PROJECT 1

Garage Parking System

Name: Cavin Kgwaile Mashilangwako

Student email: EDUV4949119@vossie.net

Student number: EDUV4949119

Project number: 1-Garage Parking System

Faculty: Information Technology

Module: Software Engineering

Module code: ITSEA0

Total marks: 100

Table of context:

Section A: Question 1

1. Actors participating in garage parking system:

- Registered client
- Unregistered client
- Website/GUI
- Payment gateway
- Number plate reader.
- Admin
- Camera in
- Camera out
- Parking sensor
- Debit machine (exit)

2. Use cases for garage parking system:

- Register account.
- Store data.
- Create profile.
- Make reservation.
- Pay online.
- Scan number plate.
- Send mail.
- Issue ticket.
- Issue ticket.
- Store sensor data.
- Debit cash card.

3. Scenarios for garage parking system

Primary scenario: Registered client reservation (daily)

The client will be able to book a reservation for a specific time and date. The system will check whether the parking slot in available, if that is true then the client will be directed to a payment gateway where they will be able to make a payment to finalise their reservation. Th admin will receive a notification that the client has booked and paid for a reservation. The client's details will now be in the system so is his reservation. The admin then must send the client a confirmation email to the client with parking details attached.

Alternative: Registered client reservation(monthly)

All the steps I the primary scenario will apply except in this scenario the parking of the client will be reserved for a constant period of 30 days in specified time durations.

Alternative: Registered client using a temporary number plate

In this case the client using a different number plate than the one they registered with will have to go into their profile 24hrs prior to them using the parking. In their profile they should specify that they will be using a temporary number plate for the day and enter the plate number that they will be using.

Alternative: Registered client overstays reservation

At the exit gate of the garage the client's ticket or plate number will be scanned and they will have to pay an extra fee either on the debit machine or it will be directly deducted from their account using their credit card details.

Alternative: the registered client did not specify their temporary number plate

If the client did not specify their temporary number plate, the client will not be given access to their reservation. Instead, they will then have to use the garage parking as an unregistered student.

