
Software Requirements Specification

for

FPay As You Go

Version 1.0 approved

Prepared by

*Sanjana Rajeshwar 20BCE0782
Shreya Thaplyal 20BCE0814
Arnav Srivastava 20BCE0831
Danish Ansari 20BCE0932*

Vellore Institute of Technology, Vellore

05/02/2023

Table of Contents

1. Introduction	3
1.1 Purpose	3
1.2 Document Conventions	3
1.3 Intended Audience and Reading Suggestions	3
1.4 Product Scope	3
1.5 References	3
2. Overall Description	4
2.1 Product Perspective	4
2.2 Product Functions	5
2.3 User Classes and Characteristics	6
2.4 Operating Environment	6
2.5 Design and Implementation Constraints	7
2.6 User Documentation	7
2.7 Assumptions and Dependencies	7
3. External Interface Requirements	8
3.1 User Interfaces	8
3.2 Hardware Interfaces	8
3.3 Software Interfaces	8
3.4 Communications Interfaces	9
4. System Features	9
4.1 Login	9
4.2 Facial Recognition	9
4.3 Payment System	10
4.4 Notes	11
5. Other Nonfunctional Requirements	11
5.1 Performance Requirements	11
5.2 Safety Requirements	11
5.3 Security Requirements	12
5.4 Software Quality Attributes	12
Appendix A: Glossary	12
Appendix B: Analysis Models	12

Revision History

Name	Date	Reason For Changes	Version
Review 1	08/02/2022	Initial Commits	1.0

1. Introduction

1.1 Purpose

Our concept presents a simple and efficient facial recognition-based payment approach to ensure that all shuttle boarders pay, and that the shuttle services do not suffer any losses

1.2 Document Conventions

The document uses the following conventions:

- Entire document should be justified.
- Convention for Main title
 - Font face: Times New Roman
 - Font style: Bold
 - Font Size: 18
- Convention for Subtitle
 - Font face: Times New Roman
 - Font style: Bold
 - Font Size: 14
- Convention for body
 - Font face: Arial
 - Font style: Italics
 - Font Size: 12

1.3 Intended Audience and Reading Suggestions

This project is a prototype for the college students (VITians); however, it is not restricted within the college premises. This has been implemented under the guidance of college professors. This project is useful for the shuttle service, students, and anyone else who utilizes the services.

1.4 Product Scope

Students often get off the shuttles without paying when they are in a hurry to reach their classes since it is a hustle to make payments using cash for an amount as small as 15 rupees or wait for the mobile data to work to make use of UPI.

There is no way for the shuttle drivers to keep a track of whether the student had paid before leaving to prevent such loss for the shuttle services, we propose a facial recognition-based payment system that will capture the pictures of boarders as they enter the shuttle and keep a detailed account of the payments due.

To ensure that everyone using the shuttle pays and that there are no losses for the shuttle services, our solution provides a simple and reliable facial recognition-based payment method.

1.5 References

- *R.S Pressman, Software Engineering: A Practitioner's Approach, Mc-Graw-Hill, Edition-7e (2010).*

- L. Li, X. Mu, S. Li and H. Peng, "A Review of Face Recognition Technology," in *IEEE Access*, vol. 8, pp. 139110-139120, 2020, doi: 10.1109/ACCESS.2020.3011028.
- M. Arsenovic, S. Sladojevic, A. Anderla, and D. Stefanovic, "FaceTime - Deep learning based face recognition attendance system," *SISY 2017 - IEEE 15th Int. Symp. Intell. Syst. Informatics, Proc.*, pp. 53–57, 2017.
- S.Milan, H.Vaclav, and B.Roger, *Image Processing, Analysis, and Machine Vision*, 4th ed. Cengage Learning, 2014

2. Overall Description

2.1 Product Perspective

*F*Pay As You Go is a replacement for an existing system.

The current system involves payments for shuttle services through cash or UPI.

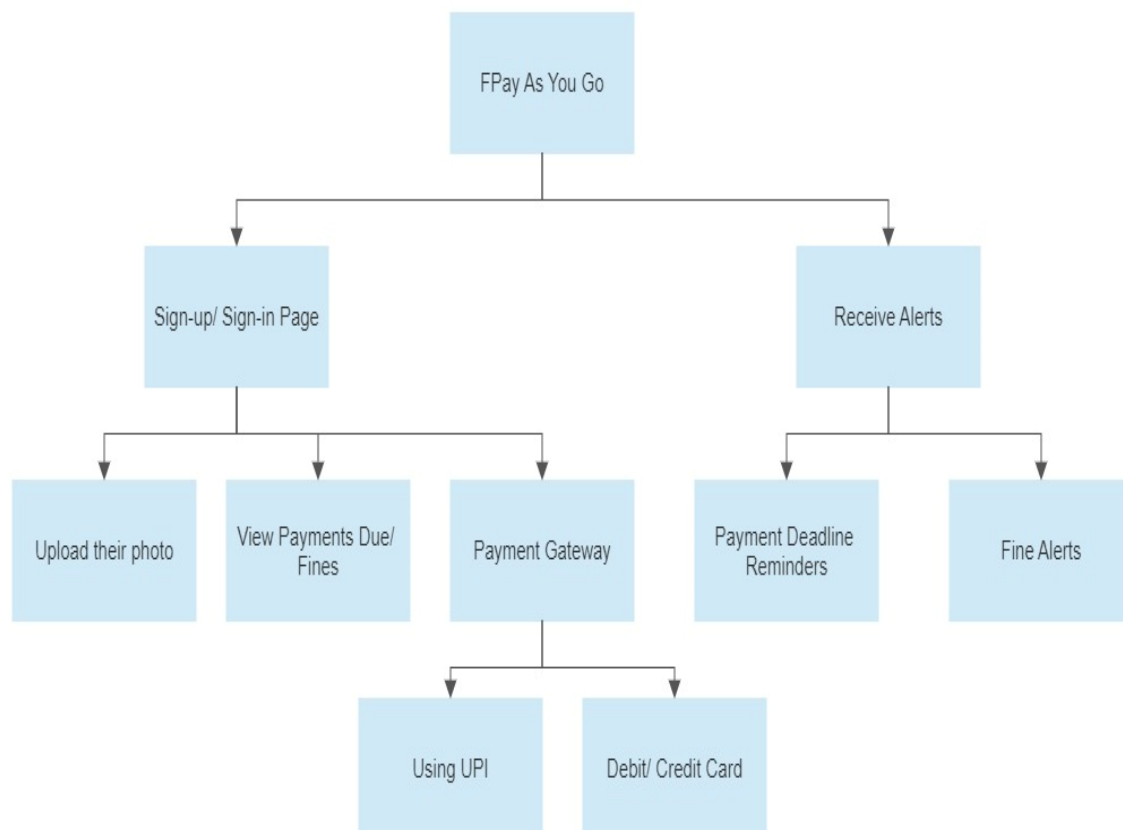
Using cash for an amount as small as 15 rupees or waiting for the mobile data to work to make use of UPI, are both time consuming and inefficient processes.

