BSCS-F24-002

03-134212-082 SADIA INAM

03-134212-059 MUHAMMAD HAIDER ALI

CareGrow

In partial fulfilment of the requirements for the degree of

Bachelor of Science in Computer Science

Supervisor: Dr Iram Noreen

Department of Computer Sciences

Bahria University, Lahore Campus

June 2025

C e r t i f i c a t e

We accept the work contained in the report titled

"Care Grow"

written by

SADIA INAM

MUHAMMAD HAIDER ALI

as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Computer Science.

Approved by:

Supervisor: Dr Iram Noreen

_____ (Signature)

June, 2025

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134212-082	SADIA INAM	
03-134212-059	MUHAMMAD HAIDER ALI	

Date : June, 2025

Specially dedicated to
my beloved grandmother, mother and father
(Sadia Inam)
my beloved grandmother, mother and father
(Muhammad Haider Ali)

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express my/our gratitude to my research supervisor, Dr Iram Noreen for her invaluable advice, guidance and her enormous patience throughout the development of the research.

In addition, We would also like to express my gratitude to our loving parent and friends who had helped and given me encouragement.

SADIA INAM
MUHAMMAD HAIDER ALI

CAREGROW

ABSTRACT

With today's hectic life, it is difficult for parents to keep track of their children's health information such as immunization history, growth, medical history, and doctor visits. Paper-based systems are disorganized, error-prone, and difficult to handle, especially where accurate information is needed to provide timely medical treatment.

CareGrow is the recommended Android medical app that is specifically designed to enable parents to track and manage the medical information of the children in a single virtual database. Provision for the user to own and operate multiple child accounts in the application, store medical history, monitor body growth by tracking BMI/BFP levels, and view auto-schedule reminders for vaccination is made. Also, there is in-built live chat and video call capability built as live communication facilities within CareGrow, allowing live parent-doctor communication directly through chat or video calling for consultative purposes.

Built using Java, Kotlin, and XML and driven by Firebase services, the system offers secure data management, real-time synchronization, and rapid UI. The application is also built with scalability, modularity, and role-based access to offer access to sensitive data while being secure and accessible to authorized staff only.

With CareGrow, parents are empowered to take more responsibility for the care of their children and ultimately to extend the quality of care and communication with health professionals. The system not only fills the gap between doctors and families but also introduces a new, handy concept of e-child care.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

BMI	Body Mass Index
BFP	Body Fat Percentage
XML	eXtensible Markup Language
WHO	World Health Organization
SDLC	Software Development Lifecycle
API	Application Programming Interface
SDK	Software Development Kit
GUI	Graphical User Interface
JSON	JavaScript Object Notation
URL	Uniform Resource Notation
HTTPS	HyperText Transfer Protocol Secure
UI	User Interface
MAP	maximum allowable pressure, kPa
MAWP	maximum allowable working pressure, kPa
OD	outer diameter, m
PCB	printed circuit board

CHAPTER 1

INTRODUCTION

1.1 Background

In the face of today's quickening pace in the digital world, a child's health needs are more challenging to navigate than ever before. Parents are finding it difficult to keep up with immunization records, track physical changes, and hold medical files in one place. Physical paper or piecemeal computer note-taking record-keeping systems risk being lost, not comparable, and laborious—particularly when physicians' office visits involve transferring medical information.

With the massification of smartphone coverage and health technology, mHealth solutions now offer viable means to help manage family and individual health. According to the World Health Organization (WHO), digital interventions such as mobile applications can support health systems efficiently through enhanced access, participation, and quality of information. Even with this global momentum, most of the mHealth platforms available focus on adult care, general telemedicine, or fitness tracking, creating a glaring shortage of platforms designed particularly for paediatric care management.

Evidence has also revealed that parent-support apps tracking children's development—i.e., monitoring growth and reminding immunizations—have the potential to enhance healthcare regularity and parental attentiveness. Nevertheless, most mHealth apps are isolated, too poorly regulated, and tainted by data privacy, accuracy, and usability shortcomings [1]. These substantiate the requirement for apps

specifically focused on bridging such loopholes while fulfilling the specific needs of families with kids.

To address these challenges, this project proposes CareGrow—a mobile app built from Android that gives parents a virtual, centralized portal on which to monitor the health of their children. CareGrow supports the establishment of over a single child profile, each of which offers medical history tracking, growth tracking in the form of BMI and BFP calculation, and vaccine reminder. It also provides live communication facilities like live chat and video consulting with physicians for bridging gaps between physicians and families in an effective and time-saving manner.

1.2 Problem Statement

Health records management is a headache in today's busy life, which further complicates the management of medical histories, vaccination schedules, and growth charts for family members. This healthcare app makes the family health record keeping, and doctor's consultation easy and simple by providing medical assistance on a click.

1.3 Aims and Objectives

The main objective of the project is to develop and implement an Android application enabling parents to maintain children's health records in a readily available, easy-to-use, and convenient manner.

Among its main objectives are:

I. To allow a secure platform upon which multiple profiles of a child's health can be built and controlled.
II. To support the taking and monitoring of medical history and diagnosis, as well as treatment.
III. To provide automatic generation of vaccination schedules and reminders based on children's age.
IV. To compute and track BMI and BFP over time.
V. To enable parents to upload and view medical prescriptions with ease.
VI. To facilitate live chat.

1.4 Scope of Project

CareGrow's project scope is to develop and design an Android health application wherein parents can keep and manage the medical history and health of their children. The system should be a single-source digital assistant wherein users can save, access, and update important health information in real time through a mobile interface. These are aspects like monitoring of clinical history, growth monitoring by BMI, and immunization scheduling with reminders.

CareGrow enables parents to create and control multiple child accounts, each one being an autonomous keeper of medical records. Parents can fill in each account with significant medical events like doctor visits, history of treatment previously administered, diagnoses recorded and prescriptions dispensed in every account. CareGrow also makes secure uploading and reading of medical scripts possible with a digital and centralized substitute to paper-based recording.

One of the most significant specifications under the project is to find out the actual time Body Mass Index (BMI) of the child on the basis of provided height and weight values at a regular interval of time. The system offers graphical display of the trend of child's growth and classifies the outcome as per health norms. The app also identifies the vaccination schedule automatically based on the child's birth date and sends reminders appropriately so that a single dose is never missed.

To ensure continuity and access to treatment, CareGrow also has a real-time chat option by which parents can communicate with enrolled physicians. The feature is employed to fill the gap in communication between the physician and patient, especially where personal visits are not feasible or delayed.

Although the original project scope encompasses key features required for effective child health management, it sets the stage for planned future upgrades as well. Plans for future upgrades would include features like health analysis based on artificial

intelligence, support for multiple languages, video calling-based teleconsultation, wearable device integration, and health blogs to inform parents.

CareGrow's sole mission in entirety is simply to provide an effective, scalable, and secure mobile platform by which parents can positively influence the healthcare experience of their children with confidence and ease.

CHAPTER 2

LITERATURE REVIEW (and/or SRS)

2.1 Background

Child health care is an ongoing and precise process that includes good records, current vaccinations, and monitoring physical growth with age. Practically, most parents employ non-electronic aid—paper records, printed immunization cards, or memory—to manage these basic processes. The result is general loss of medical records, late vaccinations, and lost health histories, with a negative impact on medical treatment in a timely manner.

In addition, in doctor visits, parents may not be able to provide thorough information, hence compromising diagnosis and treatment. Notwithstanding the availability of digital health platforms, these are mainly for adults and are applied for general health, chronic disease, or online consultations. These platforms have no or minimal child-centred monitoring aspects of health and do not accommodate central storage of child medical records in an accessible parent interface.

Apart from that, immediate real-time communication between physicians and parents is yet to be a part of most existing apps, particularly concerning the health of a child. Such a lack of communication tends to result in delayed consultations or follow-ups, particularly in rural or developing regions where visits to the hospital may be few.

Thus, there is an urgent need for a standalone mobile solution that:

- Provides centralized, secure digital storage of children's health information.
- Reminds parents about overdue or forgotten vaccines and monitors them.
- Facilitates one-on-one communication centre with physicians through live chat.

CareGrow tries to fill this gap by offering a trusty, Android-focused health application committed to child care—allowing parents to effectively and in advance optimize the health of their child.

2.2 Related Work

Over the past ten years, use of mobile technology in healthcare has transformed medical information management and provider-to-provider and provider-to-patient communication. mHealth technology has become a valuable asset in the advancement of patient outcomes, telemonitoring, and healthcare information accessibility. The majority of the mHealth apps are not specifically designed with the intention of addressing the health of a child, with its specific requirements in terms of growth monitoring, vaccination schedules, and interventions by the age of the child.

Siswati et al. [2] present design and usability testing of an mHealth application to track children's growth and development. The authors observed that computerized tracking dramatically enhanced parent engagement and early detection of abnormal growth. Likewise, Esiefarienrhe and Makokoe [3] presented an example of a parent-child immunization reminder system, which enhanced immunization coverage rates due to timely push reminders, identifying the efficacy of app reminders.

Apart from that, Uchechukwu and Ohinameuwa [4] sought to explore improving health record management with the help of mobile development frameworks, and the findings indicated that electronic systems improve better access to information, patient safety, and care coordination. This is vital in pediatric health care, where monitoring and storing information requires to be a continuous process for long durations. In another similar study, Abdullah and Alghali [5] highlighted the importance of drug information management and clinical records via mobile

applications once again demonstrating that central systems minimize errors due to labour and improve adherence to medication.

All the aforementioned researches reflect increased interest towards mobile applications as a possible area of healthcare management. Yet, with all the advancements in mHealth solutions across the board, there are no end-to-end systems for children that come equipped with features such as profile-based medical history, vaccination automatically prefilled, real-time continuous BMI tracking, and physician contact built-in. That is where CareGrow enters, an end-to-end mobile healthcare assistant specifically crafted for the purposes of children and parents.

2.2.1 Comparative Analysis of Existing Solutions

The Table 2.1 presents a comparison between CareGrow and some of the most popular healthcare applications around the world, including those available in Pakistan like Oladoc, Marham, Sehat Kahani, and BabyCenter. This comparison pinpoints the innovative child health management and feature bundling of CareGrow.

Table 2.1: Comparison of proposed CareGrow and existing applications

Feature	**CareGrow**	**Oladoc [7]**	**Marham [8]**	**Sehat Kahani [9]**	**BabyCenter [10]**
Platform Focus	**Child Healthcare**	**General Health**	**General Health**	**Telemedicine**	**Child development**
Multiple profiles	✓				✓
Medical history Logging	✓				
Vaccination Reminder	✓				✓ (Articles)
BMI Tracking	✓				✓ Partial
Prescription Upload	✓	✓	✓	✓	
Chat with Doctors	✓	✓	✓	✓	
Video Consultation	✓	✓	✓	✓	
Parent-Focused Health Monitoring	✓				✓

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Development Method

Agile Software Development Life Cycle (SDLC) has been utilized as the development method in trying to develop the application CareGrow. Agile was utilized due to its nature to be iterative and very effective in projects with high chances that their requirements are bound to change as time progresses. Because CareGrow had so many elements embedded within it—like child profile tracking, scheduling immunizations, health monitoring, integration of chat, and real-time notification—the team was able to build the system piece by piece, one might say, and be receptive to comment and inspiration in the meantime.

3.1.1 Why Agile Was Chosen?

Agile was utilized over more conventional methodologies like the Waterfall model because it supported early release of key functionalities and modular construction.
For CareGrow, with user requirements and features changing over the years, Agile allowed the team to re-prioritize, consolidate functionality from testing, and add new features such as video consultation and live chat without requiring an expansion of the project timeline. The approach also allowed the application to be functional and stable at every level of development.

3.1.2 Sprint Planning and Execution

The project was broken up into a series of cycles of creation, or sprints, and with each sprint there was an agenda of features that were addressed.

The team established its objectives, deployment planning, and established time estimates on the beginning of every sprint. User authentication, child profile, growth monitoring, vaccine reminder, and communication modules were developed iteratively and incrementally. Verification of completion and testing work done was carried out at the end of every sprint prior to advancing to the subsequent stage so that the project advances stepwise and incrementally.

3.1.3 Iterative Development and Continuous Feedback

Each round was a continuous process of development, designing, testing, and feedback. Development comprised creation of system flows and user interfaces and backend and frontend coding with Firebase and Android Studio. After preparation of all modules, the application was unit tested and integrated to confirm whether it was executing as designed and error-free. Internal test feedback and external test feedback were harvested and utilized to implement features already in production or guide coding for new functionality in upcoming sprints. The loopback was always established so that the application was always aligned with user expectation and technical ideals.

3.1.4 Feature Expansion Adaptability

New functionality emerged along the project path that was not initially planned. They consisted of live chat functionality for real-time parent-doctor interaction and video calling using third-party APIs. That the new modules were added to the existing app without stopping it from working and with extensive redeployment involved was achievable because Agile was employed. The method allowed concurrent development and maintained the current components running while adding the new components.

3.2 Tools and Technologies

CareGrow app was created using the latest programming languages with an open toolchain and third-party libraries and SDKs developed by Google. The technology used was intended to create an efficient working, quick response performance, and secure data processing in the Android system.

3.2.1 Programming Languages

It was created based on:

- Java
- Kotlin
- XML

Java was utilized as the principal programming language for back-end operations like user login, database queries, data processing, and feature management. Its reliability and utilization in Android application development made it a flawless choice to be applied when integrating core functionality.