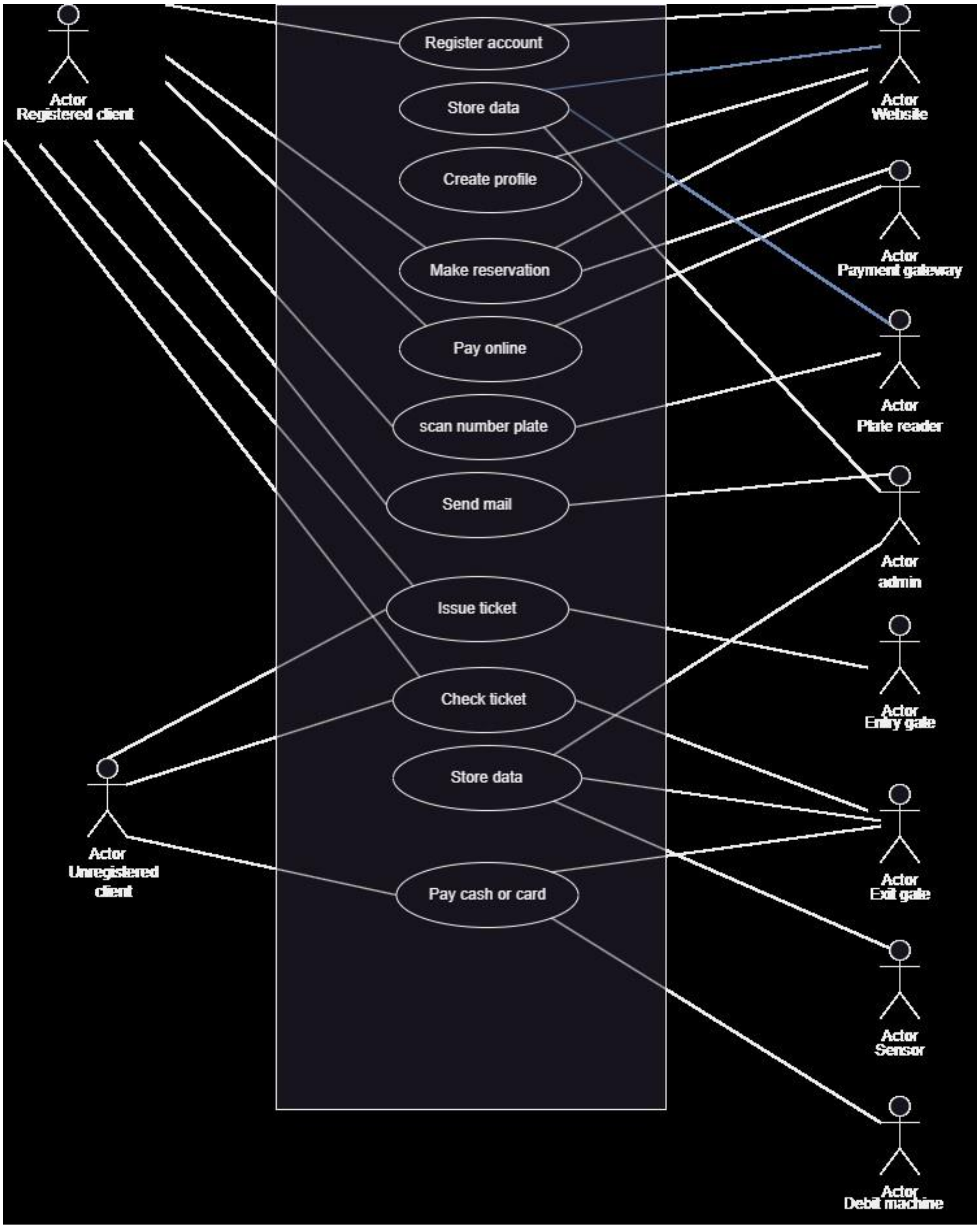
Alternative: Unregistered client (default method)

The client will be issued a ticket at the garage entry, the ticket will specify where they should park. When the client exits, they will pass by a debit machine before the exit gate. They will then be charged according to how low they were parked. The number plate reader also monitors the duration by clocking their entry and exit times. After payment is made, they will be able to leave.

Alternative: Unregistered client (parking is fully occupied

Client will be given a free exit pass, so that they can turn around to the exit and leave without paying. This will also help with the flow of traffic.

4. Use case diagram:



Use case diagram documentation:

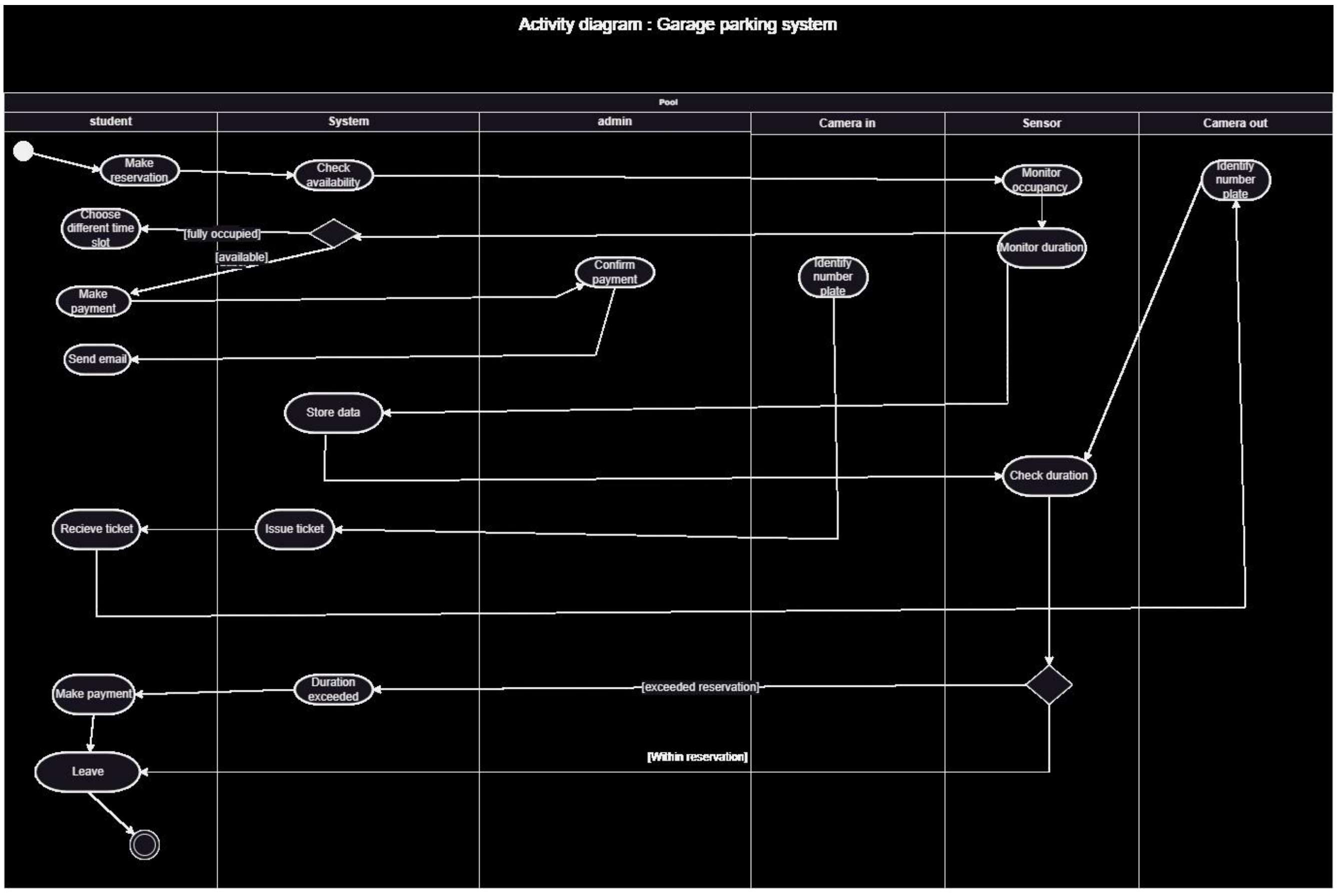
The registered client will register an account, their details are then stored, and they are given a profile. The client makes a reservation through the profile, once that is done, they can now pay online to confirm their reservation. When they enter the garage parking the plate reader scans their plate and they are issued a ticket with the reservation details. Upon exit their plate/ticket is scanned again to check the duration parked, if the overstayed they must pay extra, if not they may leave.

As for the unregisted client they will be issued a ticket at the garage entry, the ticket will specify where they should park. When the client exits, they will pass by a debit machine before the exit gate. They will then be charged according to how low they were parked. The number plate reader also monitors the duration by clocking their entry and exit times. After payment is made, they will be able to leave.

5. Activity diagram

Activity diagram documentation:

After registering on the system, the student can now make a reservation. The system is going to check if the reservation selected is available. If the reservation slot is available, it will be reserved, if not they will be prompted to insert a different slot. After confirming the slot, the student must then make an online payment. The payment is confirmed by the admin and the student is sent a confirmation email with reservation details. When the student arrives at the garage entry Camera in will scan number plate and they will be issued a ticket with parking details. When exiting Camera out will scan number plate. If the reservation was overstayed then they will be charged extra, if not they can leave without paying. **Activity diagram:**



