Project Design Phase-II

Data Flow Diagram & User Stories

Date	19 October 2023
Team ID	Team - 592813
Team Members	Sugandhi Ninad Nilesh Aadhith M
Project Name	Project - AI Enabled Car Parking System using OpenCV
Maximum Marks	4 Marks

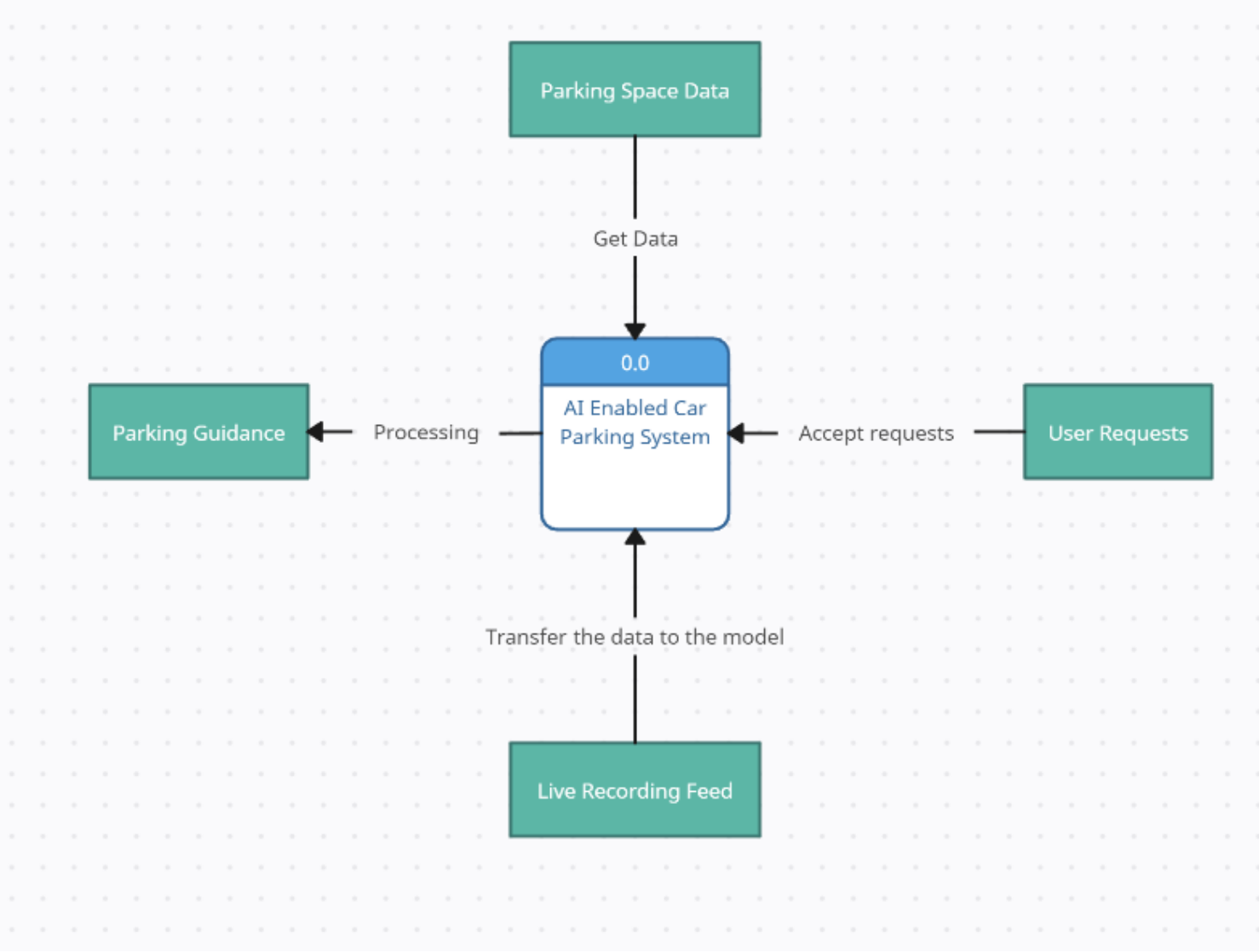
Overview On Data flow diagram:

