Cairo University

Faculty of Computers and Artificial Intelligence

Information Systems Department

**Graduation Project Report**

**MOMMY MENTOR**

**Presented by**

|  |  |
| --- | --- |
| **ID** | **Name** |
| 20200186 | *Reem Ayman Abd El-Fattah* |
| 20201064 | *Rahma Elsayed Abo-elgheet* |
| 20201056 | *Hasnaa Gamil Abd-elnaby* |
| 20200228 | *Salma Sherif Mostafa* |
| 20201218 | *Walaa Soudy Ibrahim* |
| 20200733 | *Sara Mohamed Moustafa* |

**Under the Supervisors of:**

*Dr. Dina Ezzat*

*TA. Ahmed Mohamed Galal*

*Department of Information Systems, Faculty of Computers and Artificial Intelligence, Cairo University, Egypt*

July 2024

|  |  |
| --- | --- |
| Background and Motivation | All Team |
| Problem Statement | All Team |
| Objectives | All Team |
| Project Scope and Limitations | All Team |
| Project Methodology | All Team |
| Project Report Outline | All Team |
| Related works | All Team |
| Competitive Analysis | All Team |
| Functional & Non-Functional requirements | All Team |
| Project Stakeholders | All Team |
| Use Case Diagram & System ERD | All Team |
| Domain Class Diagram & System Seq. Diagrams | All Team |
| System Scenarios & Design Class Diagram | All Team |
| Design Sequence Diagrams & System Architecture | All Team |
| Main Techniques, Technologies, and Applications | All Team |
| Testing | All Team |
| Conclusions and Future work | All Team |
| Frontend Implementation | All Team |
| Backend Implementation | All Team |
| Machine learning models | All Team |

**Acknowledgements**

**Abstract**

Caring for a newborn as a first-time mother is challenging. A first-time mother needs daily advice and guidance appropriate to her child’s age regarding food and diseases. Lack of experience and unreliable advice can harm the child. First-time mothers often receive a lot of information from different sources, which can cause confusion and lead to misunderstandings.

Mommy Mentor provides reliable information for mothers to gain experiences. It helps mothers make informed decisions about their children's health and monitor their natural development by viewing understandable charts. The platform educates mothers about vaccinations suitable for their child's age, assisting them in keeping track of their child's immunization schedule and promoting community health, furthermore, finding suitable doctor for her child. Additionally, Mommy Mentor includes three advanced machine-learning models that enable early detection and intervention: one for predicting Down syndrome and another for Jaundice from baby images, and a symptom-based diagnosis prediction model that interprets common symptoms to suggest potential illnesses. These tools, combined with what we mentioned before, make Mommy Mentor a comprehensive support system for new mothers, ensuring their confidence and peace of mind every step of the way.

**Table of Contents**

[**Chapter 1**](#_heading=h.1fob9te) **Introduction** 6

[1.1 Background and Motivation 6](#_heading=h.3znysh7)

[1.1.1 The Main Area of the Project 6](#_heading=h.3znysh7)

[1.1.2 Motivation and Beneficiary 6](#_heading=h.3znysh7)

[1.2 Problem Statement 6](#_heading=h.2et92p0)

[1.3 Objectives 7](#_heading=h.tyjcwt)

[1.4 Project Scope and Limitations 7](#_heading=h.1t3h5sf)

[1.5 Project Methodology 9](#_heading=h.2s8eyo1)

[1.6 Project Report Outline 9](#_heading=h.17dp8vu)

[**Chapter 2**](#_heading=h.3rdcrjn) **Market and literature** 11

[2.1 Related Work 11](#_heading=h.26in1rg)

[2.2 Competitive Analysis 12](#_heading=h.35nkun2)

[2.2.1 Main Differences between our system and theirs 13](#_heading=h.35nkun2)

[**Chapter 3**](#_heading=h.z337ya) **Analysis** 14

[3.1 Functional Requirements 14](#_heading=h.3j2qqm3)