6. Class diagram

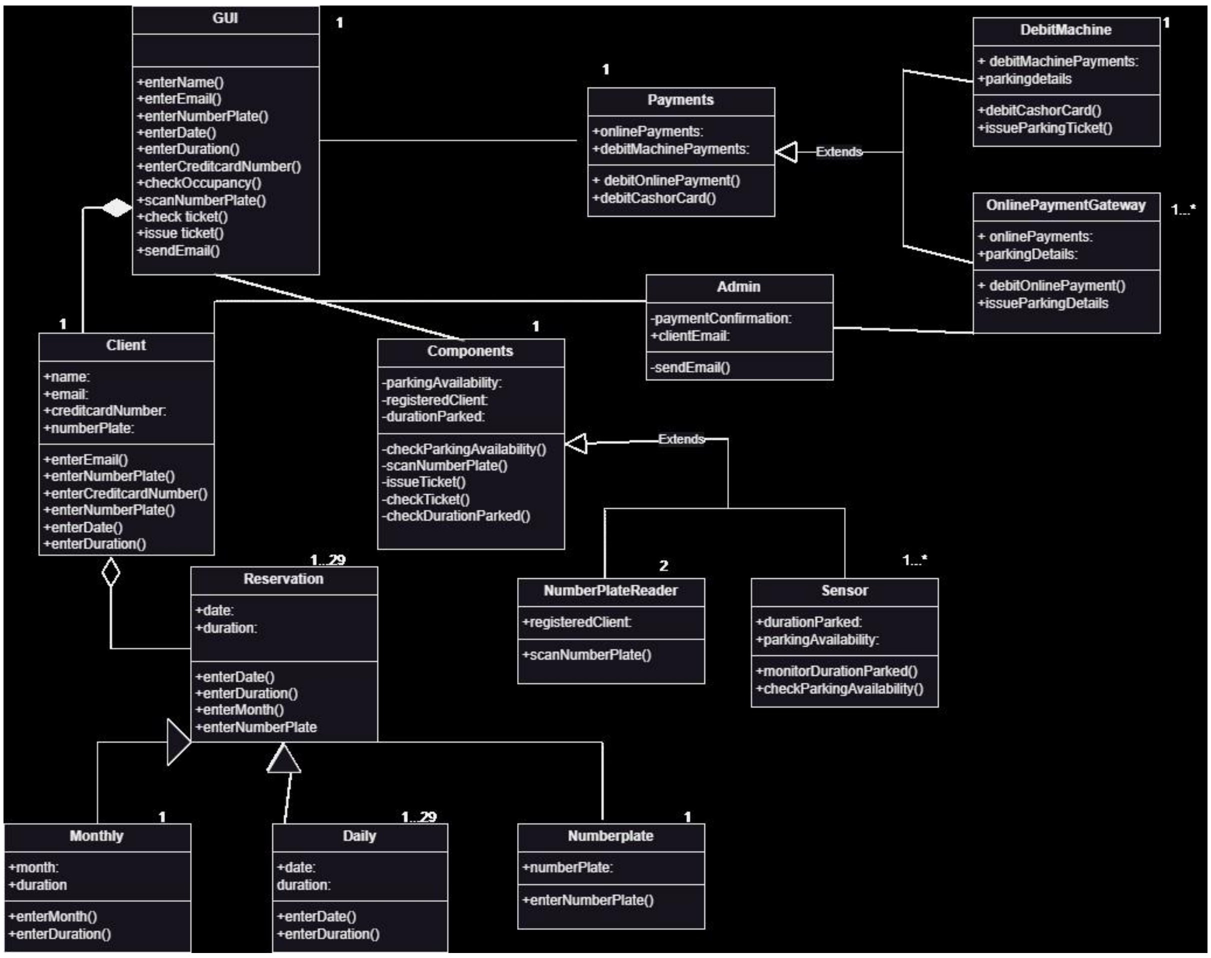
<u>Class diagram documentation:</u>

Since the client registers online a GUI class must be created for the client. The GUI is then associated with the client through aggregation. A client class is created containing the name, email, credit card number, and the number plate number of the client. There is also a reservation class which breaks down into a monthly, daily, and number plate subclasses. The number plate sub class is used in cases where the client is using a temporary vehicle with a different number plate.