Some of the modules utilized Kotlin where it was given added efficiency and less boilerplate code through new features in the language and concise syntax. Java interoperability allowed them to coexist peacefully within one project.

eXtensible Markup Language (XML) was used to specify the user interface layout. Every single aspect of the screen—buttons, text inputs, charts, and menus—was specified using XML to describe structured and dynamic UI design across a variety of Android devices and displays.

3.2.2 Development Tools and SDKs

Development tools and software development kits (SDKs) were used to bring backend integration, system interaction, and compatibility into the device.

- Firebase SDK
- Google SDK
- Android SDK
- Android Jetpack

The foundation of the system was combined through Firebase SDK, which provided core backend services such as Authentication, Realtime Database, Cloud Storage, and Cloud Messaging. These provided real-time synchronization of data, file storage in a secure manner, and instant delivery of messages.

Google SDK was utilized to provide device-level functionality such as user authentication, Firebase initialization, and integration with Google services. It provided a middle layer for cloud services and Android components.

Android SDK was used as the platform upon which to write, test, and run the application on numerous various Android-based devices. It included the system libraries and APIs necessary to provide Android-specific functionality such as activity life cycle management, permissions, background services, etc.

Apart from all this, Android Jetpack was used in a bid to make the development simpler with its collection of new architecture components and libraries. Its components such as ViewModel, LiveData, Navigation, and Room kept the codebase organized to remain clean and uncluttered without losing on modular development concepts.

With all those features and languages, CareGrow made sure that it was built on a firm foundation base, real-time features within the app, and simplicity, nothing being sacrificed on newer Android smartphones.

3.3 System Architecture

CareGrow is client-server Android application as client and Firebase as server (backend). Operations performed by user like login, insertion into registers, image uploading, or chat are obtained by invoking the calls which are dynamically mapped to Firebase services.

Layers used in the system are described below:

- **Presentation Layer (Frontend):** Android GUI implemented using Java and XML.
- **Business Logic Layer**: Contains logic for the calculation of BMI, vaccination, and chat.
- **Data Layer (Backend):** Firebase offers Authentication, real-time database, cloud storage, and messaging.

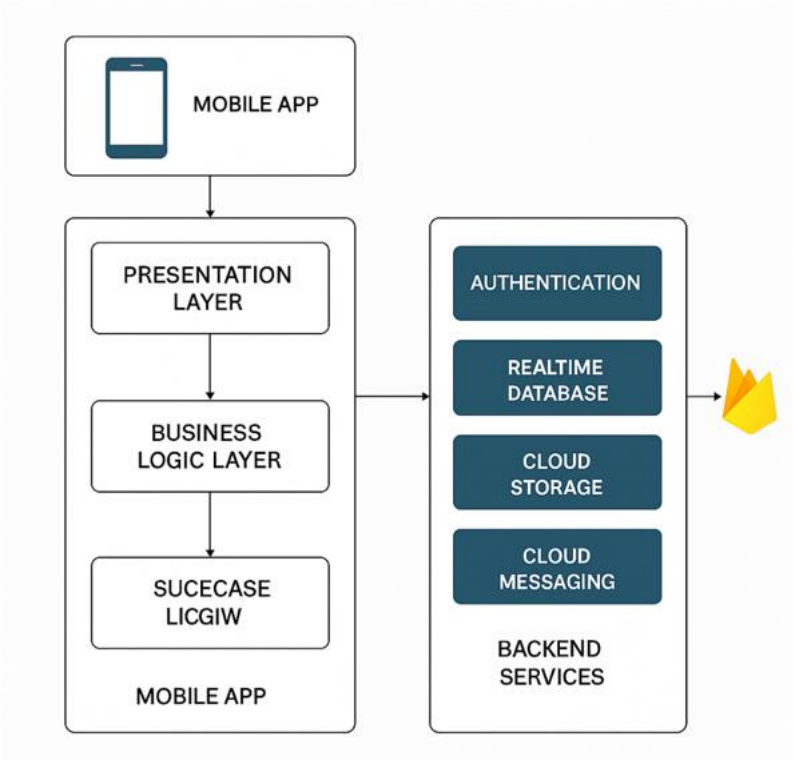


Figure 3.1: Layers used in system

3.4 Database Design

The Firebase Realtime Database is a JSON tree, and therefore mapping parent IDs to child profiles and corresponding medical records is easy.

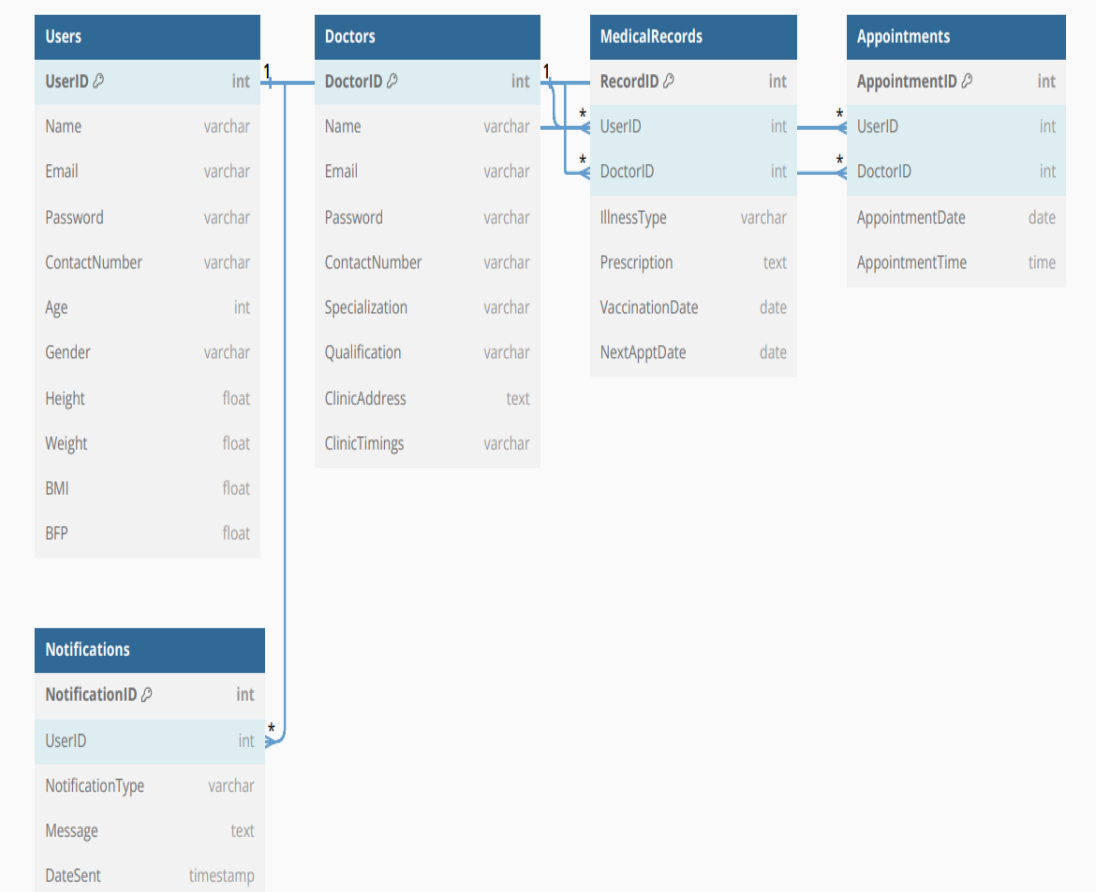


Figure 3.2: Database Schema

3.5 Interaction of Components and Data Flow

CareGrow app utilizes module and data-flow architecture in which all user activity is triggered directly or indirectly by the backend actions along with UI rendering with practically negligible lag. App triggers interaction with Firebase the moment it gets initialized by the user. Firebase Authentication authenticates the user credentials, and their validity is cross-checked. Secondly, the personal data like children profiles, medical history, and vaccines is fetched by the system from Realtime Database. The information is loaded and rendered in the dashboard dynamically using real-time listeners without any manual refreshes.

Every time a parent adds or modifies a child's profile, medical history, or growth information, the app posts structured data to Firebase under corresponding nodes. These are the "Children," "Vaccinations," "GrowthMetrics," and "MedicalHistory" branches. As soon as the data is saved, it's automatically updated in the UI, due to

Firebase's real-time syncing feature. Likewise, when the user uploads a photo of a prescription, the file is stored in Firebase Cloud Storage while a download URL is stored in the database so that it can be retrieved later. Such frontend input aggregation and backend computation ensure seamless data processing. Communication process is extended to communication between parents and physicians through chat and video calls.

When the parent initiates a message or an outgoing call, the request is forwarded to the physician's interface. The messages in a chat are kept along structured paths in the database to maintain the conversation history. Video calling, if made available as an external API, is invoked from within and combined with the child record for context. All activities have strict permission control so that the users only get to retrieve information about their accounts, thus ensuring privacy and system integrity.

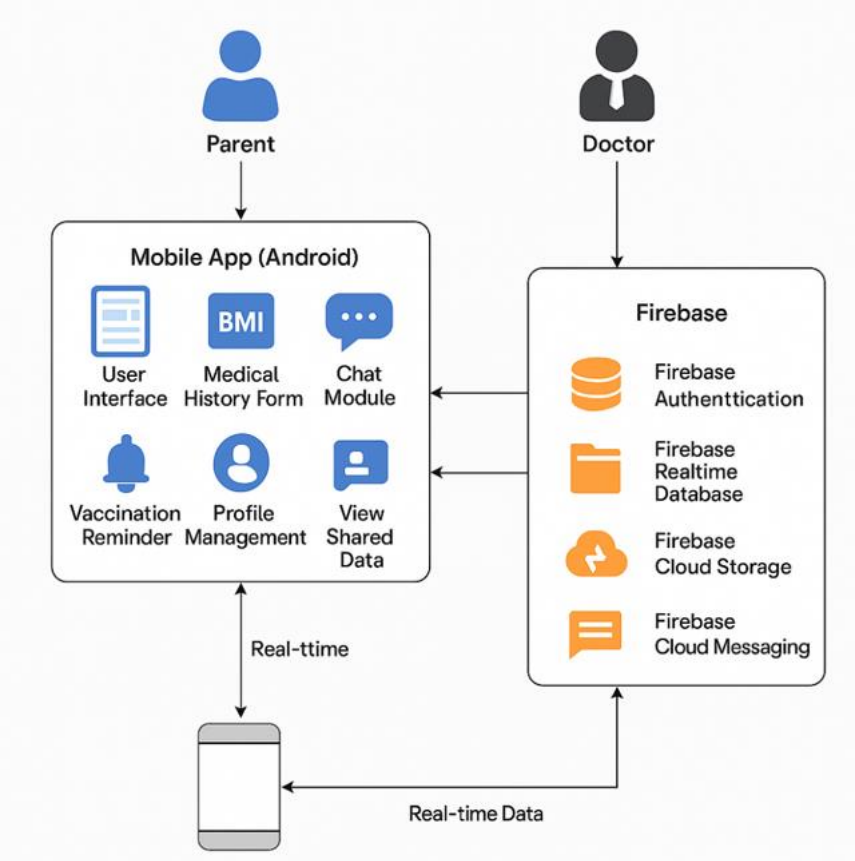


Figure 3.3: Data Flow Diagram

3.6 Use Case Diagram

The use case diagram provides the general view of interaction between all users and CareGrow system. It indicates actors employed here, i.e., Parent and the Doctor, along with their corresponding functionalities accessible by them. It illustrates the functionality offered by these actors to describe system requirements as well as users' expectations.

Parents may add and update child profiles, add medical history, monitor growth statistics, post prescriptions, and message or video call doctors. Doctors may register, view prevalent child health details, and respond to messages or video calls via parents. Such interactions are abstracted in the diagram and provide the basis to determine functional requirements and implementation modules.