[3.2 Non Functional Requirements 15](#_heading=h.3j2qqm3)

[3.3 Project Stakeholders 15](#_heading=h.3j2qqm3)

[3.4 Use Case Diagram 16](#_heading=h.3j2qqm3)

[3.5 System ERD 17](#_heading=h.4i7ojhp)

[3.6 Domain Class Diagram 18](#_heading=h.4i7ojhp)

[3.7 System Sequence Diagrams 19](#_heading=h.4i7ojhp)

[3.8 System Scenarios 23](#_heading=h.4i7ojhp)

[**Chapter 4**](#_Chapter_4_Design) **design and implementation** 30

[4.1 Design Class Diagram 30](#_heading=h.1ci93xb)

[4.2 Design Sequence Diagrams 31](#_heading=h.3whwml4)

[[4.3 Main Techniques, Technologies, and Application](#_4.3_Main_Techniques,) 35](#_heading=h.3whwml4)

[4.4 System Architecture 36](#_heading=h.3whwml4)

[4.5 Prototype 38](#_heading=h.3whwml4)

[[**Chapter 5**](#_Chapter_4_Design) **testing and Evaluation** 42](#_heading=h.3whwml4)

[5.1 Testing and Evaluation 42](#_heading=h.3whwml4)

[5.2 Development 70](#_heading=h.3whwml4)

[[**Chapter 6**](#_Chapter_4_Design) **Conclusions and Future Work** 76](#_heading=h.3whwml4)

[6.1 Conclusions 76](#_heading=h.3whwml4)

[6.2 Recommendation and Future Work 77](#_heading=h.3whwml4)

**List of Figures**

*Figure number Page Number*

[Figure 1. Project Plan. 8](#_heading=h.4d34og8)

[Figure 2. Use Case Diagram 16](#_heading=h.4d34og8)

[Figure 3. System ERD 17](#_heading=h.4d34og8)

[Figure 4. Domain Class Diagram 18](#_heading=h.4d34og8)

[Figure 5. Follow up Growth System Seq. Diagram 19](#_heading=h.4d34og8)

[Figure 6. Show Age-Related Advice System Seq. Diagram 19](#_heading=h.4d34og8)

[Figure 7. Enter Information System Seq. Diagram 20](#_heading=h.4d34og8)

[Figure 8. Filter doctor system Seq. Diagram 20](#_heading=h.4d34og8)

[Figure 9. Down syndrome model system Seq. Diagram 21](#_heading=h.4d34og8)

[Figure 10. Symptom Disease model system Seq. Diagram 21](#_heading=h.4d34og8)

[Figure 11. Jaundice Detection model system Seq. Diagram 21](#_heading=h.4d34og8)

[Figure 12. Design Class Diagram 28](#_heading=h.4d34og8)

[Figure 13. Enter Information Design Seq. Diagram 31](#_heading=h.4d34og8)

[Figure 14. Follow up Growth Design Seq. Diagram 31](#_heading=h.4d34og8)

[Figure 15. Filter Doctor Design Seq. Diagram 32](#_heading=h.4d34og8)

[Figure 16. Down syndrome model Design Seq. Diagram 33](#_heading=h.4d34og8)

[Figure 17. Show Age-Related Advice Design Seq. Diagram 33](#_heading=h.4d34og8)

[Figure 18. Symptom Disease model Design Seq. Diagram 33](#_heading=h.4d34og8)

[Figure 19. Jaundice Detection model Design Seq. Diagram 34](#_heading=h.4d34og8)

[Figure 20. Data Flow 36](#_heading=h.4d34og8)

[Figure 21. System Overview 37](#_heading=h.4d34og8)

**List of Tables**

*Table number Page Number*

[Table 1. Competitive Analysis 12](#_heading=h.2jxsxqh)

[Table 2. System Scenario 1 23](#_heading=h.2jxsxqh)

[Table 3. System Scenario 2 24](#_heading=h.2jxsxqh)