There is also a component class which splits into the "number plate" and "sensor" classes. The system also has an admin class for the emails that must be sent to the client, confirming their reservation. The payment class is also split into two sub-classes, the "debit machine" and "online payment gateway" classes.

The daily class can take up to twenty-nine reservations, the monthly class can take 1 reservation consisting of consistent days.

Class diagram:

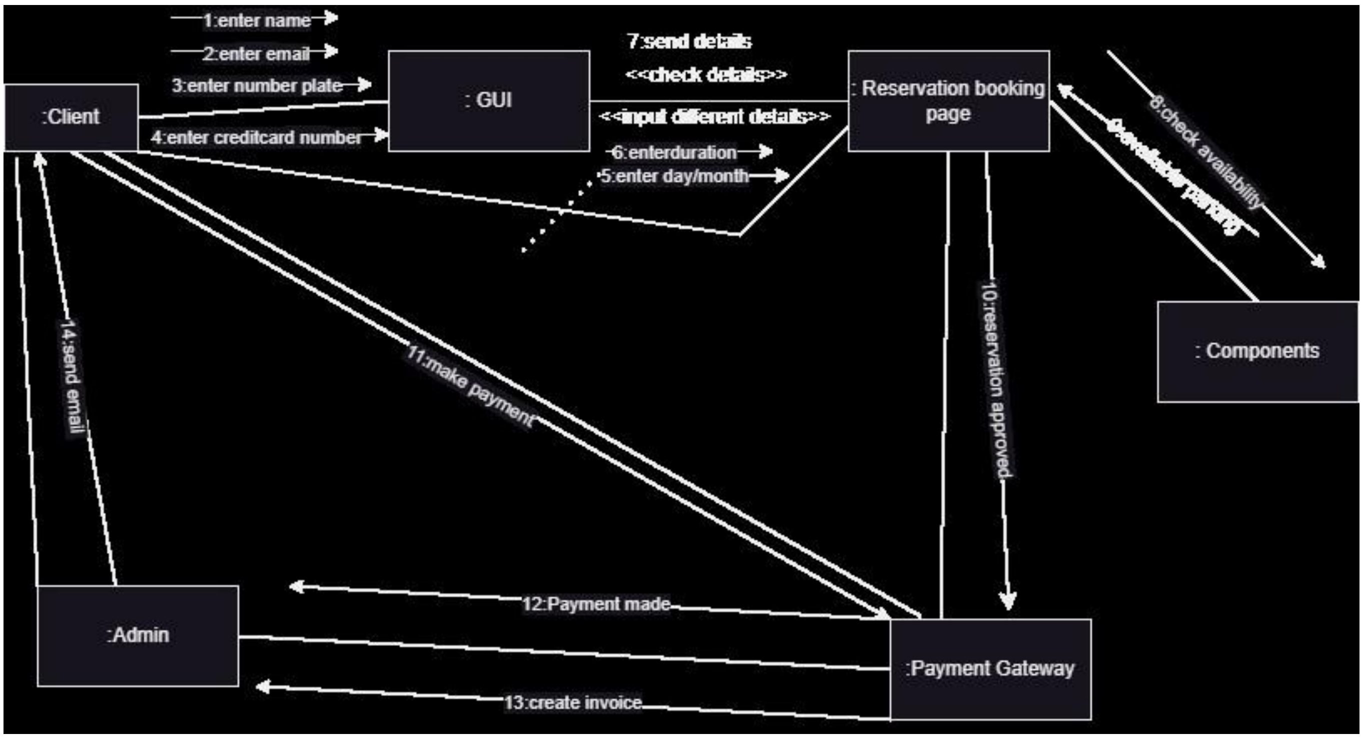


7. Communication diagram

Communication diagram documentation

The client inputs their name, email address, number plate and credit card number to register an account on the system through the GUI. The date and duration of the reservation is then sent to the reservation booking page by the system. The reservation booking page communicates with the parking garage components to check if the reservation slot entered is available. If not, the client must enter a different date or time until slot is found after the booking is done the client will be directed to a payment gateway where they will pay for the reservation. The payment is then recognised by the admin, the admin sends the client a confirmation email with the clients booking details.