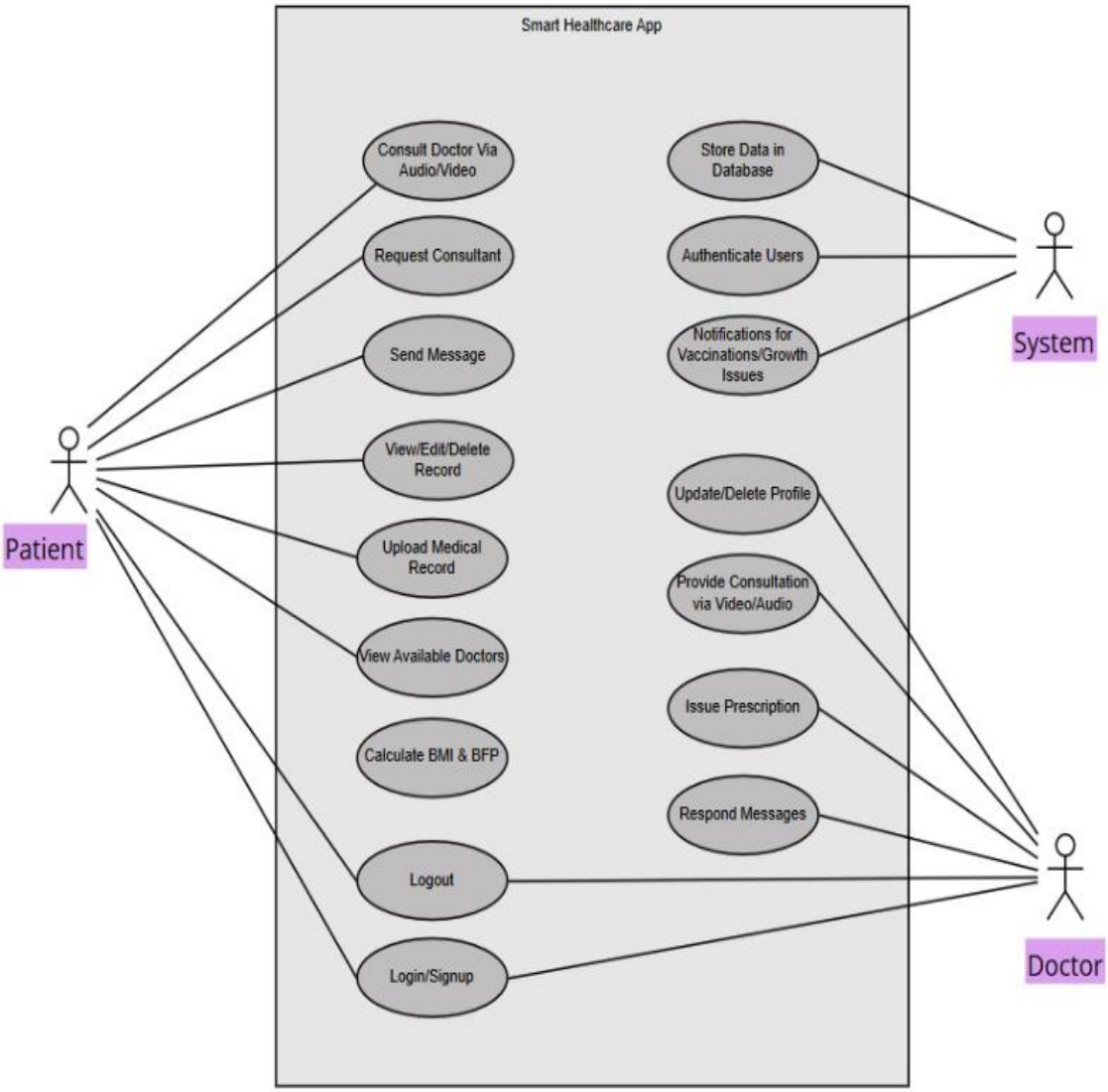


Figure 3.4: Use case Diagram

3.6.1 Use Cases

Table 3.1: Use Case 1

USE CASE ID	**UC-1**
USE CASE NAME	**Home screen / Dashboard**
Actors	**User / Patient**
Description	The system will allow the user to perform functionalities like make profile and add medical record , calculate BMI , contact with doctors.
Trigger	User will click on add record, view records , BMI calculator , available doctors , disclaimer , and contact us buttons
Preconditions	N/A
Post conditions	User can perform different operations like add record , view record, view record , bmi calculation .
Normal Flow	1. Open application 2. Application shows splash screen 3. Show on board fragments 4. Home screen appears
Alternative Flow	N/A
Exceptions	N/A
Includes	N/A
Special Requirement	N/A
Assumptions	User must understand English language
Notes and Issues	N/A

Table 3.2: Use Case 2

USE CASE ID	**UC-2**
USE CASE NAME	**ADD RECORD**
Actors	User/ Patient
Description	The system will allow the user to perform add record with image, name, father name, age , dob, height , weight , blood group , relation, gender
Trigger	User will click on save and add medical record button.
Preconditions	User must reach on home screen and press add record button
Post conditions	User will enter medical record
Normal Flow	1. Open application 2. Show splash screen 3. Home screen 4. Press add record button 5. Add record screen
Alternative Flow	N/A
Exceptions	N/A
Includes	Room persistence integration
Special Requirement	N/A
Assumptions	User must understand English language
Notes and Issues	N/A

Table 3.3: Use Case 3

USE CASE ID	**UC-3**
USE CASE NAME	**MEDICAL RECORD**
Actors	User/ Patient
Description	The system will allow the user to add his medical information with illness type ,medicine , visiting date , prescription ,doctor name , doctor contact number , vaccination type , next vaccination date , previous vaccination date
Trigger	User will click on save button
Preconditions	User must save the basic bio record on add record screen
Post conditions	User will save all record in local database
Normal Flow	1. Open application 2. Shown splash screen 3. Home screen 4. Press add record button 5. Add data on add record screen 6. Click on Save and add medical record button
Alternative Flow	N/A
Exceptions	N/A
Includes	Room persistence integration
Special Requirement	N/A
Assumptions	User must understand English language
Notes and Issues	N/A

Table 3.4: Use Case 4

USE CASE ID	**UC-4**
USE CASE NAME	**VIEW RECORD**
Actors	User / Patient
Description	The system will allow the user to view all record stored in local database
Trigger	User will long press on record button then popup will appear with update and delete
Preconditions	User must save any record
Post conditions	User can see profile , update and delete it
Normal Flow	Open application Shown splash screen Home screen Press view button View record screen
Alternative Flow	N/A
Exceptions	N/A
Includes	Recycler view, Room persistence integration
Special Requirement	N/A
Assumptions	User must understand English language
Notes and Issues	N/A

Table 3.5: Use Case 5

USE CASE ID	**UC-5**
USE CASE NAME	**VIEW COMPLETE RECORD**
Actors	User / patient
Description	The system will allow the user to see the complete profile along medical record
Trigger	User can click back button
Preconditions	User must save any record
Post conditions	User can update or delete any record
Normal Flow	1. Open application 2. Shown splash 3. Home screen 4. Press view record button 5. Press on record button 6. View complete record screen
Alternative Flow	N/A
Exceptions	N/A
Includes	Room persistence integration
Special Requirement	N/A
Assumptions	User must understand English language
Notes and Issues	N/A

Table 3.6: Use Case 6

USE CASE ID	**UC-6**
USE CASE NAME	**UPDATE RECORD**
Actors	User / Patient
Description	The system will allow the user to update their records.
Trigger	User can click on update button
Preconditions	User must save any record
Post conditions	Profile or record will updated
Normal Flow	1. Open application 2. Shown splash 3. Home screen 4. Press view record button 5. Long Press on record 6. Select update record option from dropdown 7. Update profile screen
Alternative Flow	N/A
Exceptions	N/A
Includes	Room persistence integration
Special Requirement	N/A
Assumptions	User must understand English language
Notes and Issues	N/A

Table 3.7: Use Case 7

USE CASE ID	**UC-7**
USE CASE NAME	**DELETE RECORD**
Actors	User / patient
Description	User can delete their records
Trigger	User will click on delete on popup menu
Preconditions	User must save at least a record
Post conditions	N/A
Normal Flow	1. Open application 2. Shown splash 3. Home screen 4. Press view record button 5. Long press on record 6. Select delete record option from popup 7. Record will be deleted
Alternative Flow	N/A
Exceptions	N/A
Includes	Room persistence integration
Special Requirement	N/A
Assumptions	User must understand English language
Notes and Issues	N/A

Table 3.8: Use Case 8

USE CASE ID	**UC-8**
USE CASE NAME	**BMI-BFP CALCULATOR**
Actors	User / patient
Description	User can calculate BMI according to the height , weight then also can calculate BFP on the basis of BMI .
Trigger	User will click on calculate button
Preconditions	N/A
Post conditions	
Normal Flow	1. Open application 2. Shown splash screen 3. Home screen 4. Press BMI calculator category 5. BMI calculator screen
Alternative Flow	N/A
Exceptions	N/A
Includes	Room persistence integration
Special Requirement	N/A
Assumptions	User must understand English language
Notes and Issues	N/A

Table 3.9: Use Case 9

USE CASE ID	**UC-9**
USE CASE NAME	**DOTORS/PATIENT PANEL SIGNUP SCREEN**
Actor	Doctor/Patient
Description	In this screen doctor can make their profile according to name, email, degree, specialization, location, timing, profile picture and password
Trigger	User will click on signup button
Preconditions	N/A
Post conditions	User can login with email and password
Normal Flow	1. Open application 2. Shown splash screen 3. Login screen 4. Click on signup to switch screen 5. Signup screen appear
Alternative Flow	N/A
Exception	N/A
Includes	Firebase authentication, fire base real time database and firebase storage integration.
Special Requirement	Internet must require
Assumptions	User must understand English language
Notes and issues	N/A

Table 3.10: Use Case 10

USE CASE ID	**UC-10**
USE CASE NAME	**DOCTOR'S/PATIENT'S LOGIN**
Actor	Doctor/Patient
Description	The system allows the user to login into the application
Trigger	User will click on the login button
Preconditions	User must have an account
Post conditions	User successfully logged in
Normal Flow	1. Open application 2. Shown splash screen 3. Login screen shown 4. Enter email and password 5. System validates if user have account 6. Login successfully 7. Main activity appears
Alternative Flow	N/A
Exception	N/A
Includes	Firebase authentication
Special Requirement	Internet must require
Assumptions	User must understand English language
Notes and issues	N.A

Table 3.11: Use Case 11

USE CASE ID	**UC-11**
USE CASE NAME	**DOCTOR'S MAIN SCREEN**
Actor	Doctor
Description	The system will allow the user to see his/her complete profile
Trigger	user will click on more option to select update, sign-out, delete options
Preconditions	User must Logged in
Post conditions	User can update profile, sign-out, and delete account
Normal Flow	1. Open application 2. Shown splash screen 3. Login screen shown 4. Login successfully 5. Main screen shown
Alternative Flow	N/A
Exception	N/A
Includes	Firebase authentication, database, firebase storage
Special Requirement	Internet must require
Assumptions	User must understand English Language
Notes and issues	N/A

Table 3.12: Use Case 12

USECASE ID	**UC-12**
USE CASE NAME	**Doctor's update profile**
Actor	Doctor
Description	The application will allow the user to update their profile.
Trigger	User will click on update option
Preconditions	User must log in
Post conditions	Account will be updated
Normal Flow	1. Open application 2. Shown splash 3. Login screen 4. Login successfully 5. Main screen 6. Click on more button 7. Click on update profile screen
Alternative Flow	N/A
Exception	N/A
Includes	Fire base real time database
Special Requirement	Internet must require
Assumptions	User must understand English Language
Notes and issues	N/A

Table 3.13: Use Case 13

USECASE ID	**UC-13**
USE CASE NAME	**DOCTOR SIGNOUT**
Actor	Doctor
Description	The application allow the user to logout successfully
Trigger	User click on sign out option
Preconditions	User must logged in
Post conditions	User will log out successfully
Normal Flow	1. Open application 2. Shown splash 3. Login screen 4. Login successfully 5. Main screen 6. Click on more button 7. User logout successfully
Alternative Flow	N/A
Exception	N/A
Includes	Firebase Authentication
Special Requirement	Internet must require
Assumptions	User must understand English language
Notes and issues	N/A

Table 3.14: Use Case 14

USECASE ID	**UC-14**
USE CASE NAME	**DOCTOR DELETE ACCOUNT**
Actor	Doctor
Description	System allow the user to delete account
Trigger	User will click on delete account option
Preconditions	User must log in
Post conditions	User will successfully delete his/her account and data
Normal Flow	Open application Shown splash screen Login successfully Main screen Click on more button Click on delete account User delete account successfully
Alternative Flow	N/A
Exception	N/A
Includes	Firebase authentication
Special Requirement	N/A
Assumptions	User must understand English language
Notes and issues	N/A

Table 3.15: Use Case 15

USECASE ID	**UC-15**
USE CASE NAME	**DOCTORS AVAILABLE**
Actor	User / patient
Description	User can see all available doctors and make contact with doctors through call and message
Trigger	User will click on call and message button
Preconditions	Doctor must have an account in doctors' panel
Post conditions	User can make contact with doctor's
Normal Flow	1. Open application 2. Show splash screen 3. Show home screen 4. Press doctor's available category 5. Doctor available screen 6. Make call by pressing call option 7. Make text by pressing text option
Alternative Flow	N/A
Exception	N/A
Includes	Fire base real time database integration.
Special Requirement	N/A
Assumptions	User must have sim network
Notes and issues	N/A

Table 3.16: Use Case 16

USECASE ID	**UC-16**
USE CASE NAME	**ADUIO/VEDIO CALL**
Actor	User / patient
Description	User can see all available doctors and make contact with doctors through call and message
Trigger	User will click on call and message button
Preconditions	Doctor must have an account in doctors' panel
Post conditions	User can make contact with doctor's
Normal Flow	8. Open application 9. Show splash screen 10. Show home screen 11. Press doctor's available category 12. Doctor available screen 13. Make call by pressing call option 14. Make text by pressing text option
Alternative Flow	N/A
Exception	N/A
Includes	Fire base real time database integration.
Special Requirement	N/A
Assumptions	User must have sim network
Notes and issues	N/A

CHAPTER 4

DATA AND EXPERIMENTS (and/or IMPLEMENTATION)

4.1 Frontend Development

The frontend of the child-oriented healthcare application is coded in Java and XML within Android Studio, adhering to Material Design principles for a sleek and user-friendly interface. The general aim of the frontend is to give users an easy, intuitive, and responsive experience—majorly parents who are not computer literate. All the screens are designed with extreme care such that the navigation is easy and important health information is presented in a clear manner.

The application starts with a signup and login page, wherein the users can securely sign up or login as shown in Figure 4.1. The section also features role-based login along with parent and physician segregation.

