



**National University of Sciences & Technology (NUST)**

Military College of Signals (MCS)

**FINAL YEAR PROJECT**

**THESIS**

FOR

**VANGO - ALERT BASED TRANSPORTATION**

Version 2

**Team Members:**

Shazray Shah

Hadia Shafqat

Hareem Ahmed

Javeria Shahbaz

**Supervisor:**

Dr Shahzaib Tahir

**Date:**

11-11-2024

**Table of Contents:**

National University of Sciences & Technology (NUST)..... 1

**1. Introduction..... 5**

1.1 Purpose (Problem Statement)..... 6

-Safety and Security:..... 7

-Real-time Monitoring:..... 7

-Attendance Management:..... 7

-Seamless Communication:..... 7

-Convenience and Peace of Mind:..... 7

1.2 Document Conventions..... 8

-Indexing:..... 8

-Fonts:..... 8

-Structure:..... 8

-Priorities:..... 8

1.3 Intended Audience and Reading Suggestions..... 9

Intended Audience..... 9

Document contents and organization..... 9

1.4 Product Scope..... 11

1.5 References..... 11

**2. Overall Description..... 11**

2.1 Product Perspective..... 11

2.2 Product Functions..... 12

2.3 User Classes and Characteristics..... 13

2.3.1 Parents..... 13

2.3.2 Admin..... 13

2.3.3 Child..... 14

2.3.4 Drivers..... 14

2.4 Operating Environment (Technological Choice)..... 15

2.4.1 Hardware Platform:..... 15

2.4.2 Software Components:..... 15

2.4.3 Additional Software Components..... 15

2.4.4 Operating System:..... 16

2.4.5 Other supporting hardwares..... 16

- 2.5 Design and Implementation Constraints..... 16
- 2.6 User Documentation..... 17
- 2.7 Assumptions and Dependencies..... 17
  - 2.7.1 Assumptions about Commercial and External components :..... 17
  - 2.7.2 Dependency on External components:..... 17
  - 2.7.3 Assumptions about Clients devices:..... 17
  - 2.7.4 Assumptions about Internet connection:..... 17
- 3. External Interface Requirements..... 18**
  - 3.1 User Interfaces..... 18
    - User Interface Components :..... 18
    - GUI Standards:..... 19
  - 3.2 Hardware Interfaces..... 20
  - 3.3 Software Interfaces..... 21
  - 3.4 Communications Interfaces..... 22
- 4. System Features..... 22**
  - 4.1 Parent’s Registration..... 22
    - 4.1.1 Description and Priority..... 24
    - 4.1.2 Stimulus/Response Sequences..... 24
    - 4.1.3 Functional Requirements..... 24
  - 4.2 Child Registration..... 24
    - 4.2.1 Description and Priority..... 26
    - 4.2.2 Stimulus/Response Sequences..... 26
    - 4.2.3 Functional Requirements..... 26
  - 4.3 Facial Recognition..... 26
    - 4.3.1 Description and Priority..... 27
    - 4.3.2 Stimulus/Response Sequences..... 28
    - 4.3.3 Functional Requirements..... 28
  - 4.4 Location Tracking..... 28
    - 4.4.1 Description and Priority..... 29
    - 4.4.2 Stimulus/Response Sequences..... 30
    - 4.4.3 Functional Requirements..... 30
  - 4.5 Push Notifications and Alerts..... 30
    - 4.5.1 Description and Priority..... 31
    - 4.5.2 Stimulus/Response Sequences..... 32
    - 4.5.3 Functional Requirements..... 33
  - 4.6 Live Camera Feed for Parents to View Children's Journey..... 33

- 4.6.1 Description and Priority.....34
- 4.6.2 Stimulus/Response Sequences.....34
- 4.6.3 Functional Requirements.....35
- 4.7 Payment (Receipt Generation and Verification).....35
  - 4.7.1 Description and Priority.....37
  - 4.7.2 Stimulus/Response Sequences.....37
  - 4.7.3 Functional Requirements.....38
- 5. Elaborated Functional Requirements.....38**
  - 5.1 Parent Enrollment:.....38
  - 5.2 Registration of Children:.....38
  - 5.3 Recognizing faces:.....39
  - 5.4 Tracking and Storage of Child's data.....39
  - 5.4 Monitoring Location:.....39
  - 5.5 Push Alerts & Notifications:.....39
  - 5.6 Live Camera Feed for Parents to View Children's Journey:.....39
  - 5.7 Payment (Generation and Verification of Receipts):.....39
- 6. Nonfunctional Requirements.....39**
  - 6.1 Performance Requirements.....39
    - 1) Responsiveness.....40
    - 2) Managing high traffic.....40
    - 3) Loading Time.....40
    - 4) Battery Efficiency.....40
  - 6.2 Safety Requirements.....40
  - 6.3 Security Requirements.....41
  - 6.4 Software Quality Attributes.....41
  - 6.5 Business Rules.....42
- 7. Other Requirements.....43**
  - 7.1 Scalability and performance:.....43
  - 7.2 Usability:.....43
- 8. Innovative Feature Distinction.....44**
  - 8.1 Real Time Tracking and Monitoring.....44
  - 8.2 Attendance Marking with Time:.....45
  - 8.3 Live Feed with Video Archive:.....45
- 9. Contribution to United Nations Sustainable Development Goals (SDGs).....45**
  - SDG 3: Good Health and Well-being:.....46
  - SDG 4: High-quality Education.....46

SDG 9: Industry, Innovation, and Infrastructure:.....	46
SDG 13: Climate Action:.....	46
SDG 17: Partnership for the Goals.....	47
<b>10. Appendixes.....</b>	<b>47</b>
10.1 Appendix A:.....	47
Glossary:.....	47
Project Timeline:.....	47
Project Proposal:.....	48
10.2 Appendix B: Diagrams.....	48
Use Case.....	48
System Architecture Diagram.....	50

### Revision History

Name	Date	Reason For Changes	Version
SRS VANGO	July 15	The first SRS document	1.0
SRS VANGO VER2	Nov 6	Updates To Some Features Change of Software	2.0

# 1. Introduction

This document provides an introduction to VanGo, the Android application for a School Van transportation System. The primary objective of this app is to offer an efficient and reliable van service for school-going children. Through a user-friendly mobile solution, we aim to simplify the process of finding and booking a safe van that provides real-time tracking of school vans, delivers live camera feed from inside the van and gives attendance notifications to parents, while also storing all pertinent data for future reference.

This document will outline the project's scope and define the key terminologies that will be referenced throughout. Furthermore, it provides an overview of the application's target users and discusses its high level behavior and functionalities. It also addresses assumptions related to the working environment to ensure seamless operation of the system.

Furthermore, this document explores the specific low-level software requirements for each functional aspect of the system. This document contains the detailed requirements of the application explicitly, with the aim to provide a comprehensive understanding of all its capabilities. Additionally, a high-level system entity model and associated use cases will be provided to offer a visual representation of the system's behavior and functionalities.

To facilitate the development process, explicit specifications of each system feature and usage instructions will be included. They are crucial for ensuring a successful implementation of the application, and for providing a clear understanding of its intended functionality.

Lastly, this document will conclude by providing a list of references and specifying a dedicated point of contact for any additional information or queries related to the project. Through the presentation of this comprehensive document, we aim to establish a solid foundation for the development of VanGO: Alert Based Transportation System mobile application. By providing a clear understanding of its purpose, functionalities, and intended user experience, we aim to create a reliable and convenient transportation solution for parents and school-going children alike.