These are also not a very secure method as they do not insure that every passenger has paid for their ride.

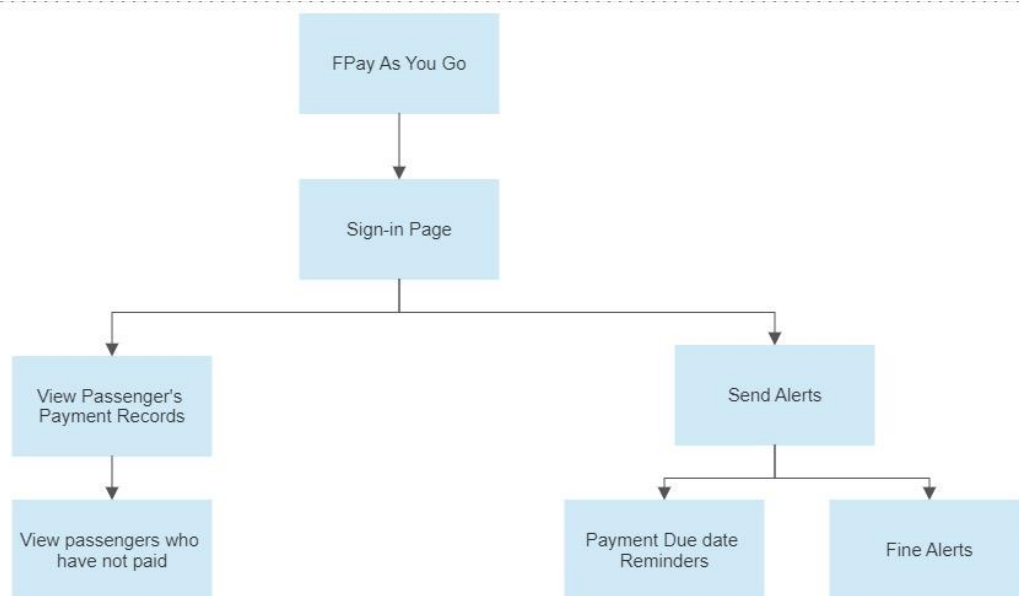
So, this project offers an easy and effective facial recognition-based payment method to make sure that everyone using the shuttle pays and that there are no losses for the shuttle services.

User-Journey (Web-App Flow Diagram):

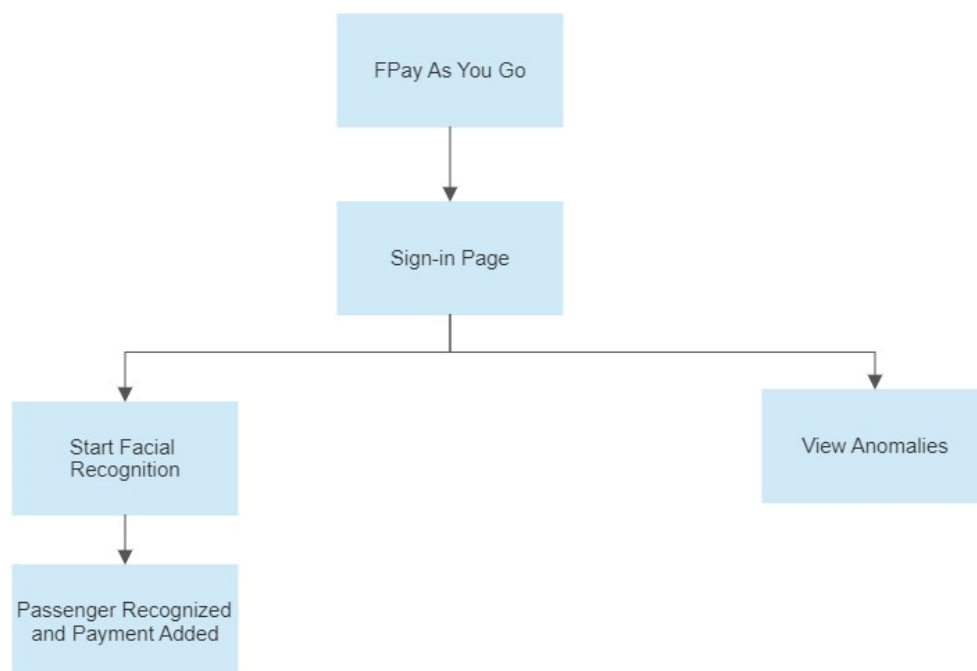
For Passenger:



For Admin



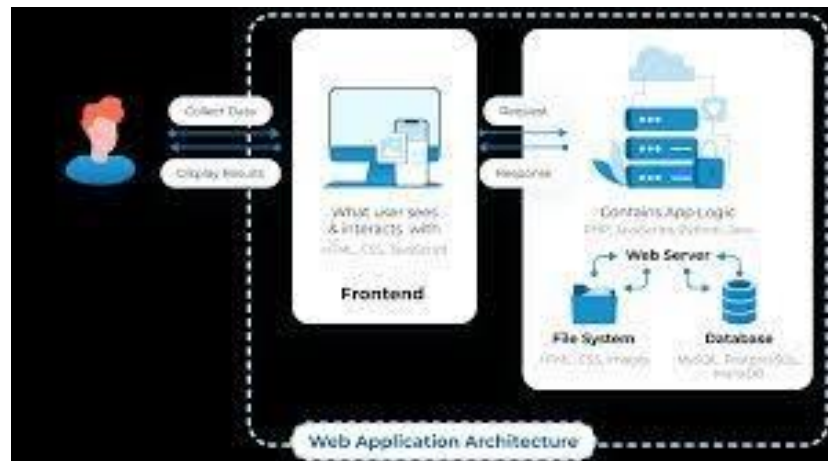
For Driver



2.2 Product Functions

- *Login*
 - *Sign-up*
 - *Sign-in*
 - *Forgot Password*

- **Facial Recognition**
 - Passengers Upload Image which are stored in Database.
 - Driver Starts Recognition Process
 - DL Model recognizes Passenger.
 - Payment is added to passenger Login.
- **Payment**
 - Passengers can view Payment Due
 - Make Payment using Payment Gateway
 - Students receive due/fine alerts.
- **Report Generation**
 - Admin Views Payment Records



2.3 User Classes and Characteristics

User Classes of our project are differentiated on the basis of subset of product functions used. The classes are as follows:

- **Passenger**
 - Upload Image
 - View Payment Due
 - Pay Using Payment Gateway
 - Get Alerts
 - Reminder for Payment Due Date
 - Overdue/Fine Alert
- **Driver**
 - Start Facial Recognition
 - View Anomalies
- **Admin**
 - View Passenger Payment Records
 - Send Alerts

2.4 Operating Environment

- **H/W Platform**
 - Core i3, i5, i7 processor
 - 2GB Ram.
 - 2GB of hard disk space in terminal machines
 - 256GB hard disk space in Server Machine
 - a cost-efficient high-quality camera like Wyze Cam v3

- Wired Security Camera and an inexpensive display device.
- **S/W Requirement**
 - Windows 7 or above operating system, macOS
 - IDEs (like VS Code)
 - MongoDB Compass
 - Dlib, OpenCV library

2.5 Design and Implementation Constraints

- **Synchronization:** uses USB 2.0, connects only to Windows 7 and above, macOS
- **Memory:** device will have 2GB internal hard drive. Device will have a SD card slot, and the software must be able to read and write to that slot.
- **Technology Tools:** VS Code, SSMS, Bootstrap Studio.
- **Language Requirements:** Python – OpenCV & Dlib, HTML, CSS, JavaScript, SQL.

2.6 User Documentation

Guidelines for Passengers

- *Upon visiting the portal, the passengers can login using their valid credentials*
- *If they forgot their password, make use of the forgot password option*
- *If not registered, the user can register using the sign-up option*
- *Then they are supposed to upload their own image, which will be used to train the model.*
- *While taking the image, ensure that there is proper lighting and that the face is clearly visible*
- *Using the view payments option, the user can view the due amount*
- *Then making use of the payment gateway, they can pay the total amount*
- *They can pay through credit/debit card or UPI.*
- *The amount should be paid within the set due date, or else they would be liable for fine/s*
- *Make sure to provide the correct email, in order to receive the reminders and alerts*

Guidelines for Drivers

- *Make use of the assigned credentials to access the system*
- *Start the recognition process on time, i.e. before the passengers start boarding*
- *During the boarding process, keep a track of anomalies and ask them to pay beforehand*
- *Don't leave the system running unattended*

Guidelines for Admin

- *Keep a track of the payment records*
- *Send alerts/reminders on time*
- *Ensure there are no inconsistencies in the database*

2.7 Assumptions and Dependencies

- Each user must have valid credentials.
- Server must be running for the system to function.
- Users must log in to the system to access any feature.
- Passenger is uploading their own photo.
- Facial Recognition System is quick and accurate.

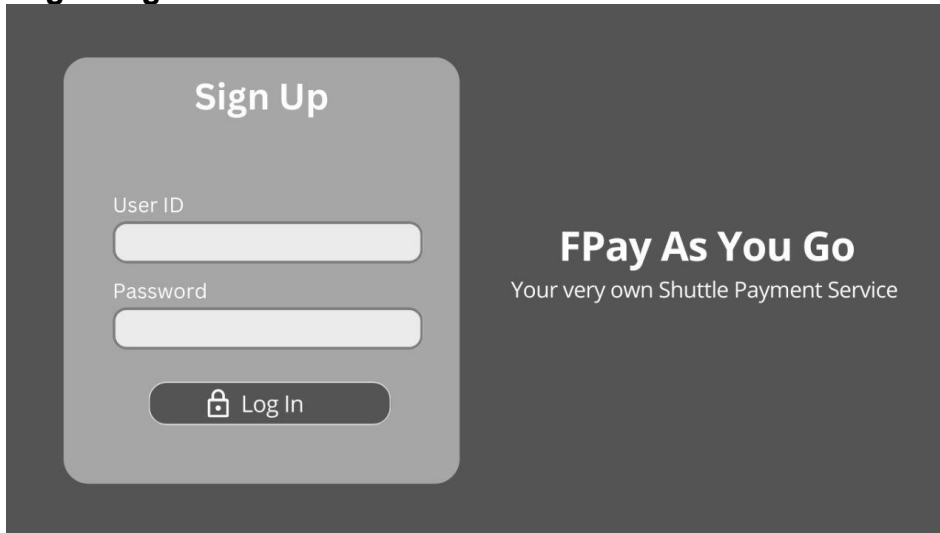
- Payment Gateway Services is always active.

3. External Interface Requirements

3.1 User Interfaces

A few sample snippets of our software's UI are as follows:

Login Page:



Passenger Dashboard Interface:



3.2 Hardware Interfaces

- **Laptop/Desktop PC/Mobile/Tablets**-To access the website and its features.

3.3 Software Interfaces

Following are the software interfaces:

- Databases are managed using MongoDB Compass.
- Operating System like Windows 7 or higher, MacOS, Linux, etc.
- Tools like Visual Studio Code to implement python code.
- Libraries like Pandas, Numpy, DLib, OpenCV, etc.
- Integrated Commercial component is payment gateway (example – RazorPay).