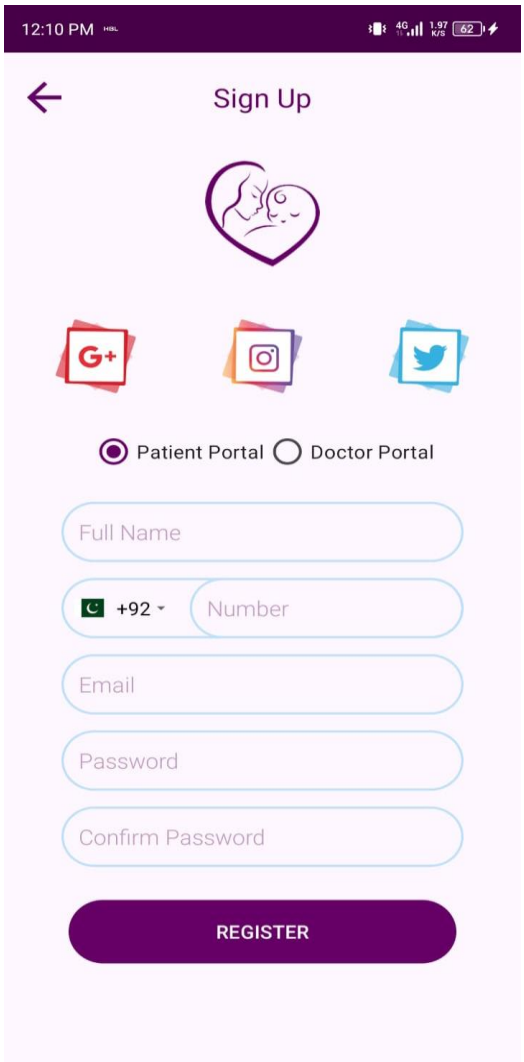


Figure 4.1: User Login and Registration Screen(user)

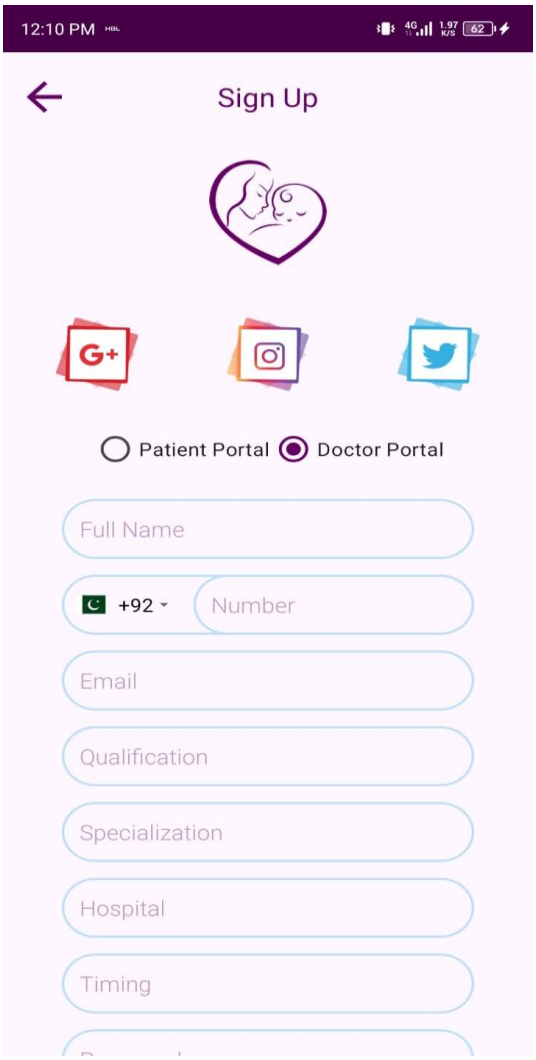


Figure 4.2: User Login and Registration Screen(Doctor)

Firebase Authentication has been used to validate user credentials and establish secure access for the individual data. Once the user logs in, he/she is directed towards the central dashboard, which is the nucleus of all the health management operations.

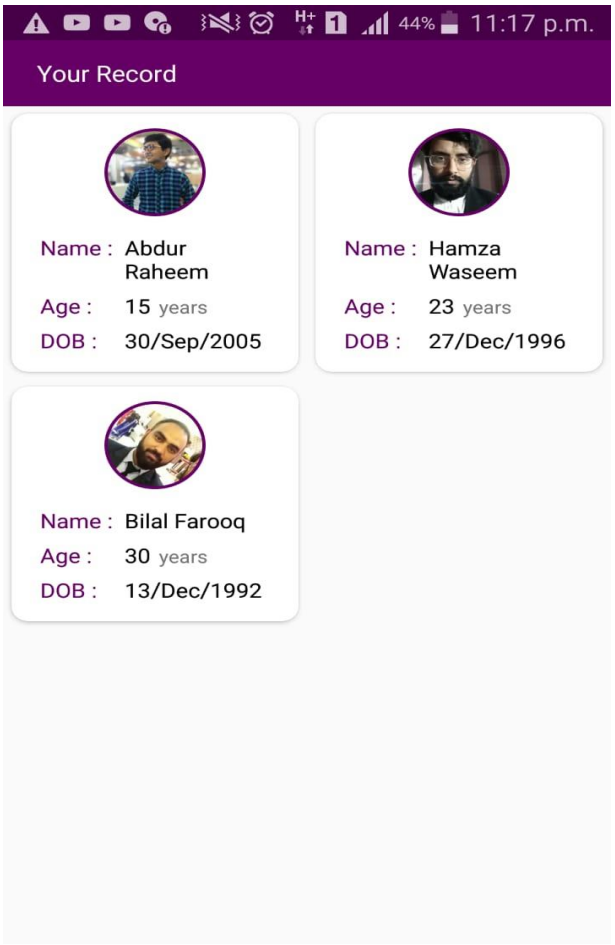


Figure 4.3: Child Profile Card

On the dashboard, parents may view a collection of child profiles they have setup. Each one is displayed in the form of a card containing the child's name, age, and their date of birth as in Figure 4.2. The user may click any card to review detailed sections such as medical record, vaccination updates, growth summary, and added prescriptions. These profiles can be inserted or edited from a special template, which is created to accommodate filling in information such as the name of the child, birthdate, and sex. Input such as a profile picture, if offered on an optional basis, contributes to it to be more user-friendly and personalized.

The vaccinations screen shows the list of completed and pending vaccinations for every child. Dates get compared automatically with the current calendar, and Firebase Cloud Messaging (FCM) is employed to send a reminder to the parent's device if a vaccination is pending. Reminders appear in the notification center as well as within the app for convenience.

For tracking height and weight to utilize for growth and BMI, the application accepts input for weight and height measurements. As show in Figure 4.3 It computes the Body Mass Index (BMI) and Body Fat Percentage based on the formula of the standard BMI, BFP and graphically shows the outcome as color-coded indicators of health like overweight, normal, or underweight.

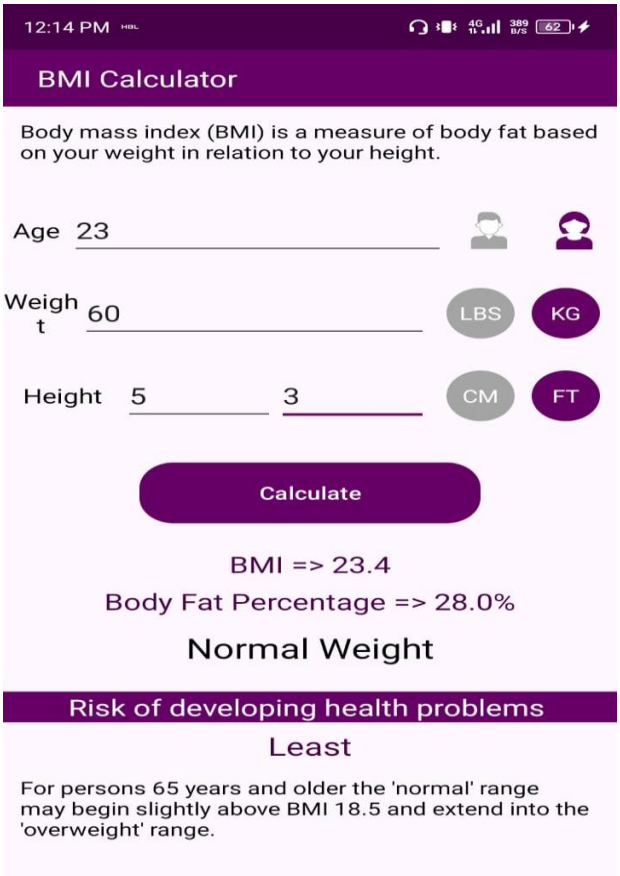


Figure 4.4: BMI and BFP Calculation

The app also enhances this feature by comparing it with age-based child growth charts that are suggested by health agencies. Growth development is represented graphically using a charting library such as MPAndroidChart so that parents can visualize over time.

One of the features that are quite interesting is uploading prescriptions or health records using the camera or gallery of the device. The uploaded pictures are safely stored in Firebase Cloud Storage, and the image links are saved in the child's profile in the Realtime Database. Storing and retrieving vital health records is easier for parents, thus a convenient experience.

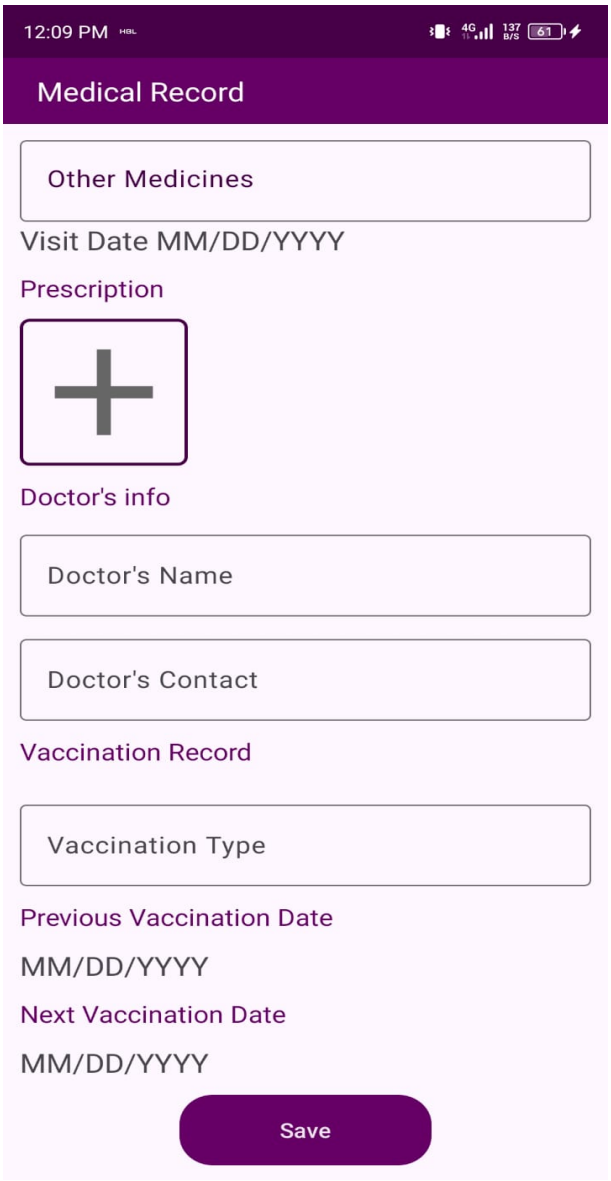


Figure 4.5: Upload prescription and vaccination record

Overall, the frontend of the application at all times maintains ease, simplicity, and usability in all things while keeping the most important functionalities within the users' reach. The direction of the design is focused on helping parents have an instrument that they know intuitively and trust and hence be able to take better care of their children.

4.2 Backend Development

Google Firebase is the health app's cloud backend platform that offers real-time database services, user authentication, cloud storage, and push notifications. Firebase is used because it integrates well with Android Studio, has good scalability, is easy to use, and has robust mobile app features. Its managed environment gives the feature team more bandwidth to code without backend setup or server maintenance to handle.

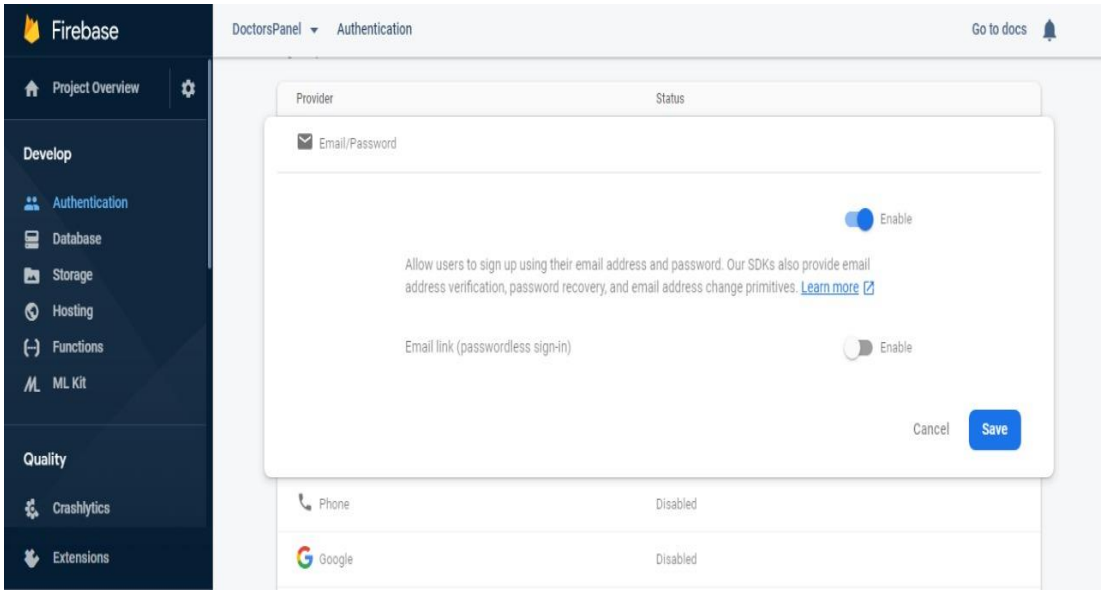


Figure 4.6: Firebase Design

The application is developed upon Firebase Authentication for secure user registration and login management. Email and password authentication as well as sole lawful access of lawful users to sensitive data only is supported. Two roles, Doctor and Parent, are held in the system. Role-based access control is achieved in a way so as to enable the users to view and manage data of their respective role. For instance, parents can have child accounts and add medical history and physicians can view only what the users have uploaded to them and add consult notes.