## **1.1 Purpose (Problem Statement)**

In Pakistan, the conventional school van system for transporting children to school has its limitations and fails to adequately address parental concerns regarding child security. Moreover, the process of finding and booking a van can be quite challenging. With the development of the VanGO mobile application, we strive to digitize the process of booking reliable school vans and ensuring the safety and security of children.

The purpose of the VanGO app is to provide a convenient, user friendly and efficient school van service for school-going children that is able to address the security concerns of parents in the current law and order situation in Pakistan. This mobile application aims to streamline the van search and registration process, provide real-time tracking of the school van, and offer live camera feed of the happenings inside the van. Additionally, it aims to deliver real-time attendance notifications to parents, and store relevant data for future reference.

### **-Safety and Security:**

VanGO prioritizes the safety and security of children during their van travel. To implement that, our app offers real-time updates and live tracking of the school van,

allowing parents to have visibility into their child's whereabouts at all times. The use of face recognition technology ensures accurate attendance tracking, providing an additional layer of security.

### **-Real-time Monitoring:**

The app enables parents to monitor their child's journey in real-time by opting to view the live camera feed from inside the van. They can view the live location of the school van on a map at all times, ensuring peace of mind and reducing concerns about their child's safety during the commute.

### **-Attendance Management:**

The app automates the attendance process by utilizing face recognition technology. It eliminates the need for manual attendance tracking and parents receive instant notifications when their child boards or exits the school van, along with the corresponding location information.

### **-Seamless Communication:**

The app facilitates seamless communication from the school van to the parents. Through push notifications, the app issues alerts and parents receive important updates like when their child enters and exits the van, the location of the van, and other things such as changes, delays, or emergencies. Notifications regarding fee slips, timely payment or warnings etc. are also sent, ensuring an effective and streamlined communication channel.



### **-Convenience and Peace of Mind:**

By providing a user-friendly interface and intuitive features, the app offers convenience to parents. They can find and book vans, register their children, monitor their commute, access attendance records, review past journeys, receive important notification alerts, all in one place. This convenience allows parents to stay informed, feel connected, and have peace of mind about their child's school transportation experience.

## **1.2 Document Conventions**

For the development of this document, the established standards and typographical conventions were followed to ensure a polished presentation:

### **-Indexing:**

- The document includes an index section that lists the major headings and subheadings along with their corresponding page numbers.
- The index section helps readers quickly locate specific sections or subsections within the document.

### **-Fonts:**

- The document follows the font Times New Roman for normal text, which is a common font style with good readability
- Different font styles, such as bold or italic, were selectively used for emphasis in appropriate places, headings, or subheadings.

**-Structure:**

- The document followed a structured format with clearly labeled sections and subsections.
- This organizational structure facilitated a logical flow and easy navigation through the document.

**-Priorities:**

- Each requirement statement was thoughtfully assigned a priority level (such as High, Medium, or Low), indicating its relative importance compared to other requirements.
- These priority levels establish a clear hierarchy of importance among the requirements in the document, which facilitates the development process.

### **1.3 Intended Audience and Reading Suggestions**

#### **Intended Audience**

The VanGO Android application is being built as a Final Year Undergraduate University Project. So the SRS document for the app is intended for various readers and stakeholders that are directly involved in the project, including the developing team, the evaluating panel, our supervisors, some users and testers, and all the team members involved in the project.

#### **Document contents and organization**

This comprehensive Requirements Specification document provides a detailed description of the specific topics outlined in the index section. It begins by presenting an overview of the app's scope, defining system boundaries, and enumerating all functionalities and features within that scope. The document explicitly lists all the

references, including external sources, standards, and frameworks referenced throughout the SRS document. The suggested reading sequence is in ascending order from the index, or the readers can directly jump to their relevant sections by locating them through the index.

The Overall Description section offers a holistic view of the system, going beyond a superficial overview. It explains the application perspective by illustrating how the system interacts with other components and systems in a broader context. Additionally, it highlights the app's capabilities, outlining its key features and functionalities. The section also addresses the needs of different user classes by providing a comprehensive list of user classes and their unique characteristics, ensuring that the system caters to their specific requirements. Furthermore, it explores the operating environment, taking into account hardware, software, and network requirements, and considers design constraints. The significance of user documentation is emphasized, specifying the necessary resources such as user manuals and help guides to aid users in effectively navigating the system. Lastly, the section acknowledges the assumptions made during development and identifies any external dependencies that might impact the system. Since this section gives a comprehensive overview of the system, it's relevant to all readers.

The External Interface Requirements section elaborates on the hardware, software, and user interfaces of the app. It describes the interfaces through which users interact with the system, including details about hardware components like cameras and GPS devices that are necessary for system functionality. The section also outlines software interfaces such as external system APIs integrated into the app and communication protocols and interfaces used for data exchange with external entities like servers or databases. This section is most relevant to developers, testers, and system integrators.

Next, the System Features section provides an exhaustive list and description of the specific functionalities and capabilities of the VanGO application. Each key feature is

explained in its own subsection, highlighting its purpose and how it interacts with other system components. Since this section provides a detailed list and description of the specific functionalities and capabilities of the VanGO app, it is relevant to developers, project managers, and testers

Next, the document's section on functional requirements describes the precise features, behaviors, and capabilities that the software system needs in order to achieve the goals that have been expressed. It outlines the interactions, inputs, outputs, and anticipated behavior of the system under various conditions, so defining how it should operate. A thorough software development roadmap is given in this section, which also acts as a basis for work on design, implementation, and testing. It guarantees that the program satisfies user and stakeholder needs while adhering to the specified functionality.

The Nonfunctional Requirements section covers various aspects, including system performance, safety considerations for child transportation, security requirements for data protection, desired software quality attributes, and specific business rules that the system must adhere to. Since it covers all of this, this section is most relevant to developers, testers, and system administrators.

Appendix A, the Glossary, serves as a reference, defining key terms used throughout the SRS document to ensure a shared understanding of terminology. Appendix B, the Analysis Models, offers visual representations and diagrams illustrating the system's architecture, data flow, and component interactions, aiding comprehension of the system's design. This section is relevant to all stakeholders and readers.

## **1.4 Product Scope**

The app's product scope encompasses providing an all in one platform for parents to register their children for van transportation services and ensuring their safety throughout the journey. The app enables parents to create their profiles and register their children, inputting necessary information.

It incorporates a live feed feature, granting parents real-time access to video footage and audio from the van's camera, allowing them to monitor their child's journey. The app also includes location tracking functionality, enabling parents to view the van's real-time location on a map. Moreover, attendance tracking through facial recognition is implemented, ensuring efficient monitoring of when the child boards and leaves the van, with instant notifications sent to parents. The app also facilitates the storage of recorded footage, enabling parents to access and review it later if needed.

## **1.5 References**

Kyle Stratis (Apr 24, 2023). Accessed on: Aug 1, 2023. [Online]. Available:

<https://realpython.com/face-recognition-with-python/>

# **2. Overall Description**

## **2.1 Product Perspective**

This application is a new, self-contained application designed to cater to the transportation needs of school-going children. It is not a follow-on member of an existing product family nor a replacement for any specific system. The app serves as a standalone

solution, providing van transportation services with enhanced security and parental involvement.

The functionality of this software is focused on streamlining the process of registering children, ensuring their safety during transportation, and facilitating parent-child communication. The app operates within the larger context of a transportation management system, where it serves as a critical component responsible for managing and monitoring the transportation of children.