DFD is visual representation of data which flows within a system. It can be manual, automated or combination as well. The data flow diagram has been divided into 3 levels such as level 0, level 1 and level 2. Where level 0 focusses on a single process which has other external entities connected to it. The level 0 is the basic or fundamental part of the DFD. The level 1 consists of all the important processes involved in the AI Automated Car Parking system and connects each component using dataflow. The level 2 is an extended version of level 1 DFD where each process involved in the system is further classified into separate functions.

LEVEL 0 DFD:

A Level 0 Data Flow Diagram (DFD) is a visual representation of a system that provides a fundamental overview of the flow of data within the system.

The Level 0 DFD is given below:



Description:

The DFD has three external entities:

1. User Requests: This entity represents the requests that users make to the parking system, such as requests to find a parking spot or to reserve a parking spot.

2. Live Recording Feed: This entity represents the live video feed from the parking lot. This feed can be used to monitor the parking lot and to identify available parking spots.

3. Parking Guidance: This entity represents the guidance that the parking system provides to users, such as directions to the parking lot or directions to an available parking spot.

The DFD also has three internal processes:

1. Accept requests: This process accepts the requests from the User Requests entity.

2. Transfer the data to the model: This process transfers the data from the User Requests entity and the Live Recording Feed entity to the parking space model.

3. Processing: This process processes the data in the parking space model to generate the Parking Guidance entity.

The DFD also has two data stores:

1. Parking Space Data: This data store stores information about the parking spaces in the parking lot, such as the location of each parking space and whether or not it is available.

2. Parking Space Model: This data store stores the parking space model, which is a representation of the parking lot that is used to generate the Parking Guidance entity.

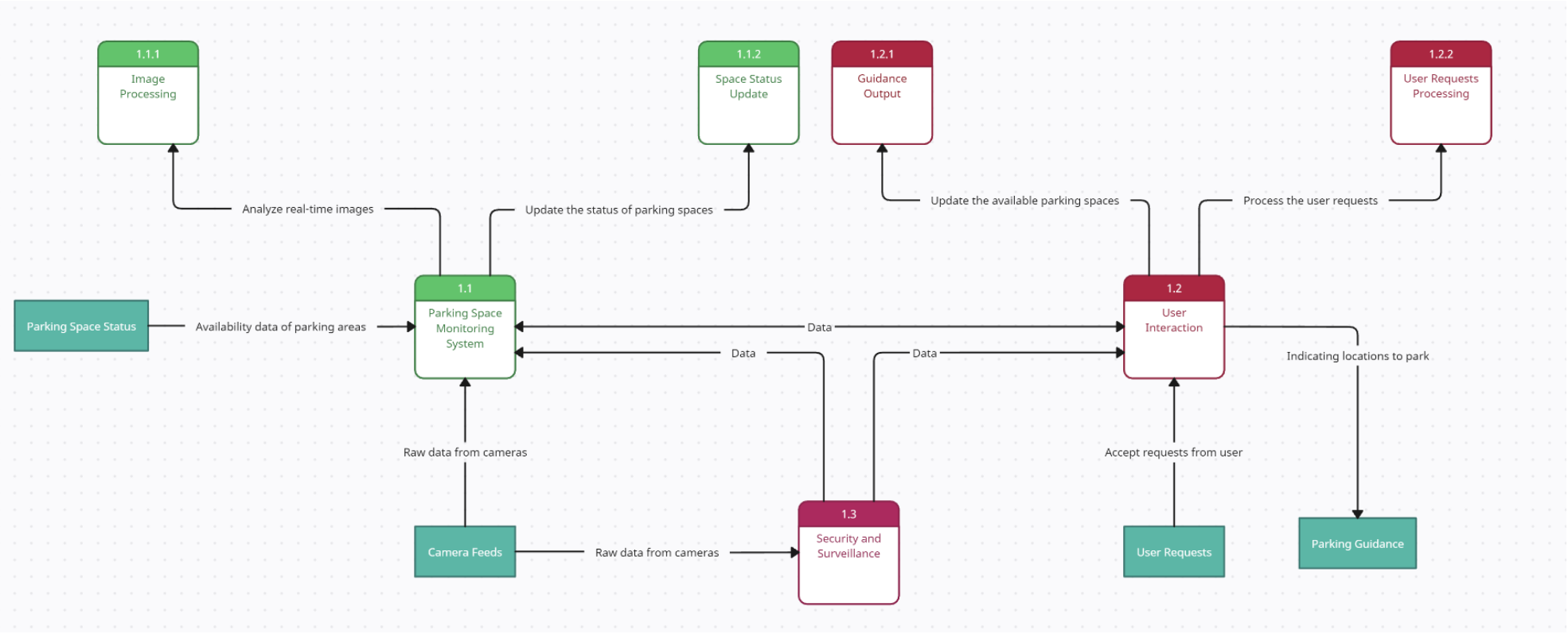
The following is a step-by-step explanation of how the DFD works:

1. The user makes a request to the parking system, such as a request to find a parking spot.
2. The “Accept requests” process accepts the request from the user.
3. The “Transfer the data to the model” process transfers the data from the User Requests entity and the Live Recording Feed entity to the parking space model.
4. The “Processing” process processes the data in the parking space model to generate the Parking Guidance entity.
5. The Parking Guidance entity is provided to the user.

LEVEL 1 DFD:

At Level 1, the DFD delves into each major process from Level 0, breaking them down into sub-processes and data flows.

The Level-1 DFD is given below:



Description:

The DFD has three external entities:

1. Raw data from cameras: This entity represents the raw video feed from the cameras in the parking lot.

2. User requests: This entity represents the requests that users make to the parking system, such as requests to find a parking spot or to reserve a parking spot.

3. Guidance: This entity represents the guidance that the parking system provides to users, such as directions to the parking lot or directions to an available parking spot.

The DFD also has three internal processes:

1. Analyse real-time images: This process analyses the raw video feed from the cameras to identify available parking spots and to track the movement of vehicles in the parking lot.

2. Update the status of parking spaces: This process updates the status of the parking spaces in the parking system based on the information from the Analyze real-time images process.

3. Process user requests: This process processes the requests from the User requests entity and generates the Guidance entity.

The DFD also has one data store:

Parking space availability: This data store stores information about the availability of the parking spaces in the parking lot.

The following is a step-by-step explanation of how the DFD works:

1. The “Analyse real-time” images process analyses the raw video feed from the cameras to identify available parking spots and to track the movement of vehicles in the parking lot.

2. The “Update the status of parking spaces” process updates the status of the parking spaces in the parking system based on the information from the Analyze real-time images process.

3. The “Process user requests” process processes the requests from the User requests entity and generates the Guidance entity.