[Table 4. System Scenario 3 25](#_heading=h.2jxsxqh)

[Table 5. System Scenario 4 26](#_heading=h.2jxsxqh)

[Table 6. System Scenario 5 27](#_heading=h.2jxsxqh)

[Table 7. System Scenario 6 28](#_heading=h.2jxsxqh)

[Table 8. System Scenario 7 29](#_heading=h.2jxsxqh)

[Table 9. Testing result 1 42](#_heading=h.2jxsxqh)

[Table 10. Testing result 2 43](#_heading=h.2jxsxqh)

[Table 11. Testing result 3 44](#_heading=h.2jxsxqh)

[Table 12. Testing result 4 45](#_heading=h.2jxsxqh)

[Table 13. Testing result 5 46](#_heading=h.2jxsxqh)

[Table 14. Testing result 6 47](#_heading=h.2jxsxqh)

[Table 15. Testing result 7 48](#_heading=h.2jxsxqh)

[Table 16. Testing result 8 49](#_heading=h.2jxsxqh)

[Table 17. Testing result 9 50](#_heading=h.2jxsxqh)

[Table 18. Testing result 10 51](#_heading=h.2jxsxqh)

[Table 19. Testing result 11 52](#_heading=h.2jxsxqh)

[Table 20. Testing result 12 53](#_heading=h.2jxsxqh)

[Table 21. Testing result 13 54](#_heading=h.2jxsxqh)

[Table 22. Testing result 14 55](#_heading=h.2jxsxqh)

[Table 23. Testing result 15 56](#_heading=h.2jxsxqh)

[Table 24. Testing result 16 57](#_heading=h.2jxsxqh)

[Table 25. Testing result 17 58](#_heading=h.2jxsxqh)

[Table 26. Testing result 18 60](#_heading=h.2jxsxqh)

[Table 27. Testing result 19 61](#_heading=h.2jxsxqh)

[Table 28. Testing result 20 62](#_heading=h.2jxsxqh)

[Table 29. Testing result 21 63](#_heading=h.2jxsxqh)

[Table 30. Testing result 22 65](#_heading=h.2jxsxqh)

[Table 31. Testing result 23 67](#_heading=h.2jxsxqh)

[Table 32. Testing result 24 68](#_heading=h.2jxsxqh)

[Table 33. Testing result 25 69](#_heading=h.2jxsxqh)

**List of Abbreviations**

|  |  |
| --- | --- |
| Abbreviation | Meaning |
| UML | Unified Modeling Language |
| ERD | Entity Relationship Diagram |
| API | Application Programming Interface |
| IOS | IPhone Operating System |
| SQL | Structured Query Language |
| NoSQL | Not only SQL |
| JSON | JavaScript Object Notation |
| XML | Extensible Markup Language |
| UI | User Interface |

# Chapter 1 Introduction

1.1 Background and Motivation

### 1.1.1 The Main Area of the Project

Our main goal is to help the first-time mothers - with no experience in dealing with newborn babies - to find it less difficult to cope with a newborn baby, take care of his health and growth and be more aware of his needs from birth until three years old in addition to getting rid of their fear.

### 1.1.2 Motivation and Beneficiary

First-time mothers encounter numerous challenges in the early stages. These include understanding their child's needs, accessing age-appropriate advice, seeking support, and ensuring their child receives timely vaccinations. Failure to adhere to vaccination schedules puts children at a higher risk of contracting diseases. For instance, there was an 18% increase in measles cases in 2022, and global deaths rose by 43% compared to the previous year due to lower vaccination rates.

Our objective is to develop an application that simplifies these challenges by providing the mother with all the resources and information related to all aspects of children and advice for her child she needs, and to gain and exchange experiences with other mothers, including vaccination list suitable for baby’s age and finding the nearest vaccination center. Additionally, the integration of advanced machine-learning models makes our app stand out by offering functionalities that other apps do not. The models include a Down syndrome prediction model, a Jaundice prediction model, and a symptom-based diagnosis prediction model. These tools enable early detection of potential health issues, offering mothers immediate, data-driven insights and advice tailored to their child's specific needs.