The app's functionality interfaces with the larger system by exchanging data with the transportation management system, ensuring seamless information flow between different components. This integration allows for the synchronization of data related to registered children, attendance records, and real-time bus locations. By integrating with the larger system, the app enhances the overall transportation management capabilities, providing a comprehensive solution for school transportation services.

## **2.2 Product Functions**

1. Parents can register their children on the app. First the parents register themselves (if not already registered). A confirmation email is sent to them after verification of which, the parent is registered. Then he registers the child. First he enters all the child's information. In the next step he takes pictures of the child from three angles and uploads them. After that a fee invoice is generated and displayed to the user. The parents can pay either through online payment service or by going to the bank themselves. After confirmation of the paid receipt, the parents are granted access to all the application's features.

2. Parents can view live feeds with audio when their child is on the bus.
3. The app consists of a facial recognition based attendance system. Every time a child gets on the bus and leaves the bus, the parent is sent a notification due to facial recognition.
4. Parents can see the live location of the bus in which the child is.
5. The live feed and attendance is also stored and parents can view it later.
6. Parents can register a complaint and notifications are sent to parents.

## **2.3 User Classes and Characteristics**

### **2.3.1 Parents**

The most important user class will be of Parent/Guardian.

Following are the characteristics that differentiate it from other classes.

- Frequency of Use: Parents will use this app regularly to monitor their children when they're on the bus.
- Product Functions: Most of the app functionalities are designed for the parents, they can register their child, view live feed and location, access attendance records and register complaints.
- Technical Expertise: Parents may vary in their technical expertise. So the apps UI will be designed in a way that both tech savvy and those parents with limited knowledge about technology can use it.
- Security or Privilege Levels: Each parent will have his/her own account which will be protected and no other parent can access other child's information.
- Educational Level: The app will be user friendly so that parents with any educational background can use it.

- Experience: Parents can have different experience with technology so the User Interface won't restrict a parent with less experience and they'll easily be able to navigate through the app.

### **2.3.2 Admin**

Admin or the people responsible for maintaining the app can also be considered as user classes. This will also be an important class.

Following are the characteristics that differentiate it from other classes.

- Frequency of Use: Admins will be monitoring the app all the time but mostly during high traffic hours.
- Product Functions: The admin would be responsible for managing the app's backend. They would be able to manage a parent's account, unregister a parent or child, and handle complaints.
- Technical Expertise: Admin must be a technical expert and have knowledge about how the app is functioning, how the servers are configured so that he can resolve any issue that might occur.
- Security or Privilege Levels: Admins will have highest privileges as they'll be able to manage user accounts and view sensitive information. Admin's information will be secured using different protocols.
- Educational Level: The admins must have the required education to easily navigate through the app and perform high privilege operations.
- Experience: Admins must have a certain level of experience with technology and mobile apps to easily handle the apps backend.

### **2.3.3 Child**



The child won't directly interact with the app but the child is the one to use bus services. His record will be stored in the database along with the pictures for facial recognition. So the child doesn't need to have any experience with technology as he is not interacting with the app.

Following characteristics differentiates it from others

- Frequency of Use: The child won't use the app only the bus.
- Product Functions: No product function is for the child but for parents.
- Technical Expertise: Child doesn't need to have technical expertise.
- Security or Privilege Levels: The child's security will be insured in the bus through cameras and the child's info will be safe in the app's database.

### **2.3.4 Drivers**

Driver is not a direct user of the app. We only need the driver's phone to access the current location.

Following characteristics differentiates it from others

- Frequency of Use: Only during working hours.
- Product Functions: Drivers mobile is used to send location to server.
- Technical Expertise: The driver does not need to be a technical expert. They will have a simple interface that will only be used to send location to server.
- Security/Privilege Levels: They won't have access to sensitive information or administrative functions.
- Educational Level: The driver does not need formal education to use the app.

- Experience: The driver does not need extensive experience with technology, be should be comfortable operating a smartphone. They will use the phone's built-in GPS tracker for location tracking.

## 2.4 Operating Environment (Technological Choice)

### 2.4.1 Hardware Platform:

The app runs on android and ios devices. The specifications of the phone should be able to support the app's features. We will be using an IP camera for live streaming and face recognition.

### 2.4.2 Software Components:

Virtual Studio Code(VS code) will be the IDE used for development of our app.

The languages used for the backend server will be Python.

The database used will be SQL Postgres.

The language used for the front end will be React Native, which is a cross platform language.

### 2.4.3 Additional Software Components

**Firebase:** Used for User Authentication and Push Notifications

**Flask:** Framework used to implement the server

All other modules shall be developed from scratch using existing Python libraries like facial\_recognition, open\_cv and others.

#### **2.4.4 Operating System:**

We'll be using both iOS and Android operating systems. The version of Android will be Android 5.0 and above.

#### **2.4.5 Other supporting hardware**

For the app to function some hardware will be required to be integrated with our bus. They are following:

Camera for recording live feed and for facial recognition.

Smart phones for drivers with GPS.

### **2.5 Design and Implementation Constraints**

The Constraints of our app is given below:

1. Our App may have budget limitations that restrict the choice of technologies, tools or resources available for development.
2. Budget constraints may limit the choice of server hosting options, forcing us to opt for more cost-effective or limited resource options.
3. The choice of hosting option may impact scalability, customization, and management capabilities, potentially limiting the ability to handle increasing user loads or meet specific requirements.
4. The app will have strict performance requirements, such as response times, scalability, or resource utilization. We will consider these requirements when choosing technologies, databases, or architectural approaches to ensure the app meets performance expectations.

5. The app will adhere to strict security standards and follow best practices for protecting user data, preventing unauthorized access, and ensuring secure communication channels. This may limit certain implementation options to prioritize data security and privacy.

## **2.6 User Documentation**

The VanGO Android application will be accompanied by an all encompassing set of user documentation in different forms to ensure users have the necessary resources to effectively use the software. These components include:

1. **Email Help:** Users will have the convenience of reaching out to the VanGO support team via email for detailed assistance. The support team will promptly respond to inquiries, providing personalized and informative responses to address users' specific concerns. The email help service will ensure that users receive the personalized support they need, enabling them to overcome any challenges they may encounter during their app usage journey.

## **2.7 Assumptions and Dependencies**

### **2.7.1 Assumptions about Commercial and External components :**

The app uses cameras, geolocation services, servers and libraries for facial recognition. It is assumed that these will be available for use and would be compatible with our app.

### **2.7.2 Dependency on External components:**

The app also depends on Camera, facial recognition libraries and geolocation services. If integration with these services fails the app wont work as desired.

### **2.7.3 Assumptions about Clients devices:**

It is assumed that Client will have an Android and IOS device with a working camera as during registration they'll have to take and upload childs pictures

If the camera is not working the child's registration process might not be completed.

### **2.7.4 Assumptions about Internet connection:**

It is assumed that both client and bus will have an active internet connection all the time so the live feed is transmitted to client at high quality at all times.

The app is dependent on internet connection for working properly.

## **3. External Interface Requirements**

### **3.1 User Interfaces**

For any application, a user friendly and intuitive User Interface is essential to allow users to interact with the app smoothly. It is through the UI that users access the key functionalities of the application, input information, navigate through the features, or ask for assistance. Thus it is essential to have a well designed UI to allow for a seamless user experience.

