NOODLE

MOBILE APPS DEVELOPMENT REPORT

Word done by

Gonçalo Faustino – 20231721

Juliana Reis – 20231590

Madalena Duarte – 20231614

Sofia Saraiva – 20231621

Professor

José Américo Alves Sustelo Rio

Introduction

Noodle is a role-based educational management app designed for administrators and teachers to manage classes, subjects and student attendances efficiently.

The main purpose of Noodle is to simplify and digitize classroom management by allowing users to create and manage accounts, classes, subjects and students all within a mobile interface.

The target users are school administrators and teachers who need a simple and intuitive way to manage classroom data.

Technologies Used

- React Native for building the mobile app interface
- Expo for fast development and testing
- Firebase Authentication for user login and password reset
- Firestore for storing and retrieving user and app data

System Requirements

To run and develop the app, the following tools and environments are required:

- **Node.js**: Runtime environment for running JavaScript code outside a browser.
- **Expo CLI**: Used to develop and run React Native apps easily.
- **Android Studio or a physical Android/iOS device**: For testing the app on an emulator or real phone.
- **Code Editor (ex: VS Code)**: For writing and editing the source code.
- **Firebase Console**: For managing authentication and Firestore database.

App Architecture

The app is organized around two user roles: **Admin** and **Teacher**, each with specific permissions and screen access.

Admin Perspective:

- **Login** – Authenticates the user.
- **AdminHome** – Main dashboard for admins.
- **ManageSubjects** – Central hub to manage subjects.
    - **CreateSubject** – Add new subjects.
    - **SubjectDetail** – View full subject info.
        - **EditClass** – Admin-level class editing.
- **CreateAccount** – Allows admins to create new users (teachers).

Teacher Perspective:

- **Login** – Authenticates the user.
- **TeacherHome** – Main dashboard for teachers.
- **SubjectDetail** – Teachers can view subject info (with limited permissions).
- **EditClass** – Teachers can update class info (with limited permissions).

User Roles and Permissions

The app supports two distinct user roles: **Admin** and **Teacher**, each with a tailored experience and specific access rights. These roles are determined during login based on data stored in Firestore.

Admin:

- Full access to all app features.
- Can **create teacher accounts**.
- Can **create and edit subjects**.
- Can **view subject details** and **edit class data**.
- Access to **AdminHome** and **management screens**.

Teacher:

- Limited access for class and subject interaction.
- Can **view assigned subjects** and **see related details**.
- Can **edit class data**, but with restrictions (no class creation).
- Access to **TeacherHome** and **subject/class details** only.

Permissions Control:

- Role data is fetched from Firestore after login.
- Based on the role (admin or teacher), the app dynamically navigates to the correct interface.
- UI components and screens are **conditionally rendered** depending on the role of the user (ex: Create class button on the subject details interface).

Challenges Faced

One of the main challenges was dealing with inconsistencies and redundant interfaces, especially before having a clear database structure. Without a solid plan for how data would be stored and accessed, it was harder to manage logic across screens and roles.

Conclusion

This project allowed us to build a functional mobile app using React Native, Expo, and Firebase. We learned how to manage different user roles, connect with a real-time database, and handle navigation between screens. Despite some initial challenges, especially with planning the structure, the result is a working admin-teacher system that can be further expanded in the future.

Group 07

Gonçalo Faustino - 20231721

Sofia Saraiva - 20231621

Juliana Reis - 20231590

Madalena Duarte - 20231614