Data items or messages are as follows:

- Images: Take as input to train and test the facial recognition model.
- Passenger Identity: Output from the facial recognition system.
- Payment: Shuttle ride payments paid by the passenger.
- Alerts/ Reminders: Received by the passengers.

3.4 Communications Interfaces

- **NIC (Network Interface Card)** – It is a computer hardware component that allows a computer to connect to a network. NICs may be used for both wired and wireless connections.
- **TCP/IP protocol**- Internet service provider to access and share information over the Internet
- **Ethernet Communications Interface**- Ethernet is a frame-based computer network technology for local area networks (LANs)

4. System Features

4.1 Login

4.1.1 Description and Priority

- **Description:** A user-friendly platform for users (passengers, drivers and admin) to sign up/ sign in. Authentication and forgot password features are also provided.
- **Priority:** High
- **Rating**
 - **Risk:** 4
 - **Cost:** 2
 - **Benefit:** 6

4.1.2 Stimulus/Response Sequences

Visit portal ->visit login page -> sign up (for new users) -> login (for existing users)

4.1.3 Functional Requirements

REQ-1: User should input valid credentials.
REQ-2: Field Validations

Anticipated Errors:

- Forgot Password Feature

4.2 Facial Recognition

4.2.1 Description and Priority

- **Description:** It is decomposed it Facial Recognition model training, Process Input Picture and Passenger Identification. Students will login and upload their images to train the facial recognition model. The driver can start the facial recognition system when passengers are boarding i.e., the camera will take pictures of passengers and facial recognition system will identify the passenger using Passenger Details datastore and the payment will be reflected in the identified passenger's login account.
- **Priority:** High
- **Rating**
 - **Risk:** 8
 - **Cost:** 7
 - **Benefit:** 8

4.2.2 Stimulus/Response Sequences

For Student

Visit portal -> login to their account -> Upload Image

For Driver

Visit portal -> login to their account -> Start Facial Recognition-> View Anomalies

4.2.3 Functional Requirements

REQ-1: Correct Input Image

REQ-2: Timely Updates on Server.

REQ-3: Camera is functioning.

Anticipated Errors:

- Facial Recognition Failed – View anomalies feature.
- Incorrect Input Image

4.3 Payment System

4.3.1 Description and Priority

- **Description:** It is decomposed into Make Payment and View payment Summary processes with Payment confirmation as its output. Once the payment is added, the passengers can login to their accounts and view the payments due. The passengers can pay using the payment gateway and subsequently, their payment will be updated in the Payment details datastore, and they will receive payment confirmation.
- **Priority:** High
- **Rating**
 - **Risk:** 7
 - **Cost:** 3
 - **Benefit:** 8

4.3.2 Stimulus/Response Sequences

For Admin

Visit portal -> visit login page -> View Passenger Payment Records-> Send Alerts

For Passenger

Visit portal -> visit login page -> Check for due payments-> Visit the Payment Page -> Use Payment Gateway-> Receive Confirmation

4.3.3 Functional Requirements

REQ-1: Payment Gateway should be active.

REQ-2: Alerts should be sent on time.

Anticipated Errors:

- Payment Gateway Server Down

4.4 Notes

4.4.1 Description and Priority

- **Description:** It is decomposed into Generate Reports and View Reports processes. Reports are generated because of this process. The passengers can view payment receipts and relevant reports on their payments and drivers can view reports on who have paid and who haven't.
- **Priority:** Low
- **Rating**
 - **Risk:** 1
 - **Cost:** 2
 - **Benefit:** 5

4.4.2 Stimulus/Response Sequences

Visit portal -> visit login page -> Check for due payments-> Visit the Payment Page -> Use Payment Gateway-> Receive Confirmation

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- **Response Time:** The system is able to recognize the passenger within a few seconds.
- **Capacity:** The system can support multiple simultaneous logins at once.
- **User-Interface:** The user interface acknowledges within a few seconds.

5.2 Safety Requirements

Due to a virus or OS error, the database could crash at any given time. As a result, taking a database backup is necessary to prevent database loss. In the event of a power supply breakdown, suitable UPS/inverter facilities should be available.

Safety Certifications

- *ISO/IEC 12207:2008*

5.3 Security Requirements

- **User Identification:** The system needs the passenger to recognize herself or himself using valid credentials.
- **Privacy:** Data Leak can result in privacy violation.

Security and Privacy Certifications:

- ISO/IEC 27001:2013

5.4 Software Quality Attributes

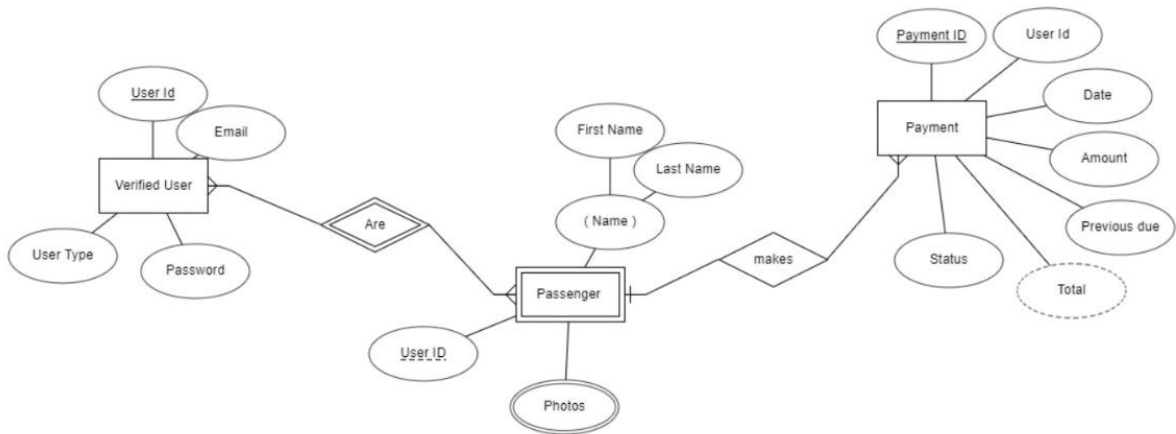
- **Usability:** The system can be used again and again without any malfunctions or distortion.
- **Availability:** The system shall be always available for use.
- **Correctness:** The system will be free of bugs and errors to fulfill the desired requirements of the users.
- **Maintainability:** The system will be maintained and updated with new features regularly to incorporate new user needs.
- **Portability:** The software is operable on multiple OS.
- **Testability:** The software will be tested using various testing techniques (manual, automated, etc) and test cases.