There should be properly designed screens and screen components to ensure easy access to each key functionality of the app. These screen designs should be visually appealing and designed to enhance clarity for navigation. It is important to display the data in a clear and understandable manner, where information is organized logically using appropriate fonts and layouts for readability. This will allow the user to grasp the relevant information correctly.

## **User Interface Components :**

- The login/authentication process provides a seamless and secure interface for users to access their accounts.
- The signup process also requires a button to access the functionality and then a form to allow the user to give input in a guided manner.
- The Register Child button and subsequent form guides users through the process for registering their children for the service.
- The Children Profile screen offers parent users to select a particular child to view his profile and access the service options specific to them.
- The Register New Child button and subsequent form guides users through the process of registering more of their children for the service.
- The Upload Child Pictures button allows users to upload the required pictures of their children while registering. This is also a guided process.
- The Payment Info screen displays the relevant payment status of the child and also has buttons to allow parents to download the fee slips or upload paid receipts.
- The Child's Home Page is a page that displays all the functionalities of the app available for the app. It has buttons to access each key functionality of the app.
- The View Live Location button is a van tracking feature that provides an interactive interface for users to track the real-time location of vans.
- The View Live Feed button is another van tracking feature that provides an interactive interface for users to view the real-time camera feed from inside the vans during travel.
- The View Attendance Record button allows users to view their child's attendance history.

- The View Past Recordings button allows users to view the stored recordings of their child's past van journeys.
- The View Payment Info button allows users to view their child's payment status and download fee challan or upload receipts.
- The notifications interface keeps users informed about important updates and announcements.
- The Register Complaint button and the subsequent complaint screen allows users to write a complaint regarding the service that'll reach system administrators.
- Help and support features provide users with easily accessible assistance, such as FAQs, documentation, or an email support system.

### **GUI Standards:**

- The app follows a clean and user-friendly design, with consistent colors, fonts, and visual components throughout.
- It incorporates clear icons and imagery to enhance the user experience and facilitate understanding.
- The design is responsive, ensuring that it adapts seamlessly to different screen sizes and orientations.

### **Screen Layout Constraints:**

- The app's interface is thoughtfully organized, with elements placed logically for easy navigation and access to important information.

- Adequate spacing is maintained between elements to avoid clutter and enhance readability.
- Consideration is given to readability and accessibility guidelines, such as choosing appropriate font sizes and contrast ratios.

### **3.2 Hardware Interfaces**

The software product interfaces with various hardware components in the system, encompassing supported device types, data and control interactions, and communication protocols.

The app is designed to seamlessly integrate with mobile devices running on the popular operating system Android and IOS. Communication between the software and the hardware occurs through standard protocols, ensuring reliable data transmission and control.

For live feed and video streaming, the app utilizes network interfaces, enabling real-time communication between the child's location, the events happening within the bus and the parent's device.

To facilitate facial recognition-based attendance, the app employs the camera interface to capture facial images, which are processed using algorithms to identify and notify parents when the child boards or disembarks from the bus. It leverages the device's camera to capture images of children for attendance purposes, relying on camera functionality and image processing capabilities.

Additionally, the app relies on GPS technology and interfaces with the device's location services to provide parents with real-time updates on the bus's whereabouts.



Lastly, the app also relies on a remote IP camera, which transmits video streams from inside the bus to the app, in order to allow parents to view the happenings inside the bus during the transit.

### **3.3 Software Interfaces**

The software product establishes connections with several specific software components, including databases, operating systems, Servers, tools, libraries, and integrated commercial components like Firebase. The app utilizes PostgreSQL as the primary database to store and manage the application's data, such as parent and child information, attendance records, and complaints, bus routes and information, school information, invoice generation and attendance records, camera information and feed recording etc.

It relies on Android and IOS operating system, depending on the targeted mobile platform, to provide a secure and reliable environment for app execution.

Additionally, the app employs various tools and libraries, such as Android Studio and VS code to develop and build the application.

The software product exchanges data items and messages with different components within the system. Data items coming into the system include parent and child information during the registration process, which consists of personal details, contact information, and uploaded images for facial recognition. Outgoing data items include real-time location updates of the bus, live feed and audio streams for parental monitoring, attendance notifications based on facial recognition, and complaint notifications. The purpose of each data item or message is to enable seamless communication between parents, children, and the app, ensuring efficient transportation services and parental engagement.

The software product requires various services for smooth operation and communication. These services encompass network communication services for real-time data

transmission, facial recognition services for attendance management, payment services for secure financial transactions, and notification services to inform parents about important updates and events. The nature of communications relies on standard protocols and APIs to ensure compatibility and interoperability among different software components.

Detailed application programming interface (API) protocols are specified in separate documents, defining the methods, parameters, and data formats required for seamless integration between the software product and its dependencies.

Data sharing across software components involves the secure exchange of information between the app, the database, and other integrated components. The data sharing mechanism is implemented using established APIs and protocols, ensuring secure data transmission and adherence to privacy regulations.

### **3.4 Communications Interfaces**

1. Email Communication:
  - The app requires email communication for user registration confirmation.
2. Web Browser Communication:
  - The app will support standard web communication protocols like HTTP (Hypertext Transfer Protocol) and HTTPS (HTTP Secure) for secure data transmission.
3. Network Server Communication:
  - The app will communicate with the network server to handle user registration, child information storage, attendance tracking, and other functionalities.
  - The communication protocol can be HTTP or HTTPS, depending on the security requirements and encryption considerations.

- Message formatting for server communication will be in JSON (JavaScript Object Notation).

## 4. System Features

This section reveals the essential features and powers of the software system that needs to be created. We explore the high-level parts and modules that together comprise the system's core in this section. These characteristics act as the cornerstones for achieving the particular requirements and goals listed in the SRS. In addition to providing designers, developers, and quality assurance teams with a solid roadmap for converting requirements into a fully functional and effective software solution, this section helps stakeholders get a thorough understanding of what the software will offer. This part serves as a roadmap, laying out the path for the system's development and making sure it complies with user expectations and project objectives.

### 4.1 Parent's Registration

Use Case Name	Parent Registration
XRef	Section 4.1, Parent Registration SRS Section 4, System Features
Trigger	The user selects the “Sign Up” Button on the app
Precondition	The user has downloaded the app on the main page.

Basic Path	<ol style="list-style-type: none"> <li>1) The user selects the Sign Up button and is taken to the first registration page.</li> <li>2) User enters an email and password which is sent to firebase and the user is sent an authentication email.</li> <li>3) After authentication the user is taken to the next page where he enters other information and then all the clicks on Register and all the information is sent to the database via server.</li> </ol>
Alternative Path	If the user provides the wrong email the screen never moves to the next screen and he keeps seeing “authenticate email” message.
Post Condition	The user is registered and now can register a child.
Exception Paths	<p>The server is down.</p> <p>Firestore is not working.</p>

#### 4.1.1 Description and Priority

To use the app the parents are first required to register themselves

It is a high priority feature as the parent cannot access the app's features before registering themselves.

Firebase is integrated with the app for this feature to authenticate the parents' emails.

#### 4.1.2 Stimulus/Response Sequences

- User Action: Parents opens the app and goes to SignUp and enters email and password

Response: The app sends the email and password to firebase which sends a link to the parents email and the parent is shown a message that to move forward he has to authenticate his email.

- User Action Parent authenticates his email.

Response: The app then moves to the next screen where the parent has to enter other information.

- User Action Parent enters other information.

Response: The information along with authenticated email and password are sent to the server which then stores it in the database and the parent is registered.

#### 4.1.3 Functional Requirements

REQ-1: System should be able to register parents.