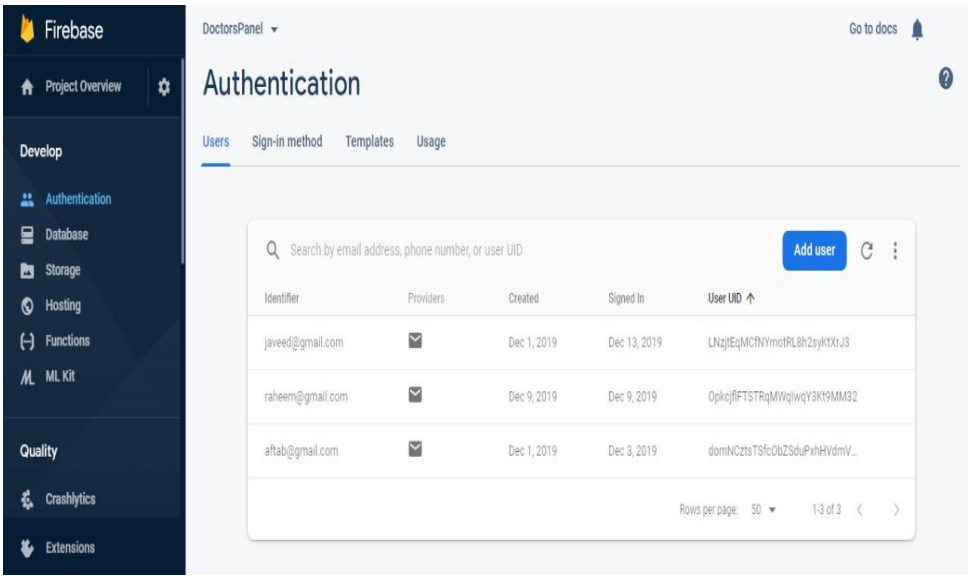


Figure 4.7: Firebase Authentication

Data storage at the center level is managed through Firebase's Realtime Database with hierarchical, JSON-form-type data structures. The child profiles, medical history, immunization status, and growth entries are stored as database nodes. It is simple to enable real-time synchronization between the user interface and database. Whatever user change is being made—altering a growth measure or adding a prescription—is synchronized automatically within the app without requiring the user to issue a refresh.

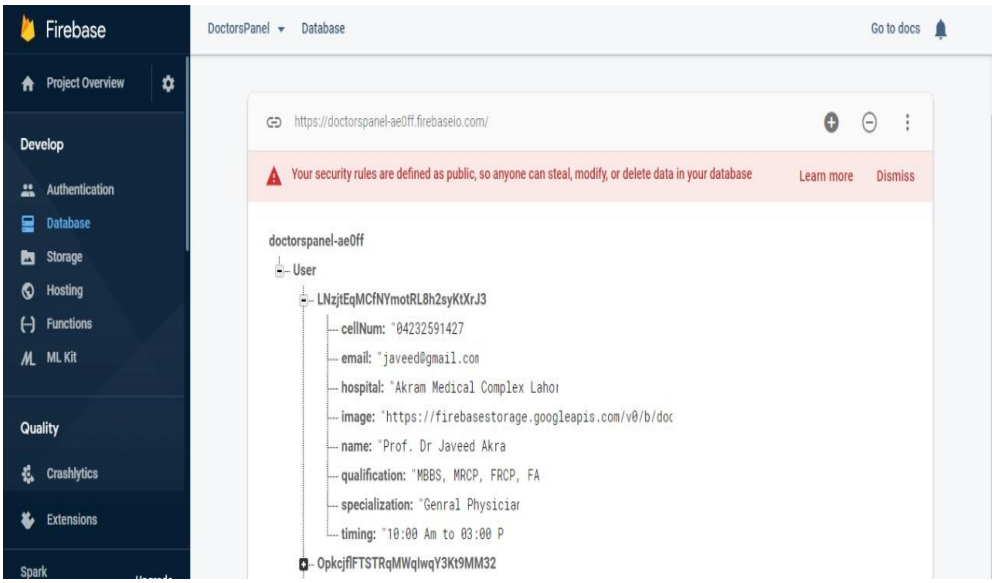


Figure 4.8: Firebase Design

The backend also leverages Firebase Cloud Storage to securely store media files such as prescription images and other health information. The image is stored securely in a Firebase bucket when a parent uploads it, with an associated public download URL generated. The download URL is then linked to the corresponding child's profile in the Realtime Database. Separation of metadata (in the database) and binary data (in storage) is more performance- and scalability-efficient for large files.

For reminders and notifications, Firebase Cloud Messaging (FCM) is implemented in the app. Reminders are made on future vaccination schedule dates, delayed growth logging, or registration success. All these reminders are posted in real-time, and it doesn't matter whether the app is in foreground status or background status, so that parents are reminded repeatedly about health milestones.

Following is a conceptual data model of Firebase:

- Users node stores login and role data (email, name, type: doctor/parent).
- Children node stores children data for the parent user ID.
- Vaccinations node stores a list of vaccine entries each with due date, completion status, and reminders.
- GrowthMetrics node stores height and weight entries dated from which the BMI is computed.
- Prescriptions node stores Cloud Storage photos and has notes.

Firebase is offline-enabled as well, and the application therefore continues working in the event of continuous loss of network connectivity. Upon reestablishment of connectivity, Firebase synchronizes locally made updates from the cloud database automatically. It proves useful for parents under whom internet connectivity is patchy.

Security-wise, Firebase Security Rules are defined on each node restricting access on UID and the logged-in user's role. A parent, for instance, can read from and write to her own user ID, while a doctor can read what has been written by a parent with him

once he is authorized. All data communication is done over HTTPS and access is well regulated through Firebase's rule-based system.

Briefly, Firebase backend infrastructure delivers fast, secure, and stable data administration for securely saving health records and readily available. It offers low development overhead and time and high-performance capabilities adequate for a child-focused healthcare app.

4.3 Implementation of Key Features

This subsection presents the essential features of the healthcare app that render it best suited for monitoring children's health profiles. The features are obtained by combining Firebase services and Android features to facilitate real-time operations and scalability. The features aim at improving accessibility, reliability, and automation of day-to-day health monitoring.

4.3.1 Medical History Monitoring

The app allows parents to maintain the history of a child's treatments, drugs given, diagnosis, and physician visits. This is stored on a per child's individual ID in Firebase Realtime Database. Parents utilize a pre-formatted format to create new entries and edit or delete old ones as required. The electronic diary eliminates the agro of manual recording and has previous health records available if needed by doctors.

4.3.2 Immunization Reminders

The key characteristic of the software is vaccine control. Each profile of a child has a vaccination plan with pre-defined paediatric time intervals by default. Once date of birth is entered, the system automatically defaults to calculate due dates for vaccinations using pre-defined rules.

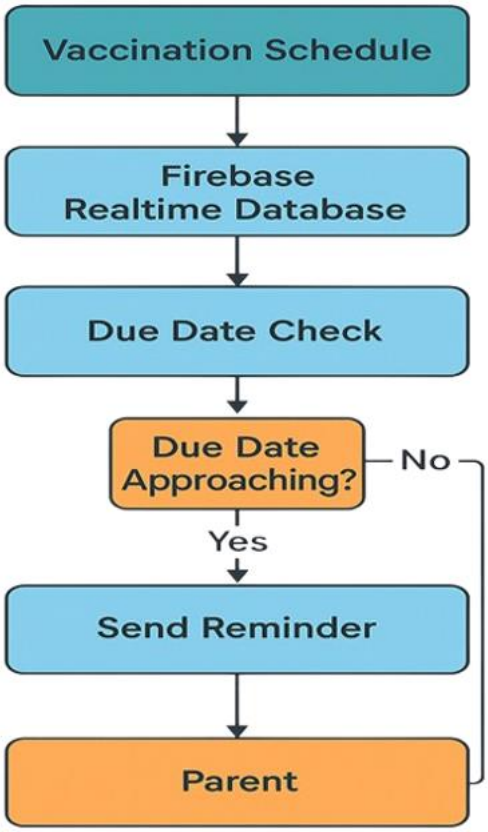


Figure 4.9: Vaccination Timeline and Reminder Notification Flow

Immunization records are stored in Firebase and tracked using background logic. Firebase Cloud Messaging (FCM) makes sending reminders to parents prior to due dates easy. Alarms display on the app user interface and the mobile alert to avoid missing critical health milestones by the user.

4.3.3 Growth Monitoring BMI and BFP Calculation

The application provides daily input of parents' recording of the child's height and weight. With this in mind, the BMI (Body Mass Index) and BFP (Body Fat Percentage) must automatically be calculated using the general formula:

For BMI:

$$BMI = \text{weight (kg)} / \text{height}^2 (m^2)$$

Table 4.1: BMI Categories [6]

BMI Range	Category
<14	Underweight
14-18	Normal Weight
>18	Overweight

For BFP:

BFP = (Total Fat Mass / Total Body Mass) *100

Table 4.2: BFP Categories [6]

BFP Range (for men)	BFP Range (for women)	Category
2-5%	10-13%	Essential fat
6-13%	14-20%	Athletes
14-17%	21-24%	Fitness
18-24%	25-31%	Average
25%+	32%+	Obese

The computed BMI and BFP is compared against WHO growth charts to identify whether the child is underweight, normal, or overweight. The outcome is color-coded for ease and over time by graphical charts. The graphical method makes it easier for parents to identify trends and act when necessary.

4.3.4 Prescription Uploads

Parents can post pictures of prescriptions or of individual child-specific medical history. The app stores such pictures securely through the Firebase Cloud Storage API and stores the URL of each uploaded picture in the database against the entry of the corresponding child. The app displays thumbnails and allows one to look at pictures full-size by a click.

This maintains all of them in one place and within easy reach as and when needed—whether for visit to another doctor or follow-up visit.

4.3.5 Doctor Account and Patient Interaction

Physicians will be registered as healthcare providers via a customized registration process. Physicians, subject to approval, will be able to view child summaries submitted by the parent (permissions-wise), making it easy to give informed consultations. A messaging feature (in later versions) will allow easy two-way physician-parent messaging.

CHAPTER 5

RESULTS AND DISCUSSIONS (or USER MANUAL)

5.1 System Walkthrough (Parent-Doctor Manual)

This is an extensive parent-doctor guide for parents and doctors on the usage and handling of the functions of the application. The application is easy to use and controls all the features needed by a simplified step-by-step process.

5.1.1 Splash and Intro Screens

When the application is launched, the users are presented with a Splash Screen with the logo and tagline of the app (e.g., "Smart Health for Growing Kids"). The screen is shown for a brief period of time while the application is being loaded.

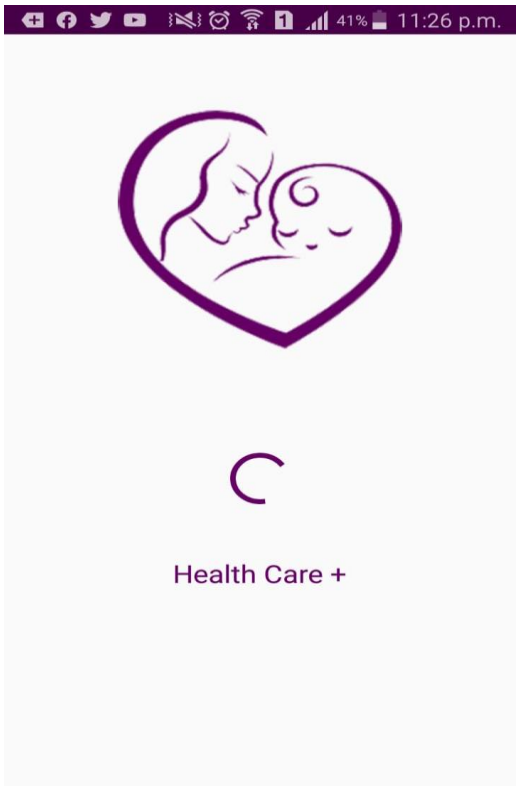


Figure 5.1: Splash Screen-1

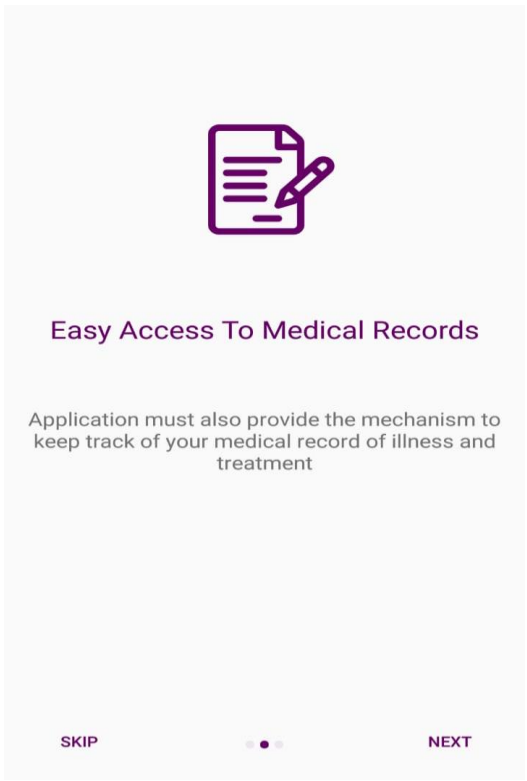


Figure 5.2: Splash Screen-2

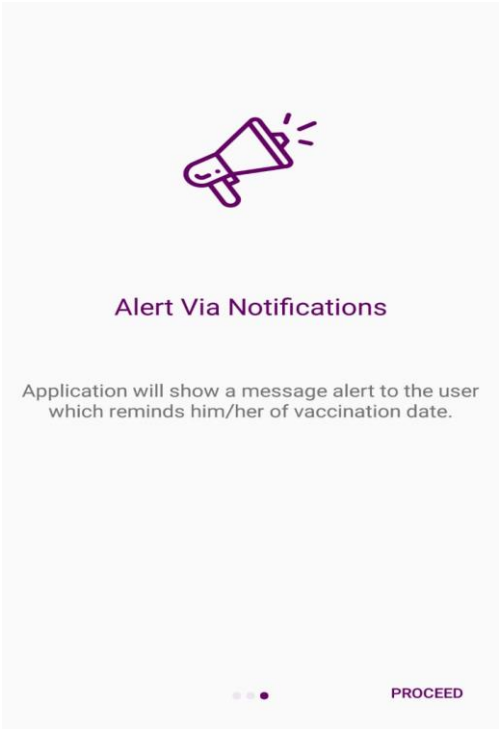


Figure 5.3: Splash Screen-3

Following the splash screen, there appears an Intro Slider or Onboarding Screen (for the first time, for newcomers). It includes 2–3 as shown in figure 5.1, figure 5.2, figure 5.3 slides where one discovers the core features of the app—for instance, managing profile, keeping track of vaccines, and keeping tabs on the BMI. Setting expectations and easing newcomers here in one manner of it.