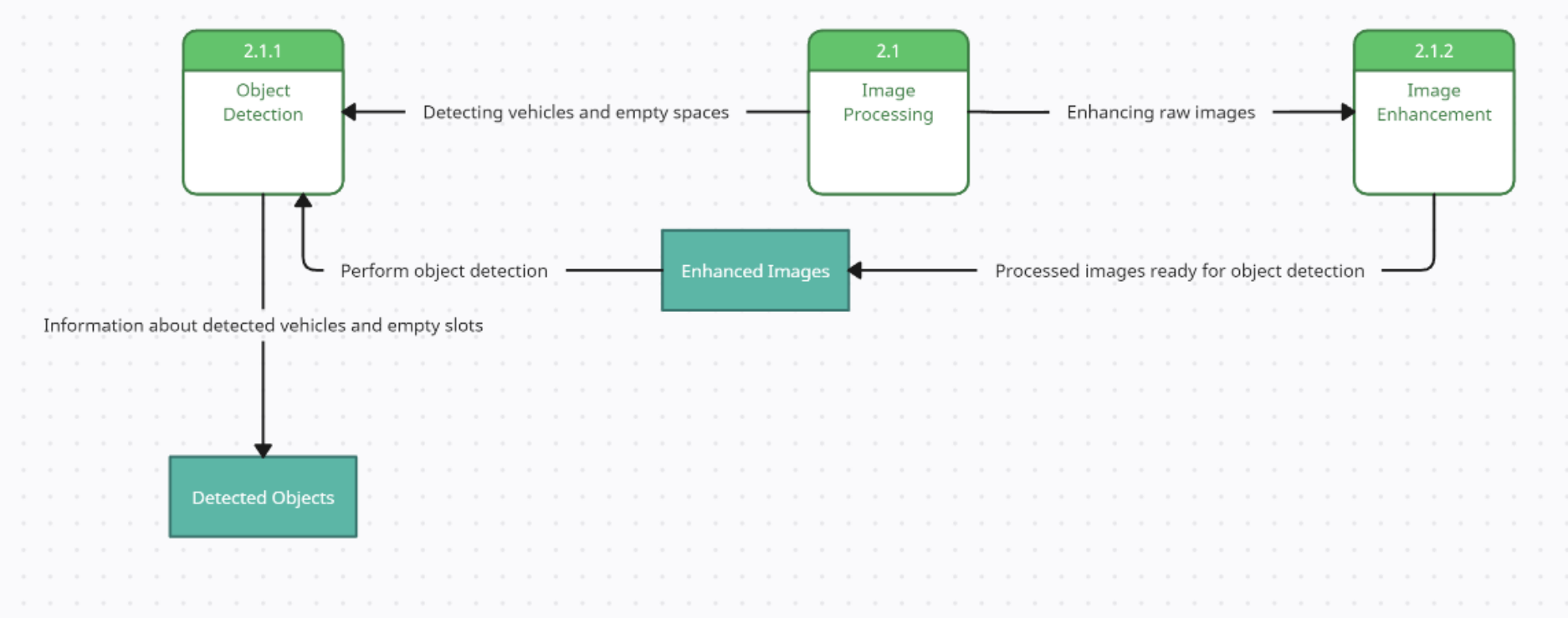
4. The Guidance entity is provided to the user.

LEVEL 2 DFD:

At Level 2, the DFD further breaks down Level 1 processes into detailed sub-processes, providing a granular view of the system's operations.

The Level-2 DFDs are given below:

2.1. Image Processing Sub-Process:



Description:

The DFD has three external entities:

1. Input image: This entity represents the image that is input to the system.
2. Output image: This entity represents the image that is output from the system.
3. Detected objects: This entity represents the objects that have been detected in the image.

The DFD also has three internal processes:

1. Enhance image: This process enhances the input image to improve the accuracy of the object detection process.
2. Detects objects: This process detects the objects in the input image.
3. Label objects: This process labels the detected objects in the image.

The DFD also has one data store:

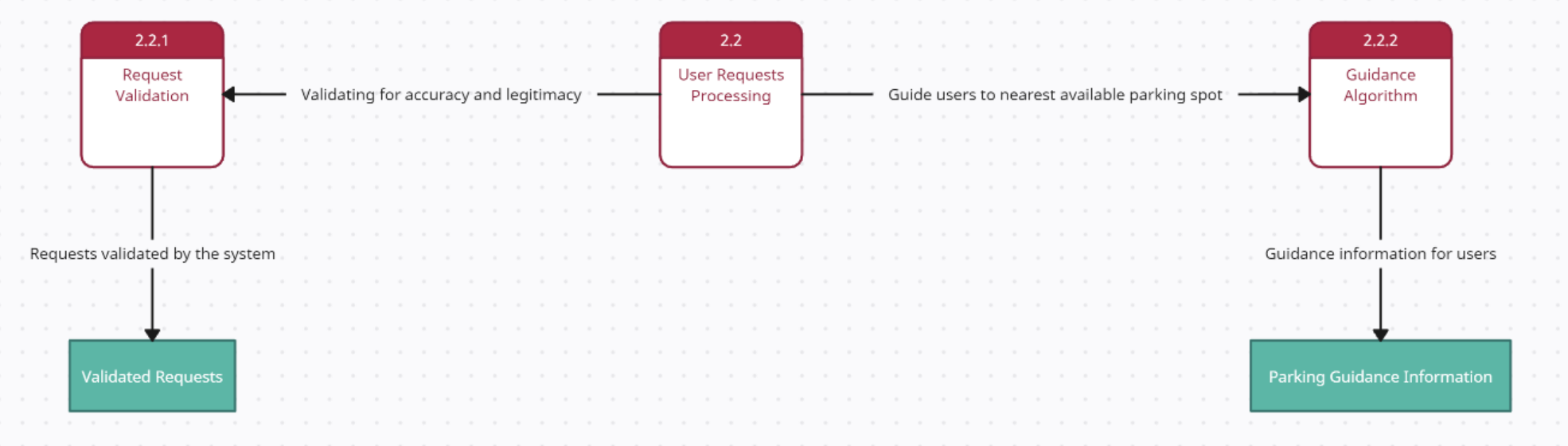
Object database: This data store contains information about the objects that the system can detect.

The following is a step-by-step explanation of how the DFD works:

1. The “Enhance image” process enhances the input image to improve the accuracy of the object detection process.
2. The “Detect objects” process detects the objects in the input image.
3. The “Label objects” process labels the detected objects in the image.
4. The “Output image” entity is output from the system.

5. The “Detected objects” entity is output from the system.

2.2. User Requests Processing Sub-Process:



Description:

The DFD has three external entities:

1. User requests: This entity represents the requests that users make to the parking system, such as requests to find a parking spot or to reserve a parking spot.

2. Parking lot data: This entity represents the data about the parking lot, such as the location of the parking lot, the number of parking spaces in the parking lot, and the availability of parking spaces in the parking lot.

3. Guidance: This entity represents the guidance that the parking system provides to users, such as directions to the parking lot or directions to an available parking spot.

The DFD also has three internal processes:

1. Process user requests: This process processes the requests from the User requests entity and generates a query to the Parking lot data entity.

2. Query parking lot data: This process queries the Parking lot data entity to determine the availability of parking spaces.

3. Generate guidance: This process generates guidance for the user based on the results of the Query parking lot data process.

The DFD also has one data store:

Parking lot data: This data store stores information about the parking lot, such as the location of the parking lot, the number of parking spaces in the parking lot, and the availability of parking spaces in the parking lot.

The following is a step-by-step explanation of how the DFD works:

1. The user makes a request to the parking system, such as a request to find a parking spot.

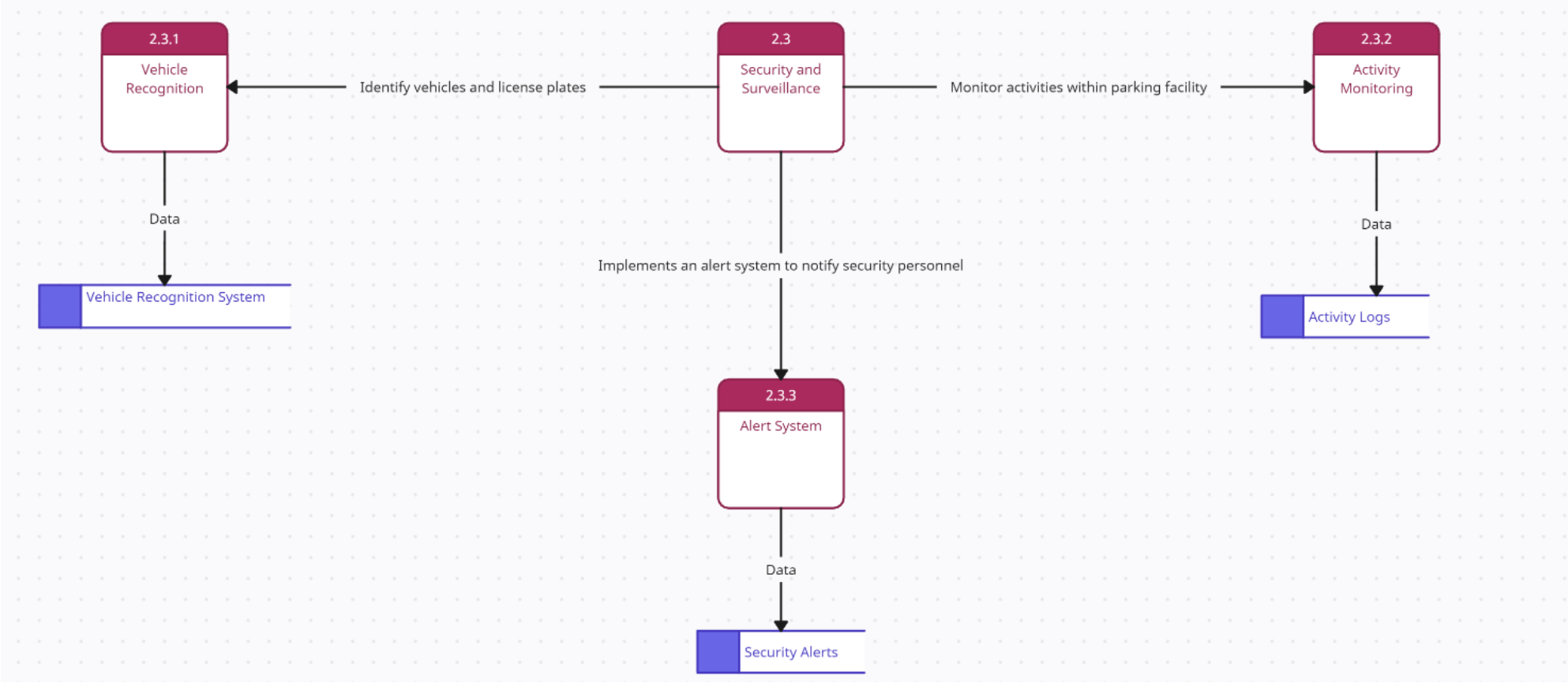
2. The “Process user requests” process processes the request from the user and generates a query to the Parking lot data entity.

3. The “Query parking lot data” process queries the Parking lot data entity to determine the availability of parking spaces.

4. The “Generate guidance” process generates guidance for the user based on the results of the Query parking lot data process.

5. The Guidance entity is provided to the user.

2.3. Security and Surveillance Sub-Process:



The DFD has three external entities:

1. Raw data from cameras: This entity represents the raw video feed from the cameras in the parking lot.

2. User requests: This entity represents the requests that users make to the parking system, such as requests to find a parking spot or to reserve a parking spot.

3. Guidance: This entity represents the guidance that the parking system provides to users, such as directions to the parking lot or directions to an available parking spot.

The DFD also has three internal processes:

1. Analyse real-time images: This process analyses the raw video feed from the cameras to identify available parking spots and to track the movement of vehicles in the parking lot.

2. Update the status of parking spaces: This process updates the status of the parking spaces in the parking system based on the information from the Analyze real-time images process.

3. Process user requests: This process processes the requests from the User requests entity and generates the Guidance entity.

The DFD also has one data store:

Parking space availability: This data store stores information about the availability of the parking spaces in the parking lot.

The following is a step-by-step explanation of how the DFD works:

1. The “Analyse real-time images” process analyses the raw video feed from the cameras to identify available parking spots and to track the movement of vehicles in the parking lot.

2. The “Update the status of parking spaces” process updates the status of the parking spaces in the parking system based on the information from the Analyze real-time images process.

3. The "Process user requests” process processes the requests from the User requests entity and generates the Guidance entity.