REQ-2: System should authenticate the parent's email through firebase.

REQ-3: System should successfully send the data to the server and then store it in the database.

## 4.2 Child Registration

Use Case Name	Child Registration
XRef	Section 4.2, Child Registration SRS Section 4, System Features
Trigger	The user selects the “Register” Button on the app
Precondition	1)The user is registered. 2)The user has logged in.
Basic Path	1) The user selects the “Register Child” button and is taken to a screen where he has to enter the child's information. 2) Then he clicks next and has to enter some more information and three pictures of the child. 3) All the data and pictures after being checked are sent to the server and then to the database to be stored.
Alternative Path	If incomplete data is entered, the user is asked to enter correct data.

	If the pictures aren't of the face and of correct angles the user is asked to take pictures again.
Post Condition	Payment slip is generated for the child.
Exception Paths	The server is down.  Wrong data uploaded.

#### 4.2.1 Description and Priority

Child has to be registered to get assigned a school bus.

It is a high priority feature as all the functionalities can only be accessed once the child has been registered.

#### 4.2.2 Stimulus/Response Sequences

- User Action: Child's information is entered along with the child's pictures.

Response: System checks the information to see if it's complete and then checks the pictures to make sure the correct picture is uploaded for every angle. The data is then sent to the server which stores it in the database.

#### 4.2.3 Functional Requirements

REQ-1: System should be able to register multiple children under single parent.

REQ-2: System should send data to server and then database for storage.

### 4.3 Facial Recognition

Use Case Name	Facial Recognition
XRef	Section 4.3, Facial Recognition SRS Section 4, System Features
Trigger	Child enters or leaves the bus.
Precondition	1)The user is registered. 2)The user has logged in. 3)The child is registered and his pictures are in the database.
Basic Path	1) During registration when a child's pictures are taken and sent to the server, it extracts the child's encodings and stores them. 2) When the child enters or leaves the bus his picture is taken by the camera and sent to the server. 3) Server runs it through facial recognition code and if the match is found a notification is sent to the parent that the child has entered or left the bus.



Alternative Path	<p>Frames are continuously captured and sent to the server so in case the child isn't recognized the first time he/she is recognized it the next frames.</p> <p>If an unknown person is identified an alert is sent to the admin.</p>
Post Condition	<p>Parent receives notification and the attendance is stored in a file.</p>
Exception Paths	<p>The server is down.</p> <p>Camera is not working.</p> <p>Child not captured successfully.</p>

### 4.3.1 Description and Priority

The Facial Recognition feature is required for marking the attendance of children. Whenever the child gets on the bus or leaves the bus his attendance is marked through facial recognition and a notification is sent to the parent.

It is a high priority feature as the parent is notified about the child entering and leaving the bus through this feature.

The components required for this are a Camera and facial recognition libraries so this feature is a complex one. It will include the cost for the camera.

### 4.3.2 Stimulus/Response Sequences

- User Action: The child gets on the bus.

Response: The camera in the bus captures the child's picture and runs it through facial recognition. When matched with the database a notification is sent to the parent that the child got on the bus.

- User Action: The child leaves the bus.

Response: The camera again captures the child leaving the bus. It is then run through facial recognition and a notification is then sent to the parent that the child left the bus.

### 4.3.3 Functional Requirements

REQ-1: System should successfully perform facial recognition on the frames received and store the attendance for that child.

REQ-2: System should send notifications to parent when child is recognized.

## 4.4 Location Tracking

Use Case Name	Live Location
XRef	Section 4.4, Location Tracking SRS Section 4, System Features

Trigger	The user selects the “View Live Location” Button on the app
Precondition	The user has logged into his account and accessed the particular registered child's profile.
Basic Path	<ol style="list-style-type: none"> <li>1) The driver's phone is continually sending location information to the server.</li> <li>2) When the user accesses the live location feature, the system takes the most recent location data it has and displays it to the user using Google Maps API</li> </ol>
Alternative Path	If the child is not currently on the bus or the server is down, a “Location Not Available” message is shown.
Post Condition	The user is shown the live location of the bus
Exception Paths	The server is down.

#### 4.4.1 Description and Priority

The live location tracking feature allows parents to monitor the real-time location of their child's journey on the school bus.

This feature is of high priority and is one of the main deliverables of our app, as it is essential and most important for ensuring the safety and convenience of parents and students.

The live location tracking feature brings valuable benefits to parents by offering them real-time visibility into their child's school bus location. With this feature, parents can easily monitor the bus's progress (of pickup/dropoff), estimate arrival times, and plan their schedules accordingly, ensuring convenience for the parents and children.

The cost and risk of this feature are low compared to other features since there is no involvement of expensive hardware like a camera. The location information is gained by the help of a traditional mobile phone that the driver will be equipped with.

#### 4.4.2 Stimulus/Response Sequences

- User Action: The parent launches the application and logs in to their account.  
Response: The application authenticates the user and presents the parent with the dashboard.
- User Action: The parent selects the child's profile and chooses the live location tracking option.  
Response: The application establishes a real-time connection with the server to receive location updates.
- User Action: The parent views the map display to see the live location of the bus.  
Response: The application fetches the latest location data from the server and updates the map in real-time.
- User Action: The parent switches between different children within the application.

Response: The application updates the live location display based on the selected child.

#### 4.4.3 Functional Requirements

REQ-1: System should accurately display the live location of the bus.

#### 4.5 Push Notifications and Alerts

Use Case Name	Push Notifications and Alerts
XRef	Section 4.5, Push Notifications and Alerts  SRS Section 4, System Features
Trigger	An activity on the app has occurred.
Precondition	The user has enabled notifications for the app on his phone.
Basic Path	<ol style="list-style-type: none"> <li>1) When some activity on the app occurs, the system triggers notification to parents via FCM conveying the action name and event details.</li> <li>2) Parents promptly receive notifications, ensuring they stay</li> </ol>

	updated on important child transportation events.
Alternative Path	If the parent's device is offline or the app is inactive, parents can check for updates manually by opening the app and going to the notification center.
Post Condition	The parents receive timely updates.
Exception Paths	The server is down.  Parent phone is not connected to the internet.

#### 4.5.1 Description and Priority

The push notification feature will allow us to integrate a service such as Firebase Cloud Messaging (FCM), into the app. It enables real time communication between Parents and applications by sending notifications to their devices regarding important updates and events related to child transportation. Triggers are set up to send notification when specific events occur such as child boarding or leaving the bus etc. Notifications contain relevant information such as the child's name and event detail that shows timely and personalized communication. This keeps the parents updated and informed.

#### 4.5.2 Stimulus/Response Sequences

- User Action: Parent enables push notifications in the app settings.

Response: The app registers the parent's device token with the push notification service (e.g., FCM) and establishes a connection for receiving notifications.

- User Action: Child boards the school bus.

Response: The app triggers a notification event for boarding and sends a push notification to the parent's device.

- User Action: Parent taps on the notification to open the app.

Response: The app opens and displays the relevant information related to the child's boarding event.

- User Action: Child leaves the school bus.

Response: The app triggers a notification event for leaving and sends a push notification to the parent's device.

- User Action: Parent views the notification history within the app.

Response: The app retrieves and displays a list of past notifications, including details about boarding, leaving, and other relevant events.

- User Action: Parent submits a complaint through the app.

Response: The app registers the complaint and sends a push notification acknowledging the complaint submission to the parent's device.

- User Action: Parent receives the push notification acknowledging the complaint submission.