5.1.2 Home Screen (Dashboard)

Upon successful login, the user is taken to the Home Screen, which serves as the application's central hub. For parents, all child profiles entered and quick access icons for:

- Add New Records
- View all records
- Growth and BMI view
- Available Doctors
- Disclaimer
- Contact Us

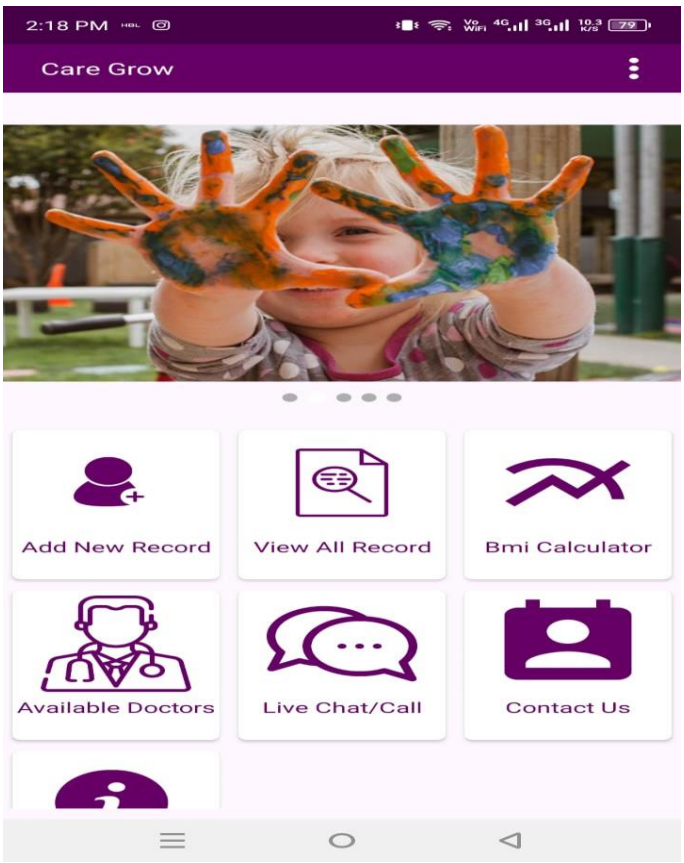


Figure 5.4: Dashboard

5.1.3 User Registration and Login

Parents and doctors can register by filling out a small form requesting their email address, password, and type (parent/doctor). After successful registration, users are redirected to their own dashboard.

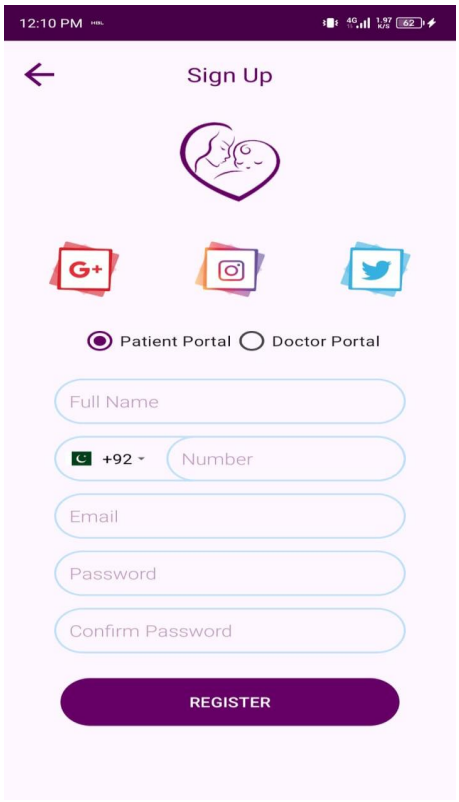


Figure 5.5: Signup(User)

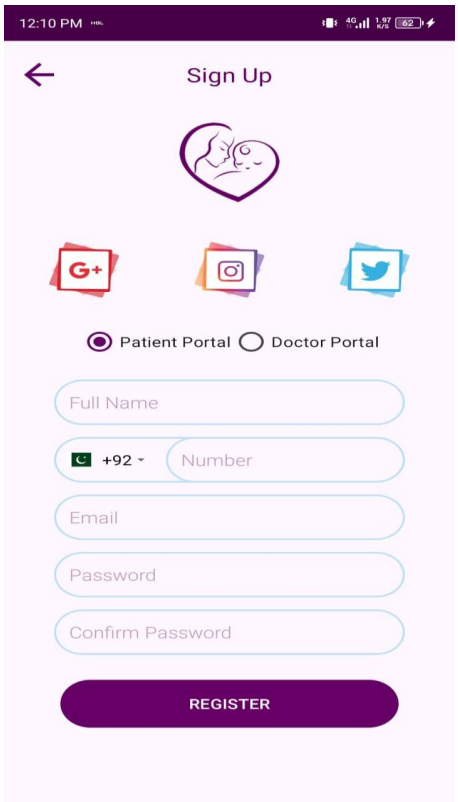


Figure 5.6: Signup(Doctor)

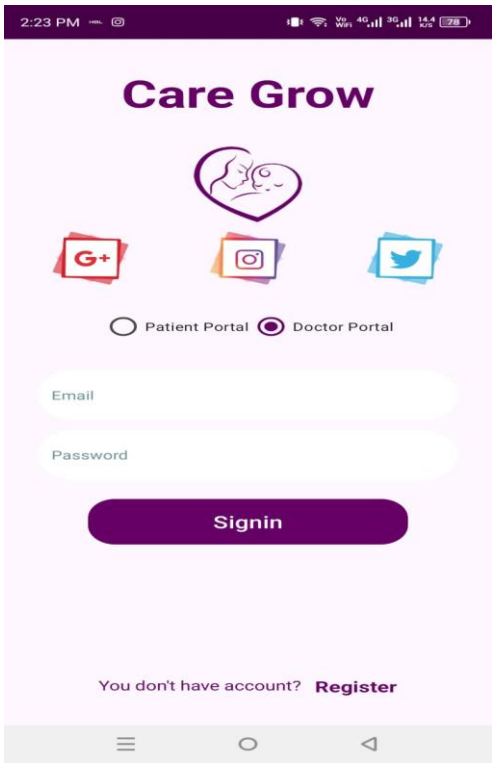


Figure 5.7: Signin(Doctor)

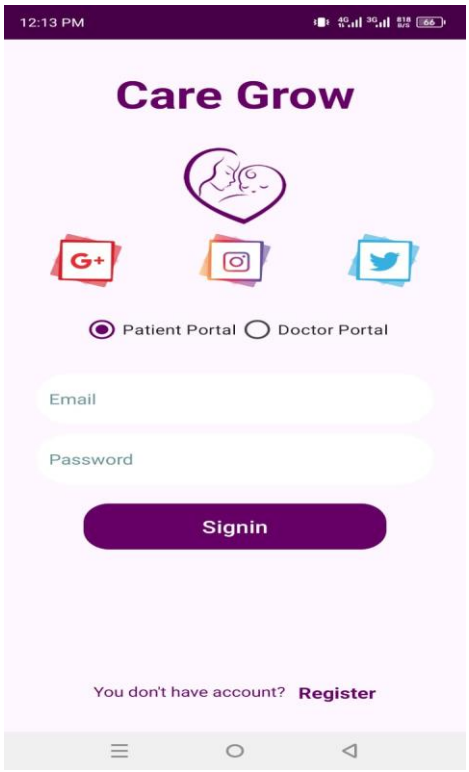


Figure 5.8: Signin(User)

For parents: Profiles of their child are displayed on the dashboard.
For doctors: The dashboard displays consultation features (view-only at present).

5.1.4 Adding and Managing Child Profiles

Parents can add child profiles by providing basic details like name, date of birth, and gender. All the profiles are stored in Firebase against the parent's account as in figure 5.9.

Figure 5.9: Add child dertail

Edit and delete options are available for every profile.

Individual profiles are used to view growth records, medical history as show in firgue 5.10.

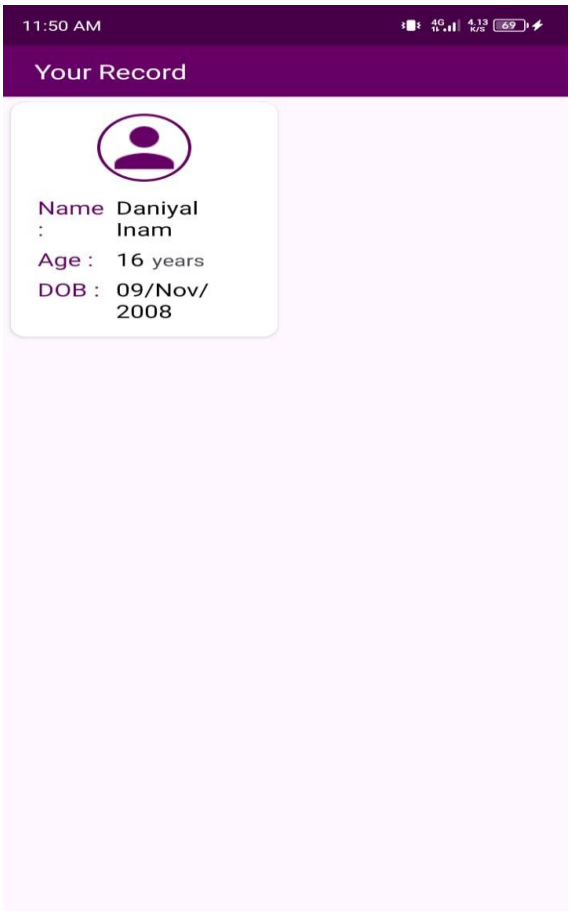


Figure 5.10: View records

5.1.5 Adding Medical History

Users can monitor doctor appointments, symptoms, medicine prescribed, and medicine taken by the child. A medical history timeline is created for every child.

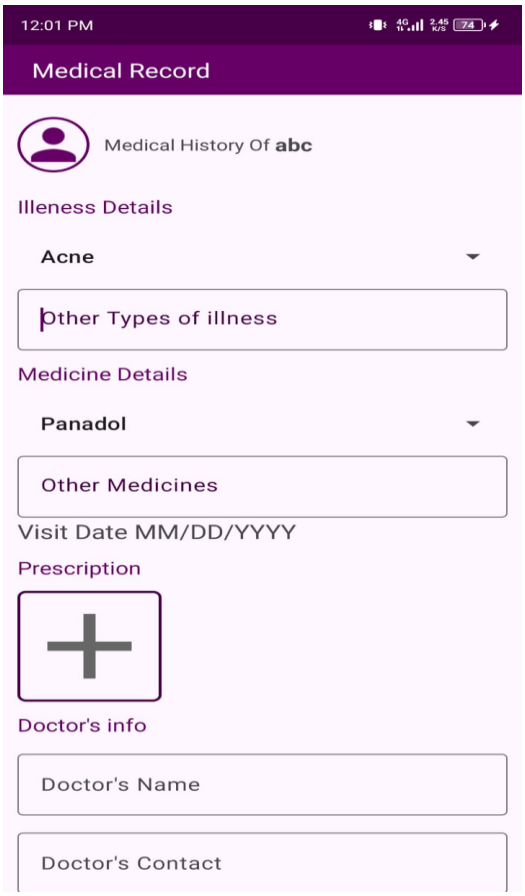


Figure 5.11: Add medical history

It is stored safely with dates.

It can be deleted or modified at any moment.

5.1.6 Tracking for Vaccines

While the child's profile is being constructed, automatically a vaccine schedule as per common paediatric practice is constructed as well. Parents are reminded before each due date.