1.2 Problem Statement

Being a first-time mother can be challenging and overwhelming. It is common for mothers to have limited knowledge about what is best for their children. During this crucial stage, it is important for mothers to educate themselves and gather information from reliable sources. Using a single application that provides accurate and useful information can greatly simplify the process instead of relying on multiple sources that may provide incorrect information. The reason behind the development of this application is the absence of information regarding essential vaccinations for children poses a potential risk to their well-being if mothers are unaware of the appropriate vaccination schedules.

1.3 Objectives

The objectives of the Mommy Mentor app can be outlined as follows:

* Help mothers to track how fast a child grows, see if they are developing normally, or not.
* Provide the mothers with a comprehensive library of articles, tips, and expert advice on various topics related to baby care, nutrition, child growth, diseases, abnormal babies and vaccinations.
* Help mothers remember when their child needs vaccines and show them where to go for shots so they do not miss any important immunizations.
* Provide a chat platform for direct communication between mothers and doctors and between the mothers, making it easy for them to ask questions, seek advice, and receive guidance.
* Ensure moms can discuss sensitive topics securely and easily seek advice while also making sure the website or app is simple to use for everyone, no matter their tech skills.
* Integrate three machine-learning models to assist mothers in early diagnosis and facilitate early awareness and consultation: a Down syndrome prediction model, a Jaundice prediction model, and a symptom-based diagnosis prediction model.

1.4 Project Scope and Limitations

**Project Scope:**

The project scope encompasses the development and implementation of a comprehensive mobile application designed to support and empower first-time

mothers in navigating the challenges of newborn care. The key features and functionalities include:

* Informative content to empower mothers with knowledge.
* Direct communication between mothers and doctors and between the mothers.
* Mothers can display vaccine suitable for baby’s age.
* The nearest vaccination center will be suggested to mothers.
* Provide information about pediatrician clinics, including locations and fees.
* Receiving advice tailored to each child’s age.
* Track the baby’s growth and visualize it using charts.
* Support the Arabic Language.
* **Advanced Diagnostic Tools (Machine Learning models) to predict some diseases in the child.**

**Limitations:**

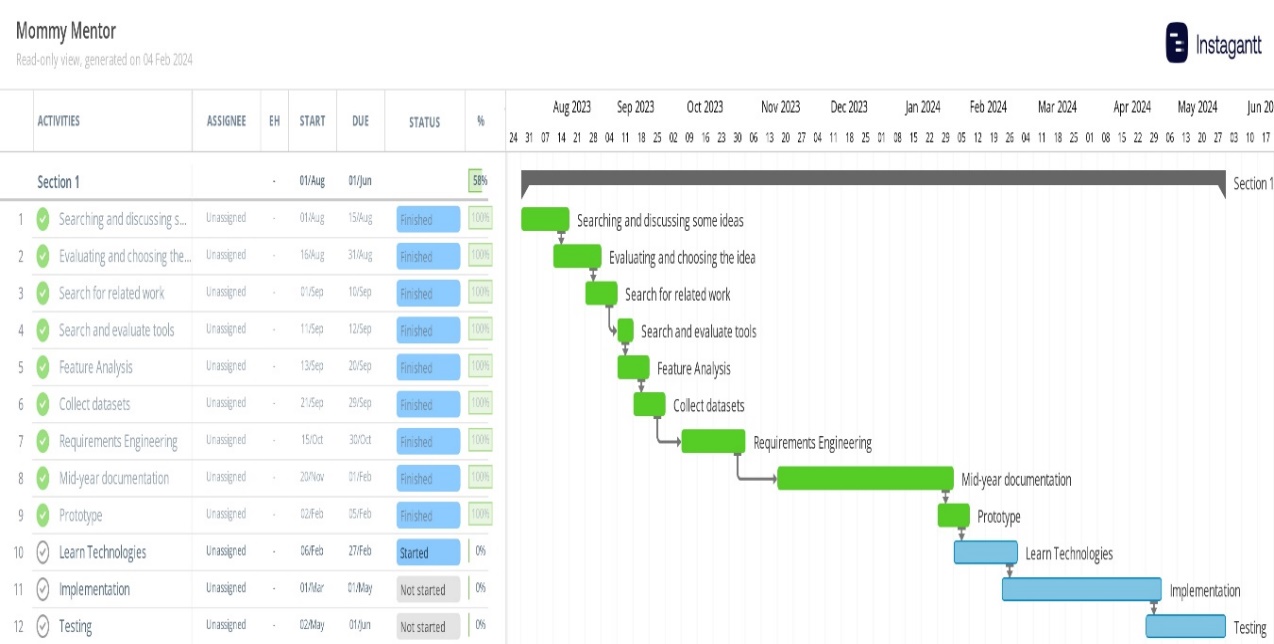
****It is the users' responsibility to ensure that the information they enter into the app is accurate and appropriate. Mommy Mentor cannot be held liable for any decisions made based on user-entered data.