Response: The parent's device receives the notification and displays it as an alert or notification message.

### 4.5.3 Functional Requirements

REQ-1: System should successfully send notifications to parent whenever any event occurs.

REQ-2: System should allow parent to turn the notifications on/off

REQ-3: System should be able to show notification history to parents.

### 4.6 Live Camera Feed for Parents to View Children's Journey

Use Case Name	Live Camera Feed
XRef	Section 4.6, Live camera feed for parents to view children's journey  SRS Section 4, System Features
Trigger	The user selects the "View Live Feed" Button on the app
Precondition	The user has logged into his account and accessed the registered child's profile.
Basic Path	<ol style="list-style-type: none"> <li>1) System establishes a secure connection to the van's camera feed.</li> <li>2) Streams real-time video to the parent's device.</li> </ol>



	3) Parent can observe their child's journey, ensuring safety and peace of mind.
Alternative Path	If the child is not currently on the bus or the server is down, a “Live Feed Not Available” message is shown.
Post Condition	The user is shown the live feed of the bus
Exception Paths	<ol style="list-style-type: none"> <li>1) The server is down.</li> <li>2) The camera on the bus is not working</li> </ol>

#### 4.6.1 Description and Priority

This functionality empowers parents with the ability to effortlessly monitor their child's journey by providing real-time access to a live camera feed inside the school van.

Through this feature, parents gain visibility into their child's surroundings and activities during the commute, creating a sense of security and peace of mind regarding their child's safety. It addresses common parental concerns regarding the whereabouts of their child, bullying on the van, the exact condition of their child during transport, etc.

This is a High Priority feature since it contributes significantly to enhancing parents' peace of mind by establishing a visual connection with their child throughout the journey. By providing this capability, the feature fosters improved safety measures and is a source of great reassurance for the parents.

#### 4.6.2 Stimulus/Response Sequences

- User Action: The parent launches the application and logs in to their account.  
System Response: The application authenticates the user and presents the parent with the Children Profiles page.
- User Action: The parent selects the child's profile and chooses the View Camera Feed option from the main menu.  
System Response: The system establishes a reliable and secure connection to the camera feed inside the van. Once it has been established, the system streams the real-time camera feed to the parent's device. This camera feed, having clear video and audio, offers a live view of the child's entire journey. Through this feature, parents can actively observe their child's activities and surroundings throughout the journey, ensuring peace of mind and monitoring.
- User Action: Parent exits the camera feed and returns to the main Child's profile page.  
System Response: The system returns the parent to the main page upon exiting the camera feed.
- User Action: The Parent enables audio to listen to conversations or sounds inside the van.  
System Response: The system enables audio transmission when requested, allowing parents to listen to the happenings inside the van.
- User Action: Parent views Past Recordings.  
System Response: The system grants access to a database of the van recording of the bus for the past 15 days. The Parent is taken by the system to a page where the

dates are mentioned. They select the relevant dates, and are able to view the recordings of that date.

### 4.6.3 Functional Requirements

REQ-1: System should be able to display live stream from inside the bus.

## 4.7 Payment (Receipt Generation and Verification)

Use Case Name	Payment
XRef	Section 4.7, Payment (Receipt Generation and Verification)  SRS Section 4, System Features
Trigger	The user selects the “Proceed to Payment” Button on the app after during child registration.
Precondition	The user registered himself and provided all child information.

Basic Path	<ol style="list-style-type: none"> <li>1) The system generates a receipt based on the location the user entered and gives the user the option to select between online or by hand payment.</li> <li>2) If the user selects by hand payment, the system prompts the user to go to his nearest bank and submit his fee dues after which he is required to upload the paid receipt challan on the app.</li> <li>3) If the user chooses online payment, he is redirected to the payment page which has Google payment API integrated into it.</li> <li>4) After verification of the payment, the user is granted access to the system features.</li> </ol>
Alternative Path	<p>If in steps 2 and 3, no option is selected, the payment dues are considered to be not submitted and no verification of payment is made.</p>
Post Condition	<p>The user is granted access to all of the app's system features.</p>

Exception Paths	The user may abandon the operation at any time.
-----------------	---

#### 4.7.1 Description and Priority

One essential part of the program is the Payment module, which lets parents finish paying for their child's registration. Sustaining financial transactions and guaranteeing a flawless user experience depend on this module. Having two payment options—"By Hand" and "Online"—is the main goal. Parents are able to select the payment option that they want, which entails creating a payment receipt. This module has a high priority since it has a big impact on financial record-keeping and user satisfaction

#### 4.7.2 Stimulus/Response Sequences

The Payment module responds to user interactions and system events in the following sequences:

- User Action: User initiates the payment process after successfully registering their child.

System Response: The system redirects the user to the Payment page.

- User Action: On the Payment page, the user is presented with two payment options: 'By Hand' and 'Online.'

System Response: The system displays the respective payment methods for the user to choose from.

- User Action: If the user selects the 'By Hand' option:

System Response: The system provides instructions for making the payment at the nearest bank, along with an option to upload the paid fee challan.

- User Action: If the user selects the 'Online' option:

System Response: The system presents the Google Payment module, prompting the user to enter their credit card information and other required details for online payment.

- User Action: After successful payment is made:

System Response: The system generates a payment receipt and stores the transaction details.

- User Action: If the user uploads a paid fee challan:

System Response: The system verifies the uploaded document and updates the payment status accordingly.

### **4.7.3 Functional Requirements**

REQ-1: System should be able to generate fee slips.

REQ-2: System should allow parents to upload paid fee slips or pay the fee online.

REQ-3: System should successfully update the payment status after the payment is made.

## **5. Elaborated Functional Requirements**

### **5.1 Parent Enrollment:**

- System should allow parents to register using an authentic email.
- System should authenticate parents' email using firebase.
- The system must ensure that the parents data is stored in the database.

### **5.2 Registration of Children:**

- System must allow multiple children to be registered under a single parent.
- System must ensure that all fields of the child form while registering are complete.
- System must ensure that the child information, along with pictures, is stored in the database.

### **5.3 Recognizing faces**

- System must perform facial recognition based attendance.
- System must maintain a record of the attendance.

### **5.4 Tracking and Storage of Child's data**

- Entry and exit of children in a bus is tracked.
- System should store timestamps and relevant data.

## **5.4 Monitoring Location:**

- System must display the live location of the buses on a map view with markers for accuracy.

## **5.5 Push Alerts & Notifications:**

- System must send notifications to parents when any event occurs e.g child attendance.
- System should allow parents to turn the notifications on and off.
- System must allow parents to view notification history.

## **5.6 Live Camera Feed for Parents to View Children's Journey:**

- System must show live camera feed from inside the bus.

## **5.7 Payment (Generation and Verification of Receipts):**

- System should be able to generate a payment slip.
- System should allow parents to upload paid fee slips.
- System must accept the payment made online.

# **6. Nonfunctional Requirements**

## **6.1 Performance Requirements**

Following performance requirements should be met for the app to function properly



### **1) Responsiveness**

The app should be responsive all the time. So the user can navigate through the app easily.

The app should not hang or crash.

### **2) Managing high traffic**

The app should not crash during high traffic. That is when many parents want to view live feed or many parents are using the app together in general the app should not slow down.

### **3) Loading Time**

The app should load quickly every time the user uses it. Thus the response time should be as quick as possible.

### **4) Battery Efficiency**

The app should minimize power consumption of clients' smartphones.

For optimizing performance apps should efficiently use network resources.

