

# Database Design Document

## Robin's Nest Parent App

### 1. Overview

The Robin's Nest Parent App uses a non-relational database design built with Firebase. The application uses Cloud Firestore to store data, Firebase Authentication for login and user sessions, and Firebase Storage for images. This structure supports real-time updates and mobile-friendly performance.

### 2. Database Type

Database System: Firebase Cloud Firestore

Database Type: Non-relational (NoSQL)

### 3. Collections Overview

- users
- attendance
- pickupPlans
- photo\_posts
- updates

### 4. Data Model

#### *A. users*

Stores parent and admin user information.

Fields:

- uid (from Firebase Authentication)
- loginEmail
- parent1FirstName
- parent1LastName
- parent1Email
- parent1Phone
- parent1Role
- parent2FirstName (optional)
- parent2LastName (optional)
- parent2Email (optional)
- parent2Phone (optional)
- parent2Role (optional)
- child1FirstName
- child1LastName
- child1Dob
- child2FirstName (optional)
- child2LastName (optional)
- child2Dob (optional)
- accountOwner
- createdAt (if used)

### ***B. attendance***

Stores attendance submissions made by parents or guardians.

Fields:

- userId
- childFirstName
- childLastName
- status
- note
- timestamp

### ***C. pickupPlans***

Stores pickup updates submitted by parents or guardians.

Fields:

- userId
- childFirstName

- childLastName
- pickupFirstName
- pickupLastName
- relationship
- note
- timestamp

#### ***D. photo\_posts***

Stores photo feed posts created by admins.

Fields:

- title
- body
- timestamp
- imageUrls (if used / future expansion)

#### ***E. updates***

Stores announcements shared with parents.

Fields:

- title
- message
- createdBy
- createdAt

## **5. Firebase Storage**

Used for storing images for the photo feed. Image URLs are saved in Firestore and retrieved by the app to display posts.

## **6. Authentication**

Firebase Authentication manages login, registration, and session persistence using unique user IDs.

Each authenticated user is assigned a UID, which is used to connect authentication data with Firestore documents.

## **7. Relationships**

- A user can create multiple attendance records
- A user can create multiple pickup plans
- Admin users create photo posts and updates
- Photo posts connect to Firebase Storage through image URLs

## **8. Conclusion**

This Firebase-based design supports scalability, real-time communication, and efficient data management for a mobile app environment. This structure reflects how data is accessed in the app, prioritizing fast reads and simple document retrieval over complex joins, which is appropriate for a mobile Firebase-based application.