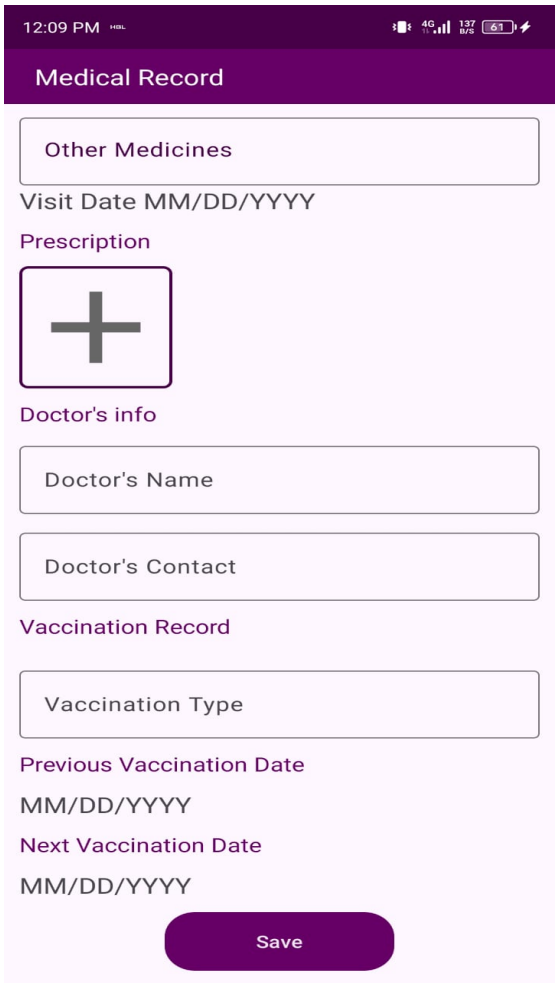


Figure 5.12: Add vaccination

All vaccinations can be finished.

Missed vaccines create follow-up reminders as shown in figure 5.13.

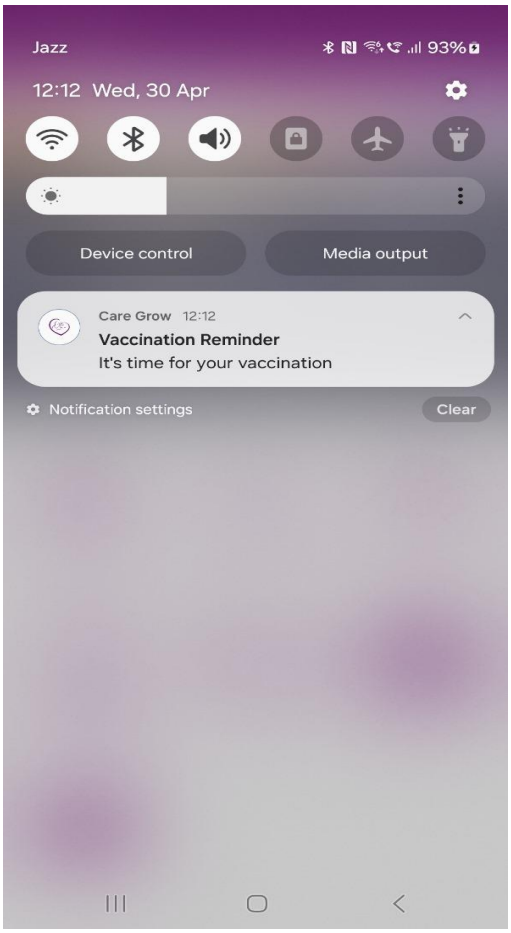


Figure 5.13: Vaccination Alert

5.1.7 Growth and BMI Monitoring

Growth and BMI are quantifiable.

Height and weight data can be entered by parents at intervals. BMI is computed by the app and displayed in color-coded ranges. In the long term, the app generates a growth chart.

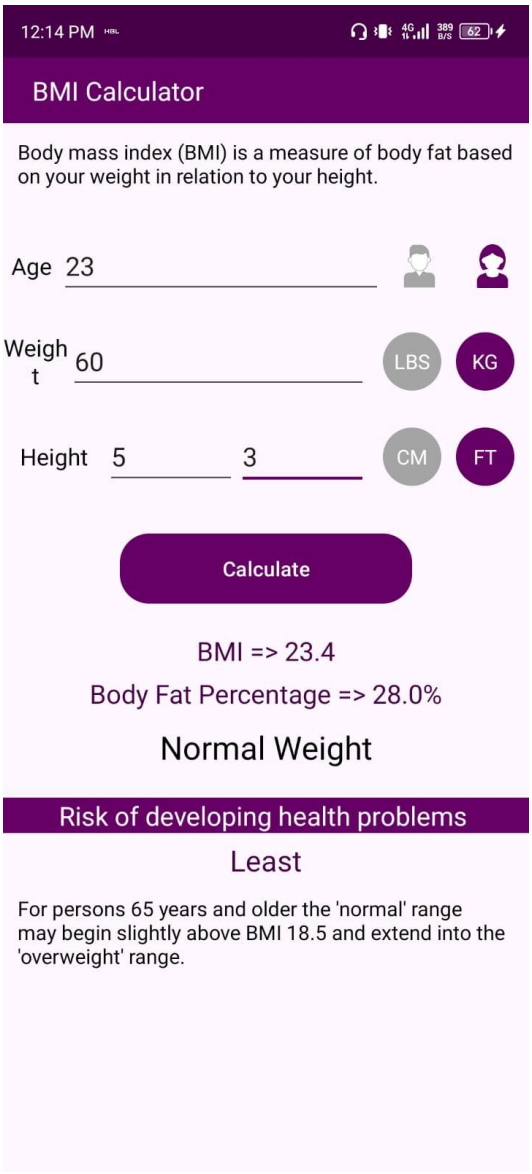


Figure 5.14: BMI

Data entry is made easy and verified.

Health trends are plotted.

5.1.8 Prescription or report upload

The application provides upload of reports or prescriptions using the device gallery or camera as shown in figure 5.15.

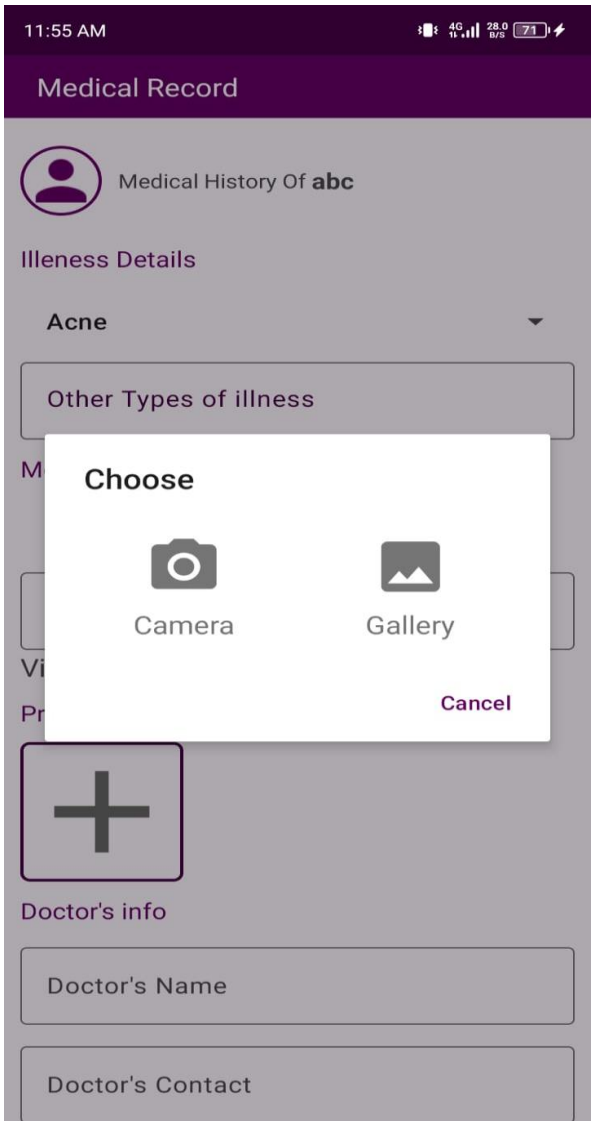


Figure 5.15: Upload Prescription

Images uploaded are stored in Firebase Cloud Storage.
The images are attached to the profile of the corresponding child.

5.1.9 View Available Doctors

Parents will be able to view list of registered doctors on clicking "Available Doctors" navigation menu or from home screen. The feature is displaying doctor profiles in introduction, availability status, specialty, and name. Doctor profiles are obtained from Firebase database, and original accounts only would be displayed on user's dashboard.

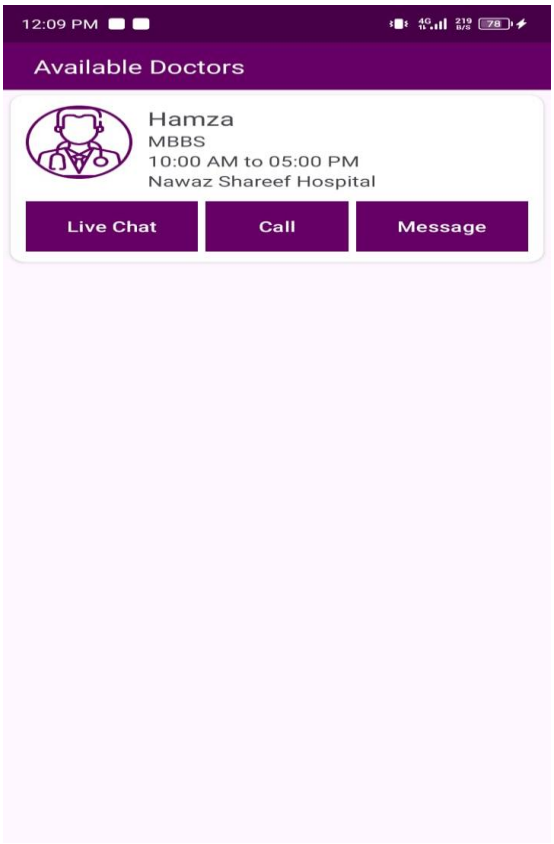


Figure 5.16: Available Doctors

Parents can tap on any card of a doctor to show an information-rich view, including the ability to start a chat or ask for a video consultation.

5.1.10 Live Chat with Doctor

CareGrow offers real-time text-based interaction between the doctors and parents. Parents can send health-related questions, order past consultations, or send symptoms about the child through this facility. The messages are pushed on Firebase's real-time database and made visible at both ends in real-time.

Figure 5.17: Chat with Doctor

The window is minimal in size and contains timestamps, user markers (parent/doctor) as well as silky smooth scrolling for old chats. All the old chats are stored under parent-doctor-child thread for future use.

5.1.11 Video Calling to Doctor

In addition to chat, CareGrow also has video calling of physicians for prolonged consultation. This proves particularly useful with conditions that demand word-based description, instant visual inspection, or when chat-based advice will not do.

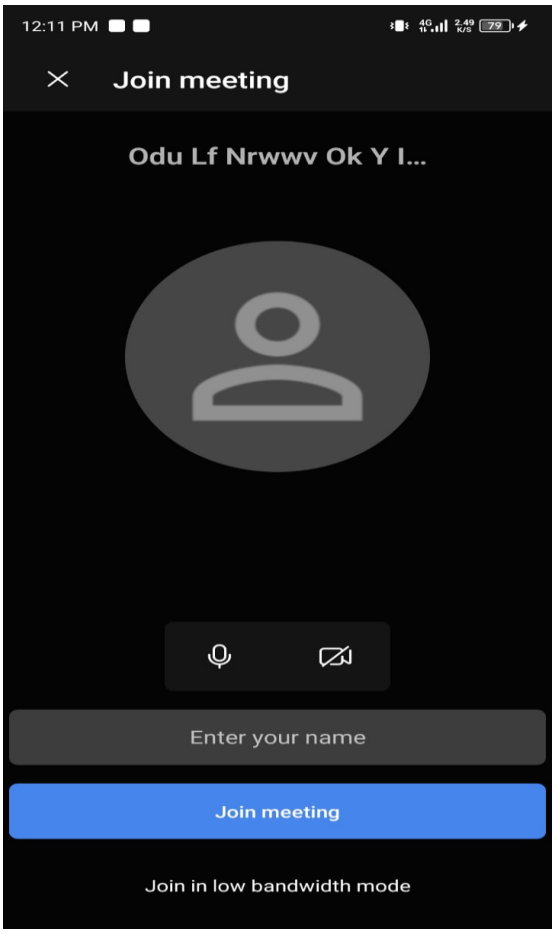


Figure 5.18: Video call with Doctor

The video calling feature is offered through a third-party API like Jitsi Meet or Agora, embedded in the app UI. The call will only be initiated after being accepted by the other party. The feature can also be offered with an appointment or request-based model of availability.

.

5.2 Testing Results

In order to determine that the health care application is running, useful, and trustworthy, various levels of tests were conducted. They are unit testing, integration testing, and UI testing on real devices and emulators. They were aimed at ensuring everything that has been created is running as desired, producing good output, and producing a smooth interface on various configurations of devices.

5.2.1 Type of Testing

1. Unit Testing:
Every operation and module, i.e., calculation of BMI, growth input validation, and Firebase uploading of data, were separately tested using JUnit as well as Logcat logs.

2. Integration Testing:
End-to-end user flows were comprehensively tested, i.e.,
Register → Login → Create Profile → Enter Medical History
Upload Prescription → View Child Growth → Receive Vaccination Reminder
It confirmed passing of data properly between screens and Firebase back-end.

3. User Interface (UI) Testing:
UI components were made responsive, layout organization was done the same way across devices with different screen sizes, and interactive components like buttons, input fields, and notification cards.

4. Device Compatibility Testing:
The app was tested on several devices ranging from Android version API 21 (Lollipop) to API 33 (Android 13), so the layout, functionality, and Firebase connections are all fine on different screen size devices and performance.