## **6.2 Safety Requirements**

As child's and parent information is stored in the app, safety protocols must be established to protect that data.

Encryption protocols should be used while transmitting data over the network for example when publishing live feed to parents.

Adhere to child's protection laws when the child is in the bus. Cameras should be working all the time for that purpose. Only authorized parents should be able to see the whereabouts of the child.

Apps should be highly reliable to prevent system failures or crashes to ensure child's safety all the time.

The app should be prepared to handle emergency situations. For example, if the camera stops working the app should send notifications to parents and inform them about the current location of children.

### **6.3 Security Requirements**

To ensure the utmost security and privacy of user data, the application has several key requirements.

Firstly, data protection measures are essentially implemented, including secure storage and transmission of personal and sensitive information (such as picture storage of children) using encryption protocols.

Access control mechanisms are implemented to grant authorized individuals, such as parents and transportation service providers, exclusive access to the application and its data.

Robust authentication and authorization procedures, incorporating strong password policies and two-factor authentication, are used for user identity verification.

Secure communication protocols, like HTTPS, are utilized to safeguard data integrity during transmission.

The application has auditing and logging mechanisms to track security events, user activities, and access attempts for proactive detection of any security breaches.

A clear privacy and policy document will be created for the users prior to app release and making it public.

## 6.4 Software Quality Attributes

In addition to the core functionalities, the product aims to exhibit several key software quality attributes to satisfy the needs of both customers and developers.

These attributes include adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability.

To ensure **adaptability**, the system is designed with modularity and extensibility, allowing for easy incorporation of new features or updates.

**Availability** is emphasized by implementing fault tolerance, error handling and load balancing techniques on the server to minimize downtime.

The **correctness** of the product is assured through testing and adherence to specified requirements.

**Interoperability** is done by following standard protocols and interfaces for seamless integration with external systems.

**Maintainability** is facilitated by adopting clean code practices, comprehensive documentation, and an organized development structure..

**Reliability** is ensured by minimizing errors and code issues, implementing error handling and using reliable hosting services.

**Reusability** is promoted through modular code design and encapsulation of components.

**Robustness** is validated by subjecting the system to stress tests, handling unexpected inputs, and ensuring graceful error recovery.

**Testability** is supported through the implementation of well-defined test cases.

Lastly, **usability** is emphasized, giving precedence to ease of use over ease of learning to enhance user satisfaction and minimize the learning curve. Also done through testing it using different users to test the usability and give ratings.

## 6.5 Business Rules

Parents can register themselves as users and only registered parents can register their children.s

Three pictures of the child from different angles must be uploaded.

Invoice is generated and payment must be made and paid invoice should be uploaded for successful registration.

Only authorized parents can have access to live audio and video feed of their children during bus rides.

Only parents associated with a registered child will get the notification whenever the child enters or leaves the bus based on facial recognition.

Only authorized parents can have access to the real time location of the bus and recorded videos and attendance logs.

Parents can register complaints regarding the transportation service and they will be notified about the status of the complaint through notification or email.

The app should prioritize the security and privacy of user data including personal information, pictures and videos recordings.

The administrator must have the authority to manage users' accounts, buses and routes , complaint resolution and ensuring data security.

## **7. Other Requirements**

### **7.1 Scalability and performance:**

The app will prioritize accessibility and ease of use, providing a smooth and enjoyable user experience through responsive design, clear notifications, and intuitive controls.

Load balancing will distribute incoming requests across multiple servers to prevent performance bottlenecks and ensure efficient resource utilization as the user base and data volume increase. Database optimization techniques such as indexing and query optimization will be implemented to minimize response times and handle large amounts of data effectively, ensuring smooth data storage and retrieval. Caching mechanisms will be utilized to store frequently accessed data, reducing database load and improving response times for better overall app performance.

### **7.2 Usability:**

The app will feature a user-friendly interface with intuitive navigation, clear instructions, and well-organized menus, allowing parents to easily register their children and access different app features without confusion. Live feed viewing will be made convenient with a seamless interface, ensuring parents can effortlessly monitor their child's transportation and view real-time video and audio feeds. The app will prioritize accessibility and ease of

use, providing a smooth and enjoyable user experience through responsive design, clear notifications, and intuitive controls.

## **8. Innovative Feature Distinction**

A number of applications providing a variety of services have emerged in the software landscape for school transportation and child safety; nevertheless, the VanGO mobile application stands out due to its innovative and distinct feature set. Amidst a plethora of programs catering to parents' worries about school van administration and child safety, With its innovation in the school transportation niche, VanGO aims to redefine parental peace of mind in terms of child safety during school van commutes with these cutting-edge features. VanGO's unwavering commitment to innovation and ongoing development is a reflection of our commitment to guaranteeing the highest level of convenience and safety for both parents and children. Our innovative offerings, such as the Face Recognition-Based Attendance System and the Live Feed including longer video storage, showcase our dedication to establishing novel benchmarks in the market and cultivating steadfast confidence in the VanGO application.

VanGO presents user friendly and advanced solutions that differs it from other apps in the following ways:

### **8.1 Real Time Tracking and Monitoring**

VanGO incorporates the use of Live Cameras inside the buses along with Facial Recognition technology to monitor the child from the moment they enter the bus till the time they leave the bus. VanGO's Face Recognition-Based Attendance System notifies parents of the bus location and time in real-time. This means that whenever the child

boards the bus, a notification will be sent to the parents stating clearly that your child boarded the bus at e.g XYZ location at e.g 123 time. Similarly, when the child gets off the bus the same notification will be sent to the parents.

This enables parents to be able to track their child's journey to and from school both visually, through location and through alert notification. No other bus/van systems provide this functionality to the parents. This alleviates the parents' security concerns and gives them peace of mind while their child travels, since they can track the whole journey. This innovative feature not only guarantees real time monitoring, but it also represents a major advancement in our handling of parental concerns.

## **8.2 Attendance Marking with Time:**

VanGO cameras inside buses mark the attendance of children as they enter the bus, and mark them as having left the bus as soon as they leave. VanGO sends real-time alerts of the entry and exit of each child to their parents. These alerts represent a significant advancement in the safety of child transportation, since they are completely synced with the attendance marking procedure. Parents get immediate information whenever a pupil gets on or off the school bus. Important details like the child's boarding or departure event and the school bus's current live position along with the time stamp are included in these alerts.

## **8.3 Live Feed with Video Archive:**

VanGO goes above and beyond with its real-time capabilities by offering users access to safely stored video archives that are kept on our servers for a maximum of 10 days. This longer video storage time raises the bar for guaranteeing a thorough history of previous

trips. It makes it convenient for parents to look back on previous incidents, handle any issues, and have a thorough record of their child's transportation history. Parental control and peace of mind are greatly enhanced by this special mix of real-time monitoring and a 10-day video archive, which provides an unmatched level of convenience and security.

## **9. Contribution to United Nations Sustainable Development Goals (SDGs)**

The VanGO mobile application was created with a core dedication to tackling several Sustainable Development Goals (SDGs) of the United Nations. VanGO's inventive features and emphasis on school transportation efficiency, improved parental communication, and child safety help to achieve multiple Sustainable Development Goals (SDGs) that are part of the larger global agenda for sustainable development.

### **SDG 3: Good Health and Well-being:**

VanGO prioritizes children's safety and wellbeing above anything else. During their school bus rides, the program tracks and monitors their attendance in real time, which is a critical component in achieving SDG 3. Ensuring the health and safety of students is a top priority, and VanGO plays a major role in helping to achieve this aim by improving security and safety for children.