Appendix A: Glossary

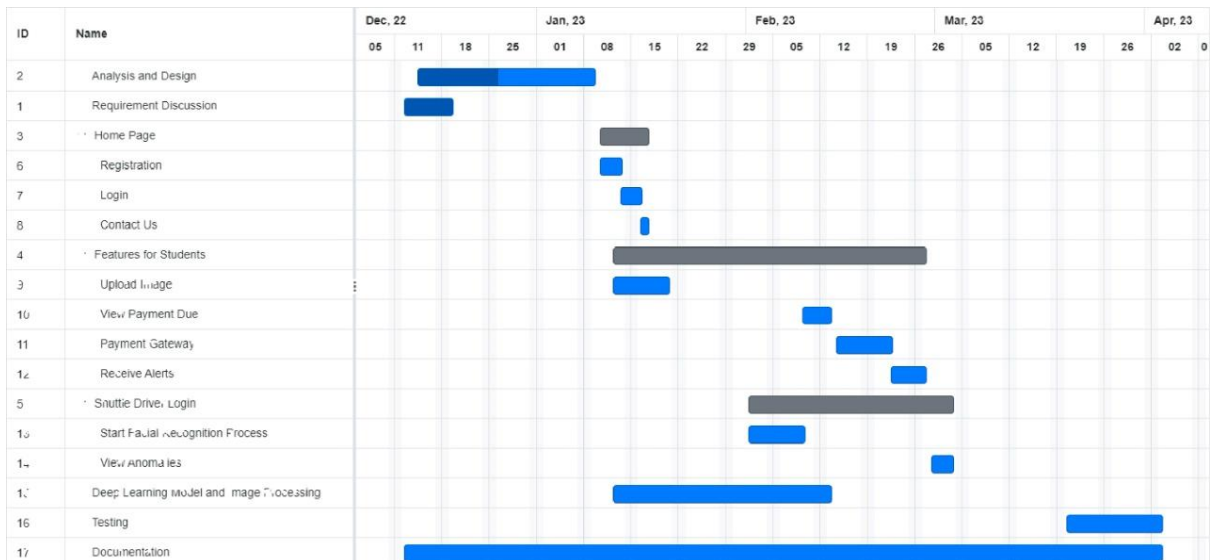
Acronyms	Abbreviations
SSMS	SQL Server Management Studio
JS	JavaScript
WBS	Work Breakdown Structure
Req	Requirements
GB	Giga Byte
SRS	Software Requirement Specifications
IDE	Integrated Development Environment
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
OS	Operating System
ISO	International Organization for Standardization
TCP	Transmission Control Protocol
IP	Internet Protocol
OPT	Optional

Appendix B: Analysis Models

Entity Relation Diagram:

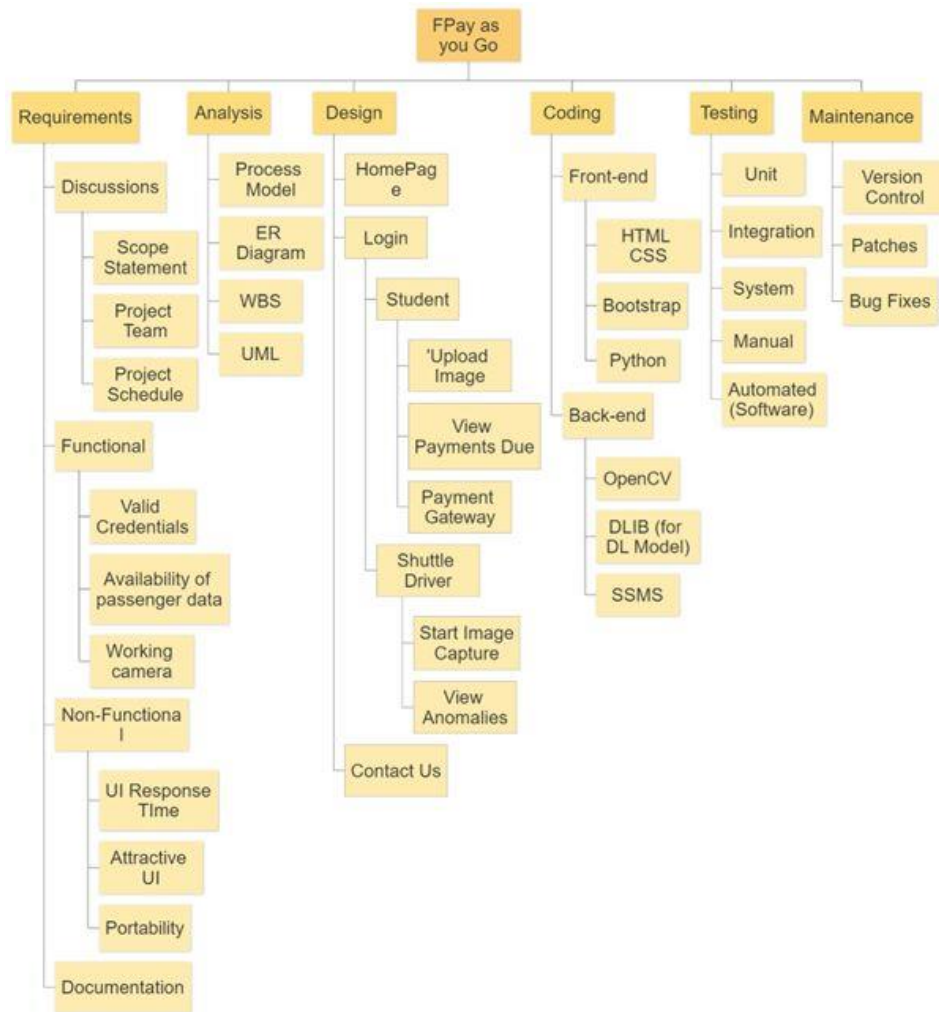


Project Scheduling Gantt Chart:

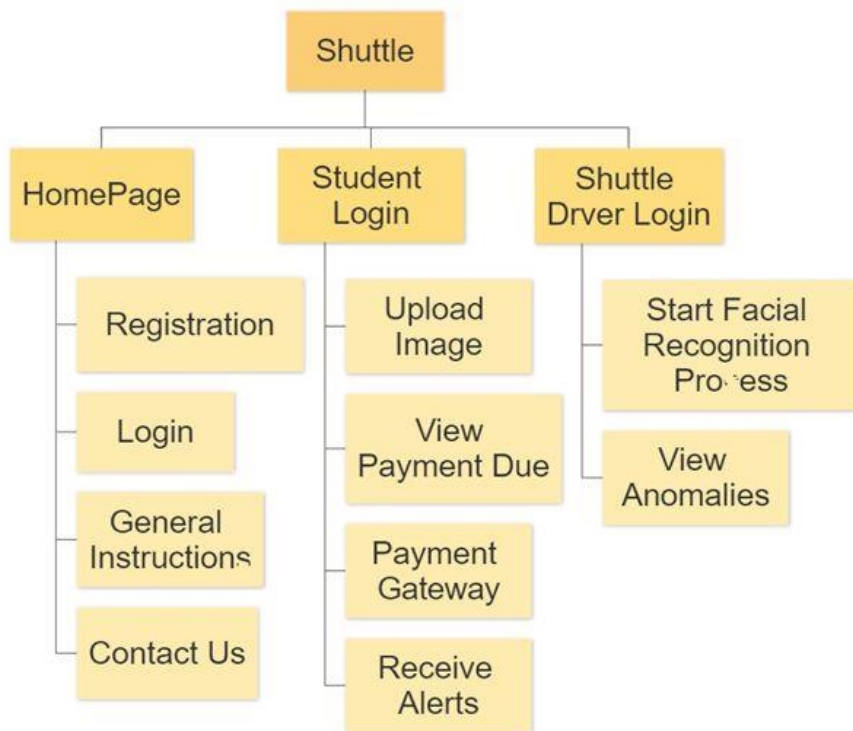


Work Breakdown Structure: -

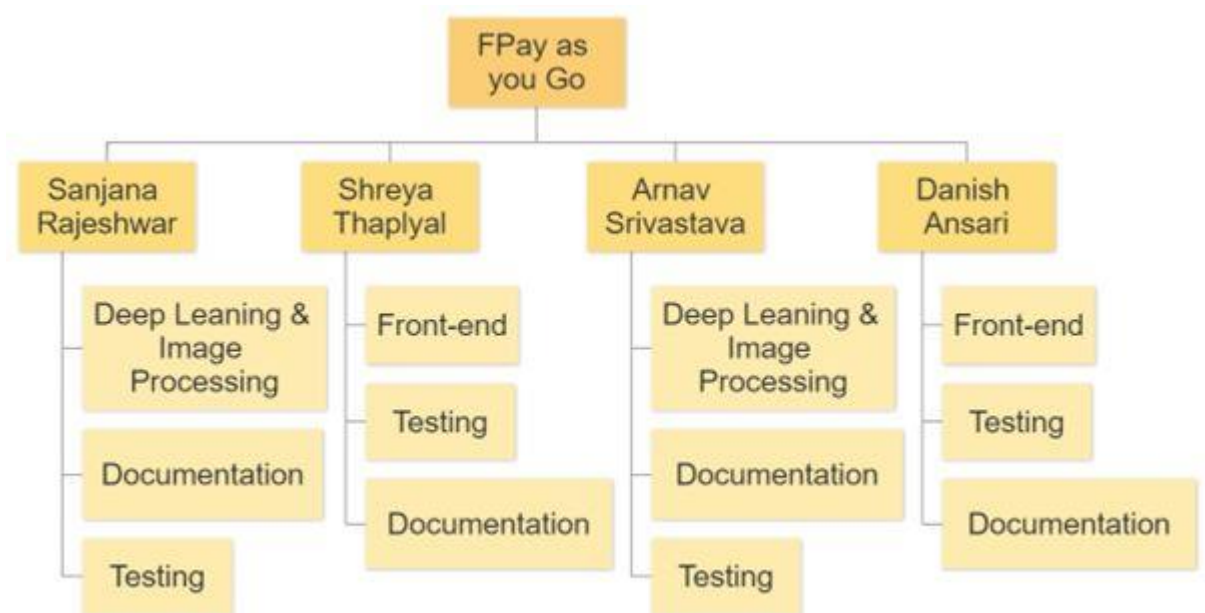
- **Phase:**



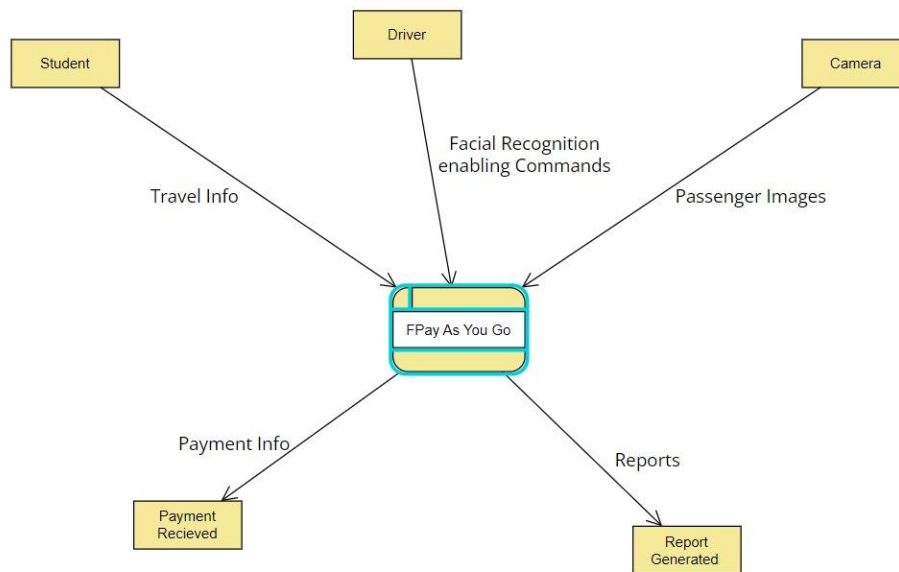
