SchoolNow: Master Software Requirements Specification (SRS)

1. Introduction

SchoolNow is a system consisting of two mobile applications designed to connect parents with school bus or van drivers who provide transportation services for primary and secondary school students. The system also facilitates efficient monthly school transportation fee management by enabling parents to make payments through the application while allowing drivers to track and manage payment records digitally.

The first application, SchoolNow, is intended for parents and students. It allows parents to monitor their child's transportation status, view the live location of the driver during active pickup or drop-off sessions (from home or school), and manage transportation payments. Students can scan and record their boarding status (whether they have boarded the bus or not) when traveling to school or returning home.

The second application, SchoolNow: Driver, is designed for school bus or van drivers. This application enables drivers to manage customers, view students' home locations, and plan optimal travel routes. The system assists drivers in generating routes from their starting point (either their residence or operator office) through each assigned student's pickup location and from the school back to the students' homes.

2. System Architecture & Tech Stack

- Framework: Flutter (iOS & Android)
- Backend: Firebase (Authentication, Firestore, Cloud Messaging)
- Maps: Google Maps SDK & Google Directions API
- Payments: Integrated Payment Gateway (for Parent-to-Driver transactions)

3. Detailed Functional Requirements

3.1 App 1: (SchoolNow: Driver) for Driver

FR-DR-1: Account Registration

- FR-DR-1.1: Registration Fields:
    - IC Number
    - Full Name
    - Email
    - Contact Number
    - Home/Operator Address (Starting Point)
    - Registered School Transportation Number
    - Vehicle Seat Capacity
    - Password
- FR-DR-1.2: Service Area Definition:

    - Select school on map
    - Choose side (North, South, East, West)
    - Specify radius (e.g., 10 km)
- FR-DR-1.3: Searchability:
    - Profile visible to parents only after verification

FR-DR-2: Authentication

- FR-DR-2.1: Login via IC number and password
- FR-DR-2.2: Backend: Firebase Authentication
- FR-DR-2.3: Error Handling: Display error and prompt retry on failure
- FR-DR-2.4: Post-Login: Redirect to main interface (Tabs: Drive, Students, Profile)

FR-DR-3: Drive Page (Service Operation)

- FR-DR-3.1: Display map with the registered service area
- FR-DR-3.2: Display list of students scheduled for the current day
- FR-DR-3.3: Route Information Logic:
    - Morning: Driver's home → students' homes → school
    - Afternoon: School → students' homes → driver's home
- FR-DR-3.4: Activation Time Restrictions:
    - 6:00 AM (Morning)
    - 1:30 PM (Primary PM)
    - 3:00 PM (Secondary PM)
- FR-DR-3.5: Live Tracking: Share location with parents in real-time when service is active
- FR-DR-3.6: Turn-by-turn navigation through all stops to final destination
- FR-DR-3.7: Generate unique session QR code for boarding
- FR-DR-3.8: Driver-side scanning of student QR codes (for students without phones)
- FR-DR-3.9: Boarding Status List: Display all students' current status
- FR-DR-3.10: Dynamic Map: Remove pickup pin once student is "Boarded"
- FR-DR-3.11: Driver Actions: One-tap call to parent or mark student as "Absent"
- FR-DR-3.12: Arrival Logic:
    - Morning: Mark all as "Arrived" at school
    - Afternoon: Mark individual students as "Arrived" at their homes
- FR-DR-3.13: Session End: Close service once all students are dropped off

FR-DR-4: Students Page

- FR-DR-4.1: Empty State: Display "No students under service" message
- FR-DR-4.2: Management: View list of students and parent profiles
- FR-DR-4.3: Direct Contact: Call parents from the student profile
- FR-DR-4.4: Approval Workflow: Approve or Reject new parent requests
- FR-DR-4.5: Refund Logic: Automatic refund to parent if request is rejected

FR-DR-5: Profile Management

- FR-DR-5.1/5.2: View and Edit profile details
- FR-DR-5.3: Secure logout

3.2 App 2: (SchoolNow) for Parent & Student

FR-PA-1: Parent Registration & Child Setup

- FR-PA-1.1: Fields:
    - IC Number
    - Name
    - Email
    - Contact
    - Home Address (Pickup Point)
    - Password
- FR-PA-1.2: Add Child: Name, Child IC, School selection via Map
- FR-PA-1.3: Auto-Account Creation: Student account uses Child IC + Parent Password

FR-ST-1/2: Student Functional Flow

- FR-ST-1.1: Login via Child IC and Parent Password
- FR-ST-2.1: Scan Driver's QR code to update status
- FR-ST-2.2: Display personal QR code for driver to scan

FR-PA-3: Drivers Page (Discovery & Booking)

- FR-PA-3.1: Search for drivers specific to a selected child
- FR-PA-3.2: Proximity filtering based on service area
- FR-PA-3.3: View driver details (Name, Contact, Monthly Fee)
- FR-PA-3.4/3.5: Payment workflow; status set to "Pending" after payment
- FR-PA-3.6/3.7: Assignment Logic: 1 student = Max 1 driver