### **SDG 4: High-quality Education**



Sustainable development is predicated on high-quality education, and VanGO indirectly advances this objective. The application contributes to the removal of obstacles to education by improving the efficiency and safety of school transportation. More kids can attend school frequently when there is safe and dependable transportation, which increases access to high-quality education and supports SDG goals.

### **SDG 9: Industry, Innovation, and Infrastructure:**

The VanGO application uses state-of-the-art technology to improve parental communication and child safety, thereby embodying the spirit of innovation and advancement. VanGO embodies the values of SDG 9 by advancing the transportation industry with capabilities like face recognition and real-time tracking. The application emphasizes how technology has the power to change and enhance infrastructure, especially when it comes to child mobility.

### **SDG 13: Climate Action:**

The VanGO system, by promoting the culture of carpooling or multiple children traveling together, effectively combats the use of single cars dropping off singular students. Reducing the number of cars on the roads reduces carbon emissions, fuel consumption and thus the negative effects on climate change.

### **SDG 17: Partnership for the Goals**

VanGO serves as evidence of the value of collaboration in attaining sustainable development. In order to successfully install and use the platform, the application

emphasizes the necessity of cooperation between educational institutions, parents, and technology suppliers. VanGO emphasizes the value of teamwork in achieving the larger objectives of sustainability while upholding the values of SDG 17.

## **10. Appendixes**

### **10.1 Appendix A:**

#### **Glossary:**

1. App: Abbreviation for Application
2. GPS: Global Positioning System
3. UI/UX: User Interface/User Experience
4. API: Application Programming Interface
5. HTTP: Hypertext Transfer Protocol

## Project Timeline:



## Project Proposal:

<< VanGo: Alert Based Transportation App >>

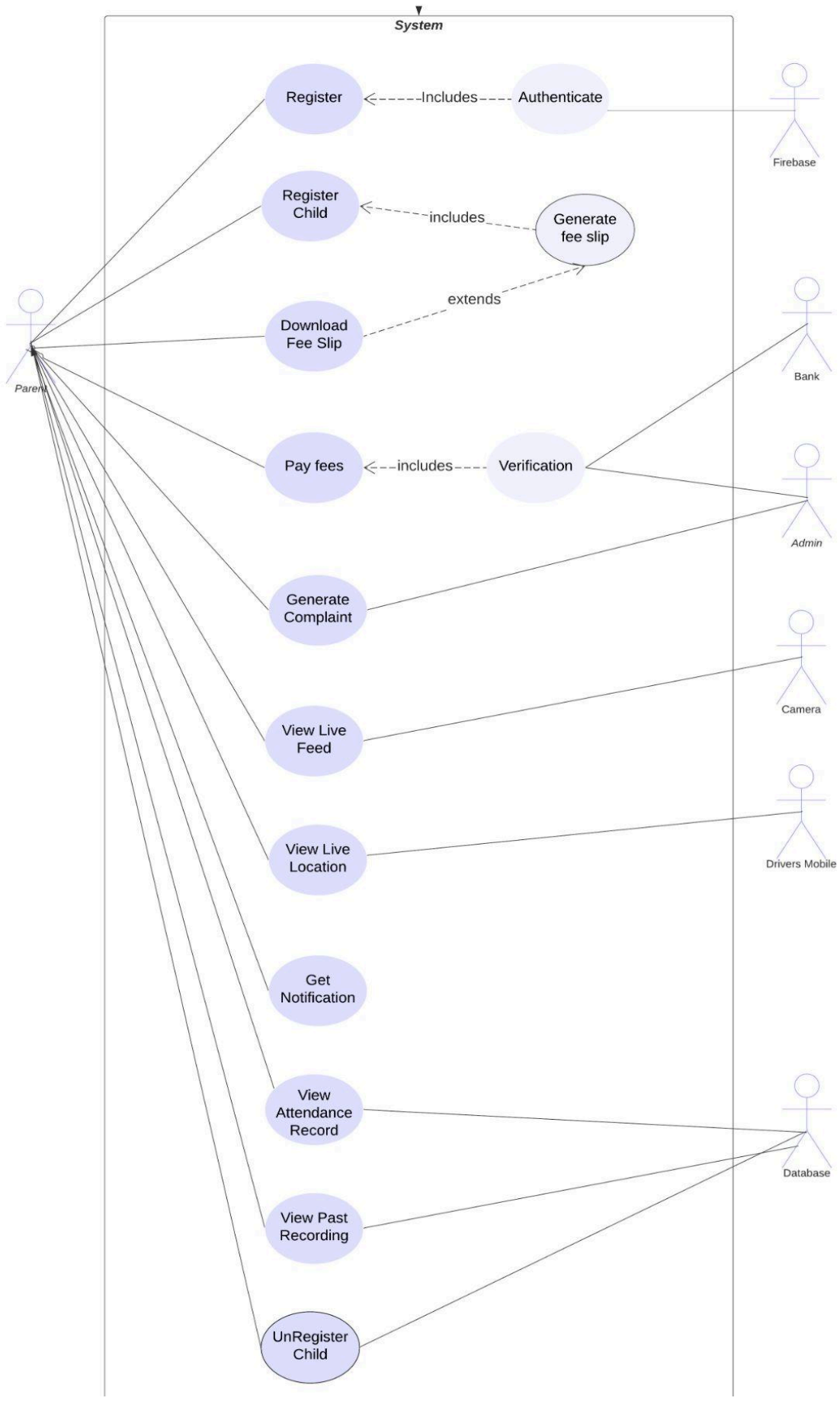
<b>Scope</b>	Our transportation app will offer a range of essential features to ensure safe and efficient student transportation. It will include live location tracking, allowing parents and administrators to monitor the real-time whereabouts of school vans. The app will also provide a live feed with audio, enabling parents to visually and audibly monitor their child's journey. Additionally, the app will incorporate face recognition attendance for accurate record-keeping, and the ability to store live feed data securely on the cloud.
<b>Deliverables</b>	<ul style="list-style-type: none"> <li>An indigenous hardware and software solution providing live location tracking and audio/video feed, ensuring high security and protection of children who are traveling alone on public transport via a bus/van</li> <li>System and Design specification document</li> <li>Market research and validation for taking the solution to the market</li> </ul>

Project Team Members	Supervisor
Shazray Shah	<b>Dr. Shahzaib Tahir</b>
Hadia Shafqat	
Hareem Ahmed	
Javeria Shehbaz	

## 10.2 Appendix B: Diagrams

### Use Case

This Use Case Diagram presents a comprehensive overview of the main features and interactions of the VanGO mobile application. It outlines the key use cases and their connections, as well as the different user roles and system components. This graphic illustration provides a thorough explanation of how the application meets the requirements of administrators, parents, students, and other stakeholders. This diagram, which highlights VanGO's high-level operations, offers an organized summary of the system's capabilities and is a useful tool for stakeholders trying to understand the main functions of the app and how they relate to each other.



## System Architecture Diagram

The basic architecture of the VanGO mobile application is shown in this High-Level System Architecture Diagram. This diagram provides a thorough overview of the architectural elements of the program, showing how they work together to provide a smooth user experience. It provides a high-level overview of the main components of the system, including the server, database, user interface, and external integrations. This picture, which offers an organized perspective of the system's architecture, is a crucial tool for comprehending how VanGO uses technology to guarantee child safety, continuous tracking, and effective communication while upholding strong architectural standards.