- **Deliverable:**



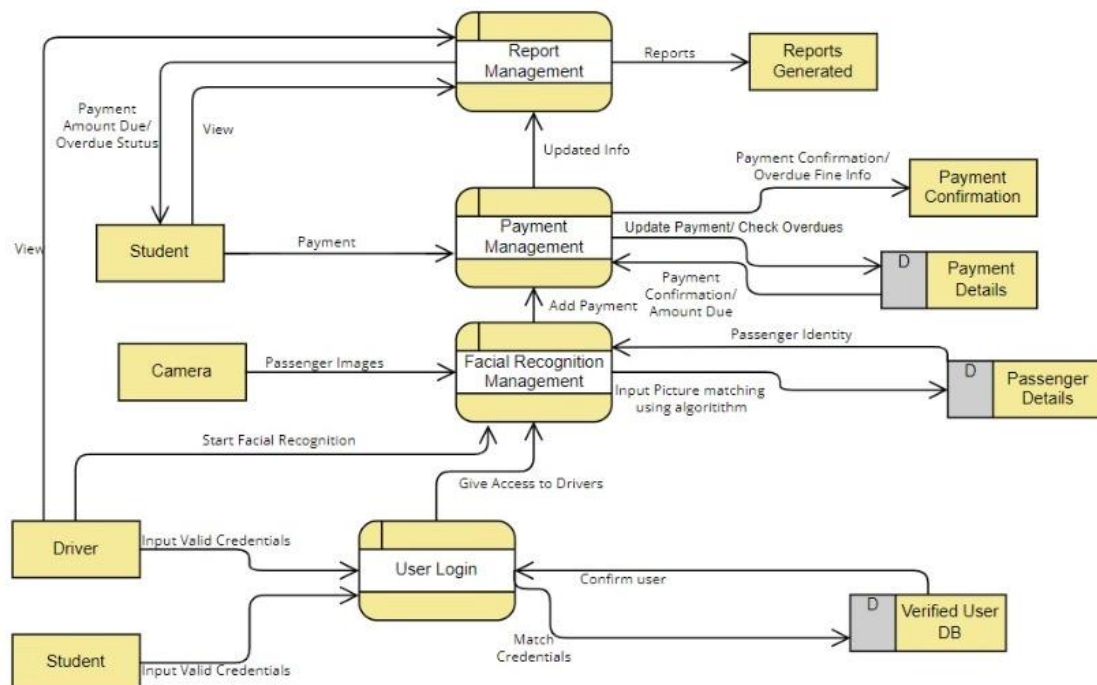
- **Responsibility:**



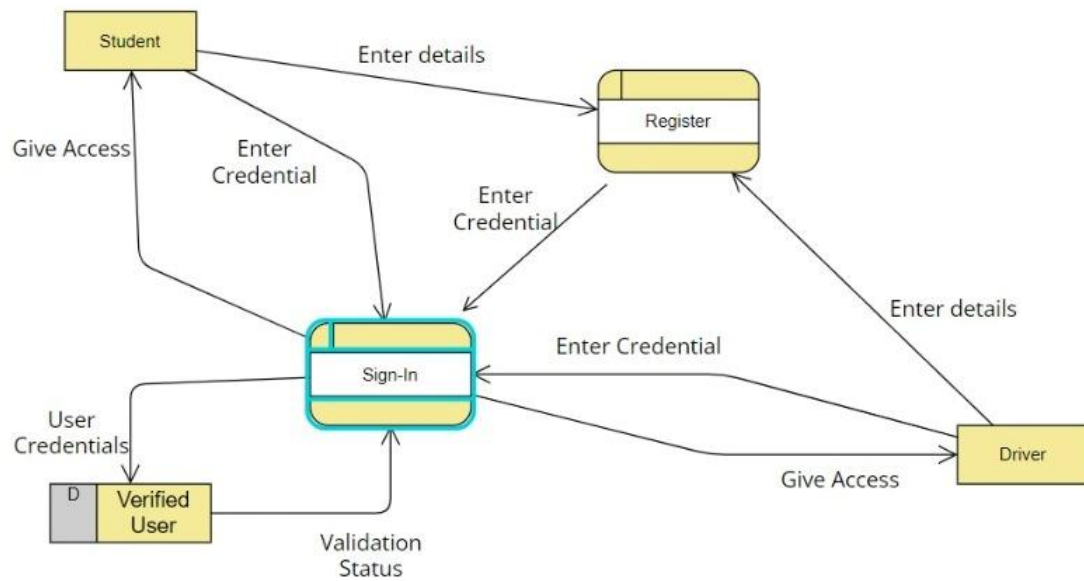
Context Level Diagram:



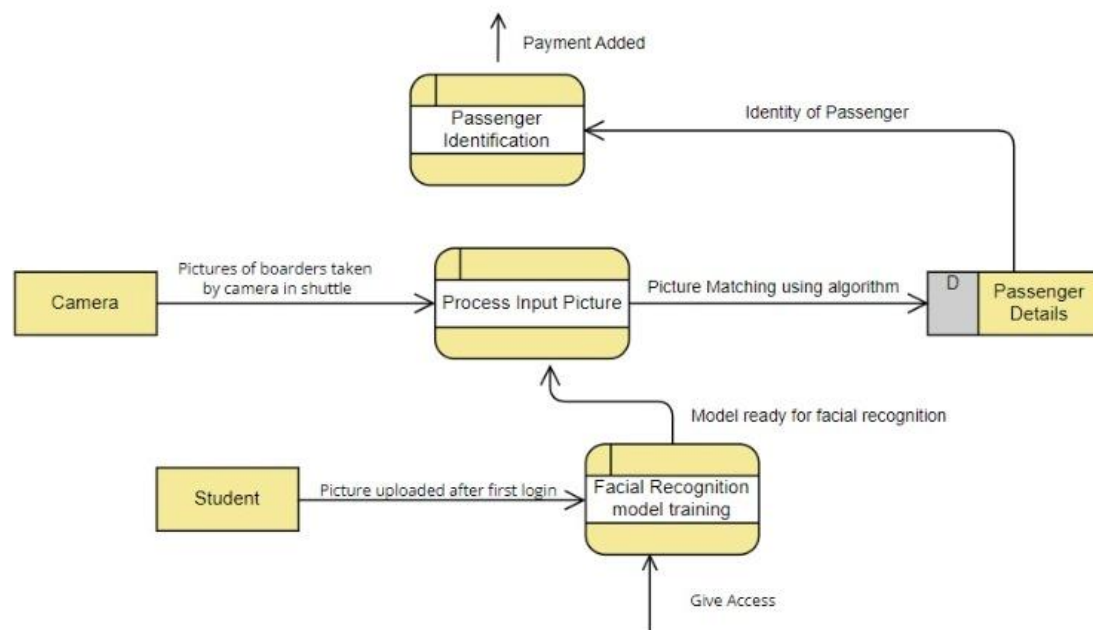
Data Flow Diagram Level 0:



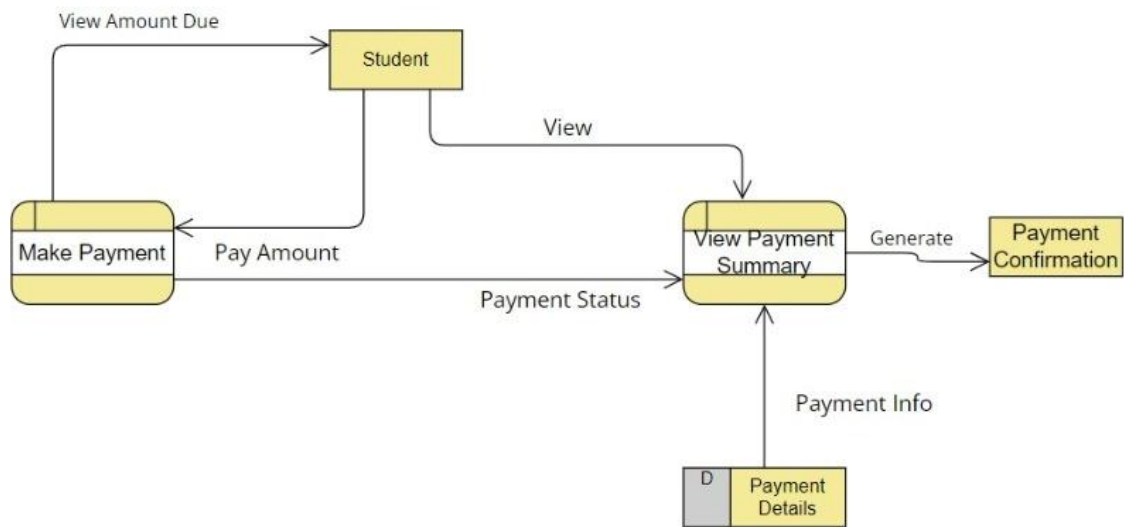
DFD Level 1: Login



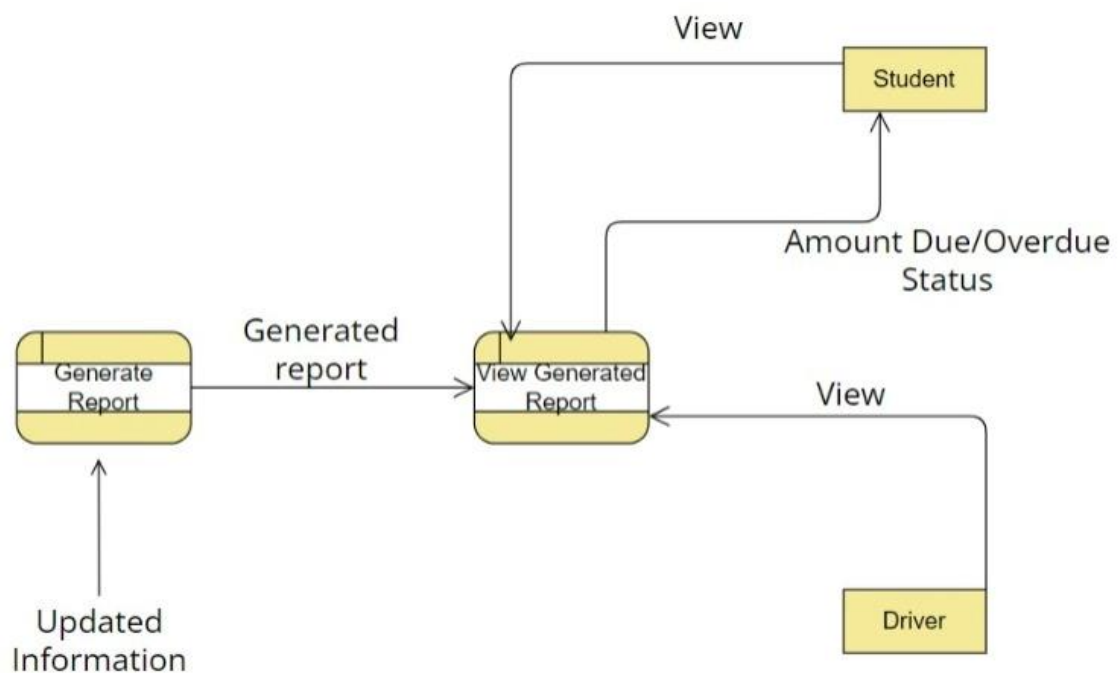
Facial Recognition Management



Payment System:



Reports Management System:



Use-Case Diagram:

Model1::UseCaseDiagram1