Communication diagram:

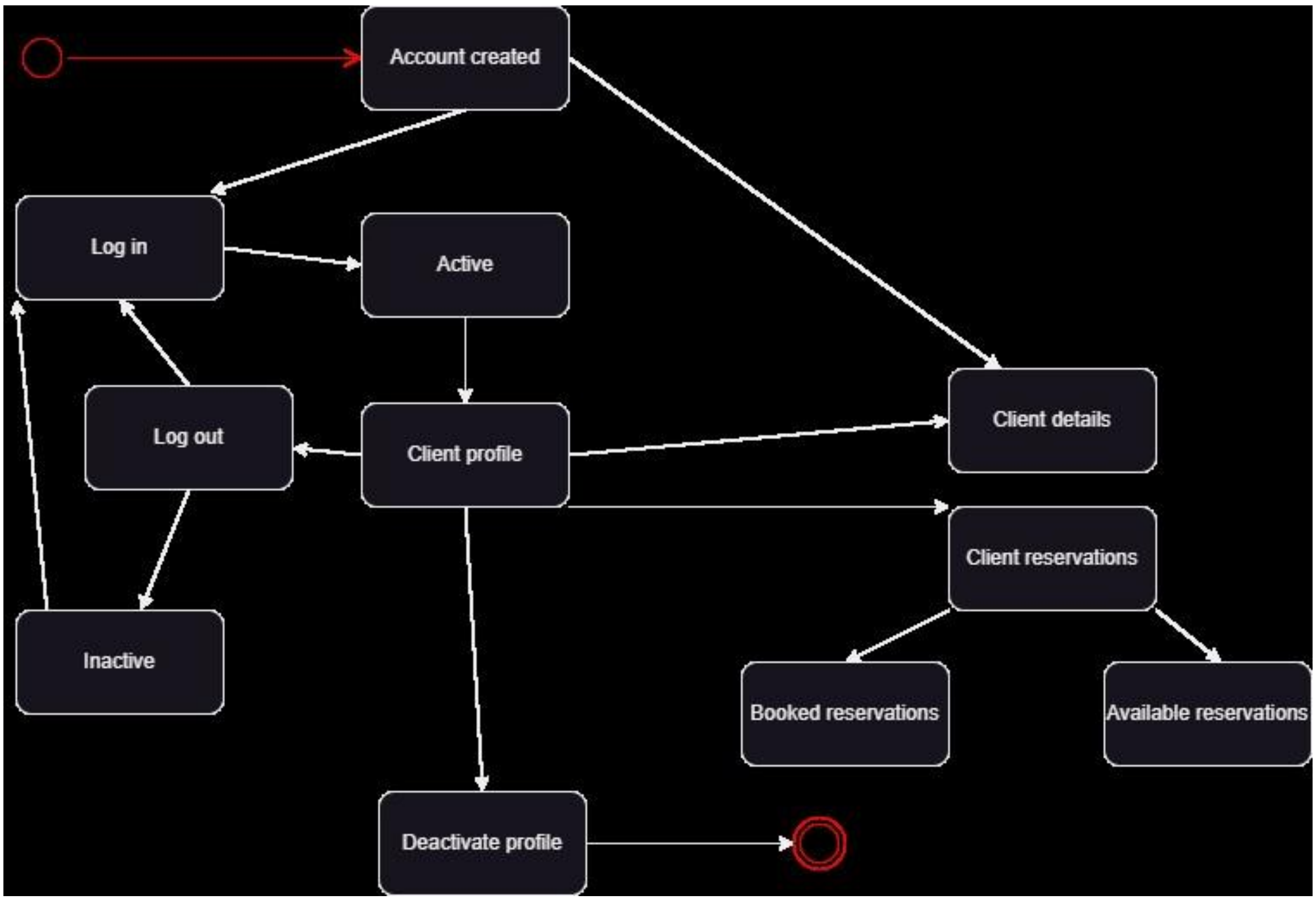


8. State machine diagram

State machine diagram document:

The client's Account can be in different states after creation. The account can be active in this state the user can access their reservation and details. The account can be inactive this is a state where the client is not using the account. The account will only be active once the client logs in to their profile. In the client's profile they have an option to deactivate their profile and account which would be the end state of the account, then an account will be created, this is the start of a new account. The reservations can also be made through the student's profile.