Figure 1. Project Plan

1.5 Project Methodology

We will use the Waterfall Methodology for the following reasons:

1. **Clear Phases and Milestones:**

Waterfall is characterized by a sequential, linear approach with distinct phases. This clear structure aligns well with the need to establish foundational elements for the Mommy Mentor app.

1. **Well-Defined Requirements:**

Waterfall is most effective when project requirements are well understood and unlikely to change significantly.

1. **Minimal Changes During Development:**

In the Waterfall model, changes during development are expected to be minimal. For Mommy Mentor there is a need for a stable and well-established information hub and tracking system before adding other features.

1. **Detailed Documentation:**

Waterfall encourages comprehensive documentation at each stage of development.

1. **Risk Management:**

Waterfall involves thorough upfront planning, which includes identifying and addressing potential risks early in the project.

1.6 Project Report Outline

- Chapter 2 presents the competitive analysis applied to compare between the functionalities and services provided by our app and the existing similar apps in the market.

- Chapter 3 presents an analysis of the project’s idea, the system’s functionalities are defined, and the required quality indicators are set, in addition to some UML diagrams describing the possible scenarios within the system and the sequence of the main functionalities.

- Chapter 4 presents the design of the user interface, in addition to some UML diagrams to clarify the way each use case acts, along with the frameworks and the database that will be used in implementing the actual code of the system.

- Chapter 5 presents some of the detailed system test cases, used testing tools, and the test results of each system requirement.

- Chapter 6 presents conclusions to validate the system needs and how the presented system solved the problem stated. Finally, describe future suggestions to improve the system.

# Chapter 2 Market and Literature Survey

2.1 Related Work

**iMumz:** It is a pregnancy and parenting application that uses the baby's milestone tracker, provides postpartum dietary tips and personalized diet plans, allows consulting with a lactation expert, includes a postpartum recovery and parenting journey, employs a baby feeding tracker, and provides expert articles.

**Baby+:** It is a parenting application with daily blogs that provides weekly development guides that show the growth of the baby, parenting guides through the first year, postpartum recovery, and activity ideas for age-appropriate activities to keep the baby entertained.

**Mamahood:** It’s a pregnancy and parenting application that provides expert advice and parenting articles, allows searching for other mothers and chatting with them, allows at-home lab testing, and includes professional care and guidance along with community engagement through fertility, pregnancy, and motherhood.

**Bebbo parenting:** It’s a parenting application to support a child’s growth from birth to age 6, which provides learning through play activities and games, expert advice and parenting articles, parent advice and tips on taking care of a child’s and parent’s well-being, easy navigation between multiple child profiles, and receiving advice tailored to each child’s developmental needs and age.