5.2.2 Test Case Summary

The following is the summary table for the main features and their respective test cases, expected and actual results.

Table 5.1 Test Case DC 1 (Signup Screen)

Preconditions	N/A
Actions	➢ Login with correct email/password ➢ Login with incorrect email address ➢ Login with incorrect password
Expected Results	➢ Navigate to Doctor Profile ➢ please Register first ➢ Please Register First
Status	Passed

Table 5.2 Test Case DC 2 (Login Screen)

Preconditions	User must have an Account
Actions	➢ Login with correct email/password ➢ Login with incorrect email address ➢ Login with incorrect password
Expected Results	➢ Navigate to Doctor Profile ➢ please Register first ➢ Please Register First
Status	Passed

Table 5.3 Test Case 3 (Doctor panel)

Preconditions	User must log in
Actions	➢ Delete doctors Profile ➢ Update doctors' profile ➢ Sign-out from doctors' profile
Expected Results	➢ Successfully deleted doctors' profile ➢ Successfully updated doctors' profile Successfully sign-out from doctors' profile and navigate to sign in screen
Status	Passed

Table 5.4 Test case 4 (Home screen)

Preconditions	N/A
Actions	➢ Show BMI calculator ➢ show add new record ➢ Show view all record ➢ Show available doctors ➢ Show disclaimer ➢ Show contact us
Expected Results	➢ Successfully navigate to BMI calculator screen ➢ Successfully navigate to add new record screen ➢ Successfully navigate to view all record screen ➢ Successfully navigate to available doctors ➢ Successfully navigate to about screen ➢ Successfully navigate to contact us screen
Status	Passed

Table 5.5 Test case 5 (BMI Calculator)

Preconditions	User must calculate BMI
Actions	➢ Calculate BMI
Expected Results	➢ Successfully calculated BMI according weight, height
Status	Passed

Table 5.6 Test case 6 (BFP Calculator)

Preconditions	User must calculate BMI
Actions	➢ Calculate BFP
Expected Results	➢ Successfully calculated BFP on the basis of BMI
Status	Passed

Table 5.7 Test case 7 (Add new record)

Preconditions	Local Database configured
Actions	➢ Add Bio Data of user ➢ Any field missing in bio-data screen ➢ when user press saves and medical record button with any empty field ➢ when user press saves and medical record button with filled fields
Expected Results	➢ Successfully add bio-data and navigate to Medical record Screen ➢ Error shown on missing field and didn't navigate to Medical record screen ➢ showing appropriate error ➢ Successfully add bio-data and navigate to Medical record Screen
Status	Passed

Table 5.8 Test case 8 (Medical Record Screen)

Preconditions	Local Database configured and enter personal details
Actions	➢ Add Medical Record ➢ Save medical record with all empty fields ➢ when user press save button with filled fields ➢ when user press save button with any empty field
Expected Results	➢ User enter the medical record and save into local storage then Navigate to Home screen ➢ Error shown on missing field and didn't save record in database ➢ User enter the medical record and save into local storage then Navigate to Home screen ➢ Error shown on missing field and didn't save record in database
Status	Passed

Table 5.9 Test case 9 (View All record)

Preconditions	User must add any record
Actions	➢ View all users ➢ View complete profile of user ➢ on click any record
Expected Results	➢ User can view all record ➢ Profile shown successfully ➢ It will navigate to user profile
Status	Passed

Table 5.10 Test case 10 (Available doctors)

Preconditions	Doctors must make account on Doctor's Panel
Actions	➢ Showing list of all available doctors ➢ User can call the doctor by pressing call button ➢ User can message the doctor by pressing message button
Expected Results	➢ Doctors list shown successfully ➢ Navigate to cell phone dialler along with doctor contact number ➢ Navigate to cell phone message box
Status	Passed

Table 5.11 Test case 11 (Contact us)

Preconditions	N/A
Actions	➢ Show the details of developer ➢ Onclick report bug click here button
Expected Results	➢ Developers details shown successfully ➢ Navigate to Gmail for mail the bug
Status	Passed

5.2.3 Observations and Results

Features behaved as expected on devices and test cases. Firebase integration (Realtime Database, Storage, and Authentication) was good with no apparent latency. UI responded and behaved the same way on devices with varying resolutions. Notifications were sent through FCM instantly in emulator and real device testing. Short UI tics (e.g., label overlap) were fixed in retest.

The application was subjected to all the functional and usability testing conducted during the test cycle. Results of tests show that the system is deployable, easy to use, and stable. Stress testing and performance optimization will be addressed in subsequent cycles, particularly if the user community goes haywire.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Summary

The aim of the project was to create a basic mobile health application using which parents would be able to maintain the health records of their children, monitor growth, note down vaccination, and get advice from medical physicians.

During the development phase, the application was developed with maximum ease of usage, data security, and live access. The features offered include child profile management, capturing of medical history, tracking of vaccines along with reminders, BMI calculation along with monitoring of growth, uploading prescription photographs, and live physician chat.

With Firebase products such as Authentication, Realtime Database, Cloud Storage, and Cloud Messaging, the backend infrastructure continues to be powerful, scalable, and data-secure. Frontend is developed on Android Studio (Java/XML) as per Material Design guidelines for a regular and uniform user interface.

All the functional bases were tried and were operational on a variety of various Android platforms. First-user testing feedback also supported the usability, practicability, and applicability of the application for everyday child health care.

6.2 Summary of the Challenges

While developing and testing the application, several challenges were encountered:

- **Firebase Data Structure**: Creating an incredibly efficient and programmable data structure of a database that would be capable of supporting multiple users, child records, and instant data access without any performance delays to the application.

- **Real-time Chat Integration**: Between maintaining live chat performance consistent in between parent and physician users in different network statuses.

- **UI Responsiveness**: UI responsiveness across various varied screen sizes and resolutions taking place in Android.

- **Security Management**: Suggesting Firebase security rules wisely in a way so that sensitive user data is not leaked but do not block when necessary for doctors.

- **Timing of Notification**: Making use of Firebase Cloud Messaging so that reminders for vaccination were triggered exactly by date.

With all these issues, all of the technical problems were corrected easily by virtue of extensive testing, tuning, and utilization of reliable cloud facilities.

6.3 Suggestion for Future Work

Over the period of the next two years, mainly two categories of features would be developed in CareGrow system; a) conventional facility, b) futuristic AI based features. Details of the proposed enhancement are as follows:

- **AI Assistive Agent**: Having an AI-powered virtual agent to help patients and physicians with utilization of the app, respond to medical inquiries, and help with appointment schedules and medication treatments.

- **AI Recommender System:** Integration of Smart suggestion engine to recommend doctors, services, or health articles to the user depending on user preference, symptoms, and medical history.

- **AI-Based Prescription Reader & History Manager:** AI-based feature to read and interpret the scanned prescription to process the medical history of the patient in a structured way to enhance diagnosis and continuity of care.

- **Social platform based Authentication Options**

Twitter and Instagram Login: Signup will also. be facilitated by embracing social login features through Twitter and Instagram APIs, where users will be able to sign up and log in at the same time.

All the above-mentioned updates in the future. are. going to make CareGrow intelligent, simple to use, and inclusive to the core, as. according. to today's modern-day modern healthcare app requirements and user experience needs.

6.3.1 Conclusion

In general, the project has met desired objectives and provides an integrated solution for electronic child health care management. The development process of this healthcare mobile application has been a learning experience that amalgamates the process of developing a mobile application, cloud computing, and actual healthcare needs. With the functionality of providing an option to save medical history electronically, the project is making healthy contributions towards health awareness, early medical interventions, and low record management stress. This project is well placed for further development and can expand even more to a large platform that unites families and health systems even further.

REFERENCES

Books:

[1] Griffiths, D., & Griffiths, D. (2021). *Head first Android development*. " O'Reilly Media, Inc.".

Journal Papers:

[2] Boulos, M. N. K., Brewer, A. C., Karimkhani, C., Buller, D. B., & Dellavalle, R. P. (2014). Mobile medical and health apps: state of the art, concerns, regulatory control and certification. *Online journal of public health informatics*, *5*(3).

[3] Siswati, T., Fathmawati, F. A. T. H. M. A. W. A. T. I., Paramashanti, B. A., Waris, L. U. K. M. A. N., & Rialihanto, M. P. (2023). Development And Usability Testing Of A Mobile Health Application On Child Growth And Development Monitoring. DEVELOPMENT, 60(11).

[4]Esiefarienrhe, B. M., & Makokoe, M. C. P. (2023). The Design and Development of Parent-Children Vaccination Alert System. Journal of Information Systems and Informatics, 5(3), 1177-1193.

[5]Uchechukwu, B. N., & Ohinameuwa, A. (2025). Enhanced Health (Record) Information Management System Using Mobile Application Development Framework. Journal of Science and Technology, 30(3).

[6]Abdullah, M. A., & Alghali, M. (2021, February). Patient Drugs and Data Management by Mobile Application. In 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE) (pp. 1-4). IEEE.

Websites:

[7] https://www.who.int/

Application:

[8] MediConnect Services Pvt Ltd, “Oladoc – The Health App,” Version 6.1.5, Mobile application, May 2025. [Online]. Available: https://play.google.com/store/search?q=oladoc&c=apps&hl=en

[9] Transformation Studios, “Marham Connect,” Version 11.8, Mobile application, February 2025. [Online]. Available: https://play.google.com/store/search?q=marham&c=apps&hl=en

[10] Sehat Kahani, “Sehat Kahani App,” Version 11.1, Mobile application, April 2025. [Online]. Available: https://play.google.com/store/apps/details?id=com.sehatkahani.corp.app&hl=en

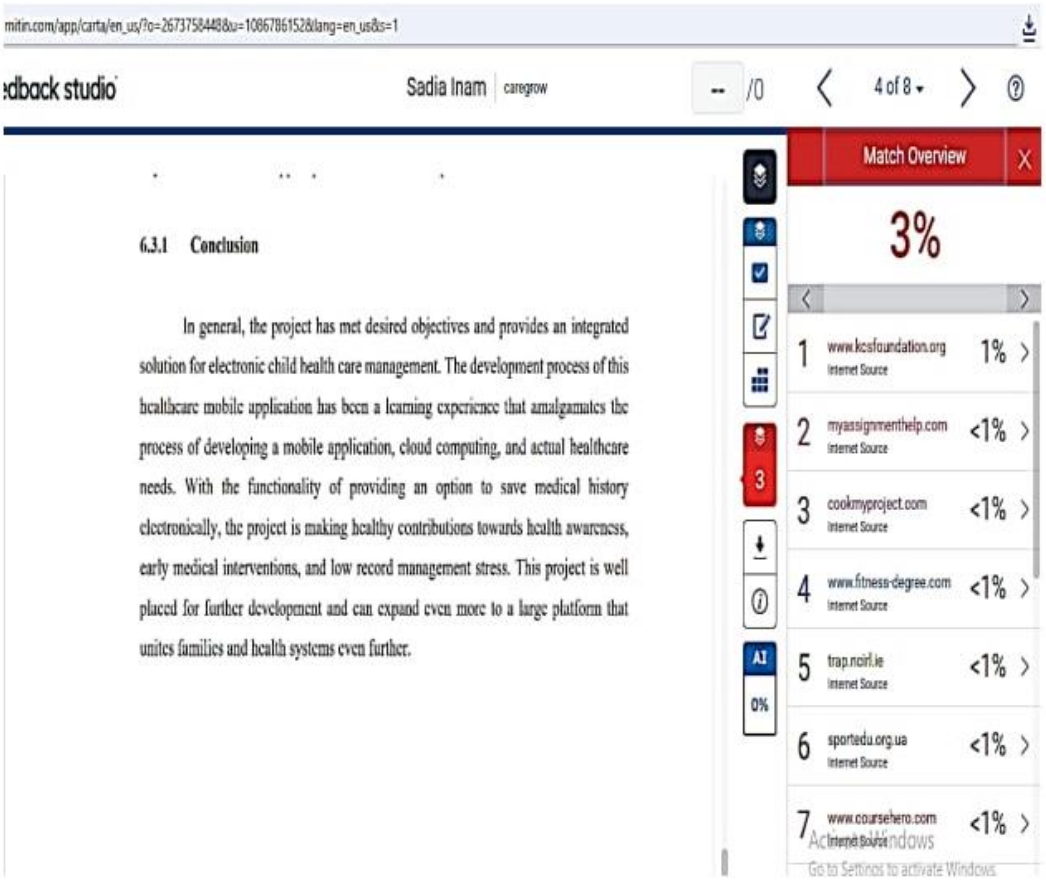
[11] BabyCenter, “BabyCenter Pregnancy Tracker,” Version 6.4, Mobile application, May 2025. [Online]. Available: https://play.google.com/store/search?q=babycenter&c=apps&hl=en

APPENDICES

caregrow

by Sadia Inam

Submission date: 12-May-2025 03:40PM (UTC+0500)
Submission ID: 2673758448
File name: CareGrow_FYP_report.docx (2.5M)
Word count: 9020
Character count: 50443



6.3.1 Conclusion

In general, the project has met desired objectives and provides an integrated solution for electronic child health care management. The development process of this healthcare mobile application has been a learning experience that amalgamates the process of developing a mobile application, cloud computing, and actual healthcare needs. With the functionality of providing an option to save medical history electronically, the project is making healthy contributions towards health awareness, early medical interventions, and low record management stress. This project is well placed for further development and can expand even more to a large platform that unites families and health systems even further.

caregrow

ORIGINALITY REPORT

3%	3%	1%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

2%

★ www.coursehero.com

Internet Source

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		