FR-PA-4: Monitor Page

- FR-PA-4.1/4.2: Select child to monitor (prompt to select driver if none assigned)
- FR-PA-4.3: Live map tracking of active driver
- FR-PA-4.4: Display status: Not Boarded, Boarded, Arrived

FR-PA-5: Profile & Settings

- FR-PA-5.2: Edit Profile (Address locked after driver assignment)
- FR-PA-5.5: Print/View student QR codes (for offline students)
- FR-PA-5.6: Default Attendance: Reset to "Attending" if not updated by parent
- FR-PA-5.7: Notifications: Proximity alerts and boarding status changes
- FR-PA-5.8: Billing: Automatic renewal reminders 2 days and 1 day before due date

4. Technical Rules for Code Generation

Logic Category	Rule Description
The "Lock" Rule	Disable home_address text field in Parent App if student.assigned_driver != null.
Morning Route	Sort stops: [Driver Start] → [Student 1...N sorted by distance] → [School].
Afternoon Route	Sort stops: [School] → [Student 1...N sorted by distance] → [Driver End].
Real-time Map	Use StreamBuilder in Flutter to listen to Firestore live_location updates.
Refunds	If ServiceRequest.status changes to rejected, trigger a Cloud Function for payment reversal.

1. Firebase Database Design for SchoolNow

1.1 Firestore Collections

1. Users

Stores all users: drivers, parents, and students.

Field	Type	Description
user_id	String (UID)	Primary Key (Firebase Auth UID)
role	String	'driver', 'parent', 'student'
name	String	Full name
ic_number	String	IC Number
email	String	Email address
contact	String	Phone number
address	String	Home or pickup address
password	String	(optional, stored only if not using Firebase Auth)
created_at	Timestamp	Account creation time
updated_at	Timestamp	Last update time

2. Drivers

Specific info for drivers.

Field	Type	Description
driver_id	String	Reference to Users.user_id
registered_school_transport	String	Vehicle number
seat_capacity	Number	Total seats
service_area	Map	{ school_id, side, radius_km }
verified	Boolean	True if admin approved
created_at	Timestamp	Creation time
updated_at	Timestamp	Last update time

3. Parents

Specific info for parents.

Field	Type	Description
parent_id	String	Reference to Users.user_id
children_ids	Array of Strings	List of student_ids
billing_info	Map	{ next_due_date, status }
notifications	Map	{ proximity_alert: Boolean, boarding_alert: Boolean }
created_at	Timestamp	Creation time
updated_at	Timestamp	Last update time

4. Students

Specific info for students.

Field	Type	Description
student_id	String	Reference to Users.user_id
parent_id	String	Reference to parent
school_id	String	School assigned
qr_code	String	Personal QR code for scanning
assigned_driver	String	Reference to driver_id
attendance_status	String	Not Boarded / Boarded / Arrived
created_at	Timestamp	Creation time
updated_at	Timestamp	Last update time

5. Schools

Stores all schools.

Field	Type	Description
school_id	String	Primary Key
name	String	School name
address	String	School address
geo_location	Map	{ latitude, longitude }

created_at Timestamp Creation time

updated_at Timestamp Last update time

6. Service Sessions

Tracks each bus trip.

Field Type Description

session_id String Primary Key

driver_id String Reference to Drivers.driver_id

date Timestamp Date of trip

start_time Timestamp Service start

end_time Timestamp Service end

route_type String Morning / Afternoon

student_ids Array of Strings List of students for this session

status String Active / Completed

qr_session_code String Unique QR code for session

7. Payments

Tracks parent payments for drivers.

Field Type Description

payment_id String Primary Key

parent_id String Reference to Parents.parent_id

driver_id String Reference to Drivers.driver_id

amount Number Paid amount

status String Pending / Completed / Refunded

payment_date Timestamp Date of payment

created_at Timestamp Record creation

8. Notifications

Push notifications.

Field	Type	Description
notification_id	String	Primary Key
user_id	String	Reference to Users.user_id
type	String	Proximity / Boarding / General
message	String	Notification content
read	Boolean	True if user has read
created_at	Timestamp	When notification sent

1.2 Firebase Realtime Database

Use Realtime Database for live tracking and dynamic updates:

1. Live Locations

```
/live_locations/
  driver_id/
    lat: number
    lng: number
    timestamp: number
```

driver_id: Firebase UID of driver

lat, lng: current coordinates

timestamp: last update time

2. Real-time Boarding Status

```
/boarding_status/
  session_id/
    student_id/
      status: "Not Boarded" / "Boarded" / "Arrived"
```

last_update: timestamp

Updates dynamically for parents and driver apps

Can be displayed in real-time without polling

3. Real-time Notifications (Optional)

/real_time_notifications/

user_id/

notification_id/

type: "proximity" / "boarding"

message: string

read: boolean

created_at: timestamp

1.3 Relationships Summary

Driver ↔ Session: One-to-many (one driver can have many sessions)

Session ↔ Students: Many-to-many (session contains multiple students)

Parent ↔ Students: One-to-many (one parent can have multiple students)

Driver ↔ Parent Payments: One-to-many (payments made by parents to driver)