2.2 Competitive Analysis

Table 1- Competitive Analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Apps***  ***Features*** | **iMumz** | **Baby +** | **MamaHood** | **Bebbo parenting** | **Mommy Mentor** |
| **Baby-age-related content** | **✓** | **✓** | **✓** | **✓** | **✓** |
| **Medical articles covering all aspects of the child** | **✓** | **✓** | **✓** | **✓** | **✓** |
| **Educational videos** | **✓** | **✓** | **🗶** | **🗶** | **✓** |
| **Growth tracking and represent it with charts** | **🗶** | **✓** | **🗶** | **✓** | **✓** |
| **Live chat with doctor** | **🗶** | **🗶** | **✓** | **🗶** | **✓** |
| **Toddler Games** | **🗶** | **✓** | **🗶** | **✓** | **🗶** |
| **Vaccine reminder** | **🗶** | **🗶** | **🗶** | **🗶** | **✓** |
| **Support Arabic and English languages** | **🗶** | **🗶** | **🗶** | **🗶** | **✓** |
| **Checking doctors' appointments** | **🗶** | **🗶** | **🗶** | **🗶** | **✓** |
| **Machine-Learning Models** | **🗶** | **🗶** | **🗶** | **🗶** | **✓** |

### 2.2.1 Main Differences between our system and theirs:

Unlike other applications, it supports the Arabic language and is considered the first of its kind in displaying vaccine suitable for baby’s age. It also recommends the mother to the nearest vaccination center according to her residential address.

The application stands out by providing the mother with the ability to access information about pediatrician clinics, including their locations and fees. Moreover, it enables direct communication with the doctor. Additionally, the app includes a recommendation system that displays posts tailored to the mother's interests.

# Chapter 3 Analysis

3.1 Functional Requirements

* Mothers can browse the articles, which is divided into several categories: Children's diseases, Child growth rate…...etc.
* Users can check available doctors’ appointments in different ways: by certain doctor's name, by location or by fees.
* Users can track baby's growth and know if it is in normal range or if there is a problem and browsing vaccinations suitable for this age.
* Users can chat with each other to share experience.
* Doctors reply to mothers’ questions and chat with others.
* Doctors can publish medical articles and add available appointemant.
* Administrators can delete shared articles and users.
* Administrators can add, edit and update advice and vaccine.
* Machine-Learning Diagnostic Models:
  + Down Syndrome Model: Allow users to upload an image or take a photo of their baby and receive a prediction on whether the baby has Down syndrome.
  + Jaundice Model: Enable users to upload an image or take a photo of their baby and get a prediction on whether the baby has Jaundice.
  + Symptom-Based Diagnosis Prediction: Collect true or false responses from mothers on symptoms like cough, runny nose, rash, shortness of breath, suddenness, and nervousness to predict potential illnesses.

3.2 Non Functional Requirements

* Usability:

The system should be easy to use, which means users could accomplish basic tasks the first time they encounter the design as well as have a friendly user interface.

* Scalability:

The system architecture should be designed to handle a scalable user base and increasing data volume. It should accommodate 100 concurrent users without degradation in response time or system performance.

* Response time:

The system should respond to user interactions within a maximum of 5 seconds under normal load conditions.

* Throughput:

Additionally, the system should be able to handle a simultaneous user load of at least 100 users without experiencing a degradation in response time beyond the defined threshold.

* Maintainability:

The system should be easy to test and repair if any fault has happened, extensible for new functionality, and easy to modify.

3.3 Project Stakeholders

**Mothers:** The main users of the system who could keep up with their baby growth and vaccines and read medical articles regarding babies in general.

