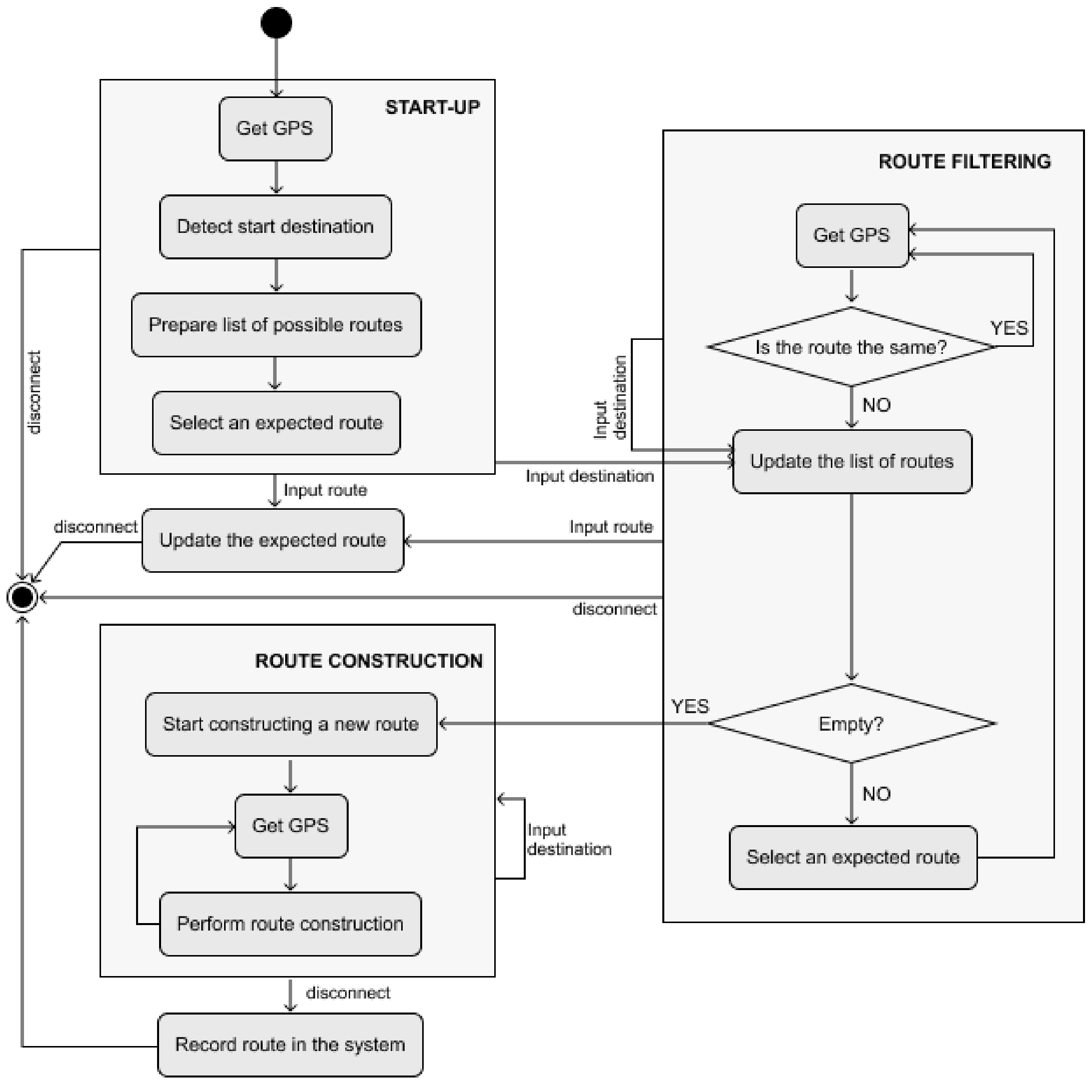

Figure 7. State diagram

Data Structures

Object-oriented programming generally employs data structures such as arrays, lists, stacks, queues, dictionaries, trees, and graphs. These data structures are used to store and manipulate collections of data in an organized and efficient manner. Based on the designed proposal of software to support operations of the DC, several data structures are planned to be used within the context of the classes and their functionalities:

1. **Lists** are useful for storing collections of objects where the order of the elements matters, and duplicates are allowed. In the context of the classes and their functionalities, the Route class may use a list (List<Location>) to maintain waypoints representing the sequence of locations to be visited along a route. The EnvironmentalPerception or Sensors classes could also use lists (List<SensorData>) to store sensor readings or data collected over time.
2. **Queues** follow the First-In-First-Out (FIFO) principle, where the first element added is the first one to be removed. The Navigation class might use a queue (Queue<String>) to manage a queue of destinations waiting to be reached or processed in sequence. The Control class could use a queue (Queue<Double>) to manage a queue of acceleration or braking commands to be executed in sequence.
3. **Dictionaries**, also known as maps, associate keys with values, allowing efficient lookup and retrieval of data based on a unique key. They can be used in various classes, such as the database class, which uses a dictionary (Map<String, Object>) to store and manage data with key-value pairs, enabling operations like inserting, retrieving, updating, and deleting data based on keys. Additionally, the SensorData class could internally use a dictionary

(Map<String, Object>) to store sensor readings with associated metadata or attributes for efficient access and retrieval.

Reference list

Bathla, G., Bhadane, K., Singh, R.K., Kumar, R., Aluvalu, R., Krishnamurthi, R., Kumar, A., Thakur, R.N. & Basheer, S. (2022) ‘Autonomous Vehicles and Intelligent Automation: Applications, challenges, and opportunities’, Mobile Information Systems, 2022, pp. 1–36. doi:10.1155/2022/7632892.

L’Afflitto, A., İnalhan, G. & Shin, H.-S. (2024) Control of autonomous aerial vehicles: Advances in Autopilot Design for civilian uavs. Cham, Switzerland: Springer.

Leiss, P.J. (2023) The functional components of Autonomous Vehicles, The Functional Components of Autonomous Vehicles. Available at: https://www.robsonforensic.com/articles/autonomous-vehicles-sensors-expert [Accessed: 19 April 2024].

Parekh, D., Poddar, N., Rajpurkar, A., Chahal, M., Kumar, N., Prasad Joshi, G.P., & Cho, W. (2022) ‘A review on Autonomous Vehicles: Progress, methods and challenges’, Electronics, 11(14), p. 2162. doi:10.3390/electronics11142162.

Pendleton, S., Andersen, H., Du, X., Shen, X., Meghjani, M., Eng, Y.H., Rus, D. & Ang, M.H. (2017) ‘Perception, planning, control, and coordination for Autonomous Vehicles’, Machines, 5(1), p. 6. doi:10.3390/machines5010006.

Reddy, P. P. (2019) Driverless Car: Software Modelling and Design using Python and Tensorflow.

Sjafrie, H. (2020) Introduction to self-driving vehicle technology. Boca Raton (Fla.): CRC Press Taylor & Francis Group.

Thakurdesai, H.M. & Aghav, J.V. (2020) ‘Autonomous cars: Technical challenges and a solution to blind spot’, Advances in Intelligent Systems and Computing, pp. 533–547. doi:10.1007/978-981-15-1275-9_44.

Zhao, J., Liang, B. & Chen, Q. (2018) ‘The key technology toward the self-driving car’, International Journal of Intelligent Unmanned Systems, 6(1), pp. 2–20. doi:10.1108/ijius-08-2017-0008.

Zhou, Z. Q. & Sun, L. (2019) Metamorphic testing of driverless cars. Commun. ACM 62(3): 61–67. DOI: https://doi.org/10.1145/3241979.