State machine diagram:



9. Five types of software development methodologies

Waterfall Model:

The Waterfall Model is one of the most dated and most straightforward methodologies in software development. It follows a sequential, linear approach, each phase must be completed before moving on to the next Requirements, Design, Implementation, Testing, Deployment, and Maintenance. It is suitable for projects where requirements are well-understood and unlikely to change significantly during the development process. However, it can be less flexible in accommodating changes or feedback from stakeholders once development has started.

Agile Methodology:

Agile is an iterative and incremental approach to software development. It emphasizes flexibility, adaptability, and collaboration among cross-functional teams. Agile methodologies include Scrum, Kanban, Extreme Programming (XP), and more, each with its own set of practices and principles. Agile encourages delivering working software in small, frequent increments, allowing for continuous feedback and improvement. It is well-suited for projects with evolving requirements, where customer needs may change over time.

Scrum:

Scrum is a specific Agile framework that focuses on delivering software in short, time-boxed iterations called Sprints. It emphasizes teamwork, transparency, and frequent inspection and adaptation. Scrum roles include Product Owner, Scrum Master, and Development Team, each with specific responsibilities. Key Scrum artifacts include the Product Backlog, Sprint Backlog, and Increment. Daily Scrum meetings, Sprint Planning, Sprint Review, and Sprint Retrospective are essential Scrum events.

Kanban:

Kanban is another Agile methodology that visualizes the workflow and limits work in progress. It aims to continuously improve efficiency and flow by identifying bottlenecks and optimizing the delivery process. Kanban boards visualize work items as cards moving through various stages. Work items are pulled from one stage to the next as capacity allows, ensuring a steady flow of work. Kanban encourages limiting WIP to focus on completing tasks before starting new ones, thereby reducing multitasking, and increasing throughput.

DevOps:

DevOps is a culture, set of practices, and automation approach that aims to unify development and operations teams to deliver high-quality software more rapidly and reliably. It emphasizes collaboration, communication, and automation throughout the software development lifecycle (SDLC). DevOps practices include Continuous Integration (CI), Continuous Delivery (CD), infrastructure as code, and automated testing. By breaking down silos between development and operations, DevOps aims to streamline processes, reduce lead time, and improve overall software quality and reliability.

9.1) Out of the five methodologies, I believe that the Agile methodology will be the best fit /best suited for the garage parking project, particularly the Scrum framework.

10. Motivate answer

Overall, adopting Agile methodologies, particularly Scrum, would provide adaptability, and collaborative framework necessary to effectively manage the development of the parking garage system while addressing the challenges of occupancy tracking and reservation management. Because it uses iterate development, flexibility, cross-functional teams, time-boxed sprints, and regular feedback.

Bibliography

- **Booch94**
  Grady Booch, *Object-Oriented Analysis and Design with Applications*, 2nd ed., Benjamin/Cummings, 1994.
- **BRJ97**
  Grady Booch and James Rumbaugh and Ivar Jacobson, *Unified Modeling Language User Guide*, Addison-Wesley Object Technology Series, Summer 1998.
- **FS97**
  Martin Fowler with Kendall Scott, *UML Distilled: Applying the Standard Object Modeling Language*, Addison-Wesley Object Technology Series, 1997.
- **JBR97**
  Ivar Jacobson and Grady Booch and James Rumbaugh, *The Objectory Software Development Process*, Addison-Wesley Object Technology Series, Summer 1998.
- **JCJO92**
  Ivar Jacobson and Magnus Christerson and Patrik Jonsson and G. G.
- **MyLMS, Software engineering course**