4. The Guidance entity is provided to the user.

User Stories

The below given table describes the user stories for the product

User Type	Functional Requirements (Epic)	User Story Number	User Story/ Task	Acceptance Criteria	Priority	Release
New User	Registration	USN-1	As a new user, I want to create an account on the mobile app to access parking services.	The registration process should include fields for name, email, phone number, and password. Upon successful registration, the user should receive a confirmation email.	High	Sprint-1.0
Registered User	Login	USN-2	As a registered user, I want to log into the mobile app to access parking information and services.	Users should be able to log in using their email and password. The system should validate the credentials and grant access upon successful login.	High	Sprint-1.0
Driver	Parking Space Availability	USN-3	As a driver, I want to easily find an available parking space so that I can park my car without hassle.	The system should provide real-time updates on available parking spaces. The driver should receive clear and accurate guidance to the selected parking space.	High	Sprint-1.0

	Reservation System	USN-4	As a driver, I want to reserve a parking space in advance so that I can ensure a spot during peak hours.	The system should allow users to reserve parking spaces through the mobile app. It should confirm the reservation and provide a QR code for entry. If the user doesn't arrive within the specified time, the reservation should be cancelled and the space made available for others.	Medium	Sprint-1.2
	Profile Management	USN-5	As a driver, I want to manage my profile information such as name, contact details, and vehicle information.	Users should be able to edit their profile details. Changes made should be reflected in the system immediately.	Medium	Sprint-1.1
	Payment Management	USN-6	As a driver, I want to manage my payment methods for parking services.	Users should be able to add, edit, or remove payment methods. The system should securely store payment information and allow users to select a default payment method.	Medium	Sprint-1.1
	Parking History	USN-7	As a driver, I want to view my parking history, including previous parking sessions and payments.	Users should have access to a detailed parking history section displaying past sessions, payment details, and duration of each session.	Medium	Sprint-1.2

	Favourites & Reminders	USN-8	As a driver, I want to mark certain parking facilities as favourites and set reminders for specific parking sessions.	Users should be able to mark preferred parking facilities as favourites for quick access. The system should allow setting reminders for upcoming parking sessions, sending notifications to users.	Low	Sprint-1.2
Parking Facility Operator	Space Utilisation Optimization	USN-9	As a parking facility operator, I want to optimise parking space allocation to accommodate maximum vehicles and enhance revenue.	The system should analyse parking data to suggest optimal space allocations. It should provide reports indicating the optimised space utilisation and revenue increase.	High	Sprint-1.1
	Real-time Occupancy Dashboard	USN-10	As a parking facility operator, I want a real-time dashboard displaying occupancy and availability data.	The system should provide a live dashboard showing the current occupancy status, including the number of occupied and available parking spaces. The dashboard should update in real-time.	High	Sprint-1.0
	User Feedback Analysis	USN-11	As a parking facility operator, I want to collect and analyse user feedback to improve services.	The system should allow users to provide feedback within the app. Feedback data should be collected and analysed for insights. Regular reports should be generated based on feedback analysis.	Medium	Sprint-1.1

Security Personnel	Suspicious Activity Detection	USN-12	As a security personnel, I want to be alerted about any suspicious activities within the parking facility to ensure a secure environment.	The system should monitor parking activities and send real-time alerts in case of unauthorised access or suspicious behaviour.	High	Sprint-1.1
Application Administrator	Data Analytics and Insights	USN-13	As an application administrator, I want to access detailed analytics and insights to understand user behaviour and improve system performance.	The system should provide comprehensive analytics reports, including user interaction patterns, popular parking times, and frequently used features. It should allow exporting these reports in various formats for further analysis.	Medium	Sprint-1.1
	System Monitoring	USN-14	As an administrator, I want to monitor the system's performance, including server health and user activity.	The system should provide logs and reports on server health, user interactions, and error occurrences. Administrators should receive notifications for critical issues.	High	Sprint-1.0
UI/UX Designer	UI Enhancements	USN-15	As a UI/UX designer, I want to enhance the user interface with intuitive design elements and seamless navigation.	The app should undergo design improvements, including user-friendly menus, intuitive icons, and smooth transitions between screens. Users should find the app visually appealing and easy to use.	High	Sprint-1.1