**Doctors:** They can reply to mother’s questions, review, and upload the medical articles’ information.

**Admins:** Who controls the system, edit, update and delete some app’s information .

3.4 Use Case Diagram

 Figure 2. Use Case Diagram

3.5 System ERD

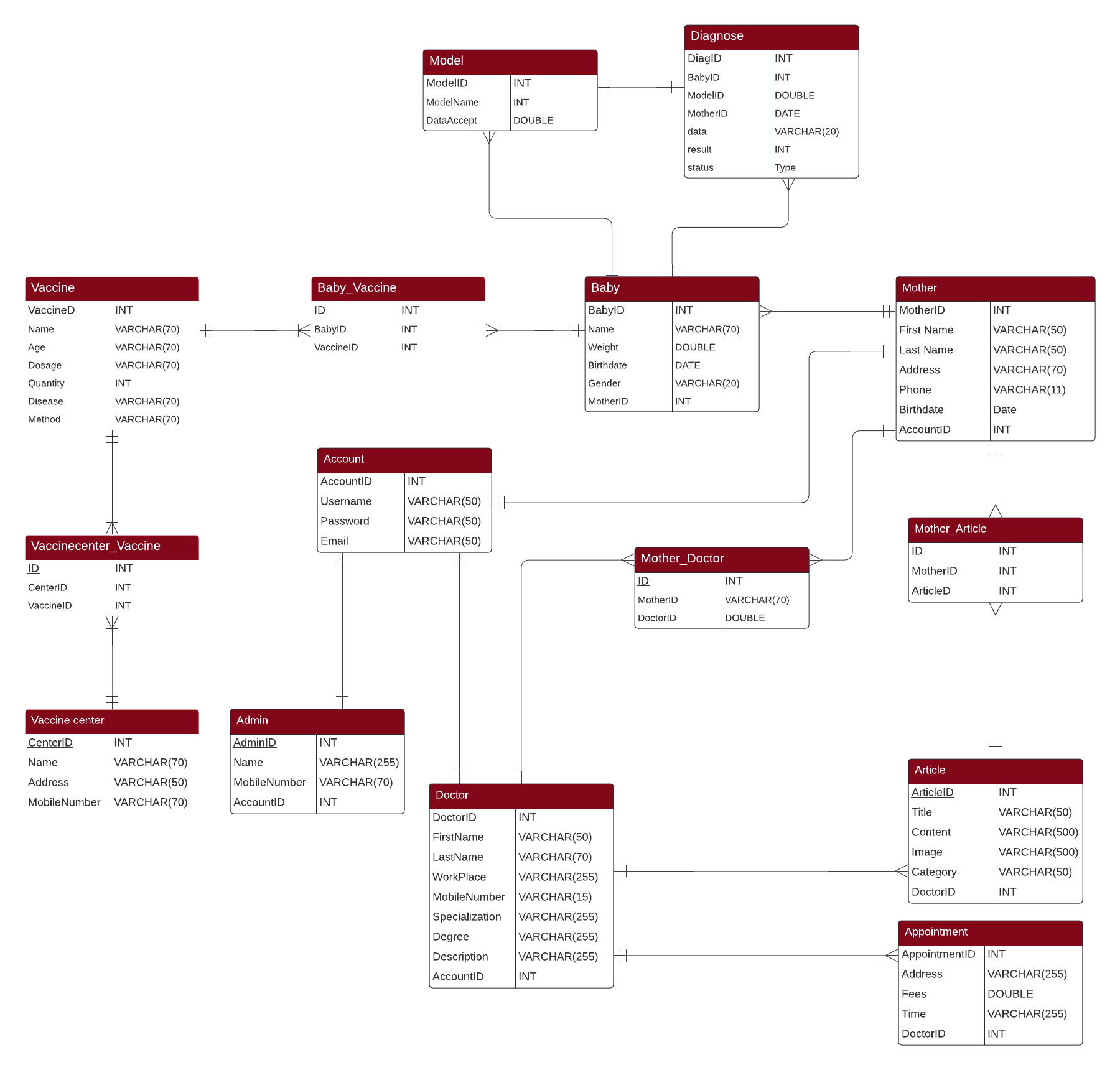


Figure 3. System ERD

3.6 Domain Class Diagram

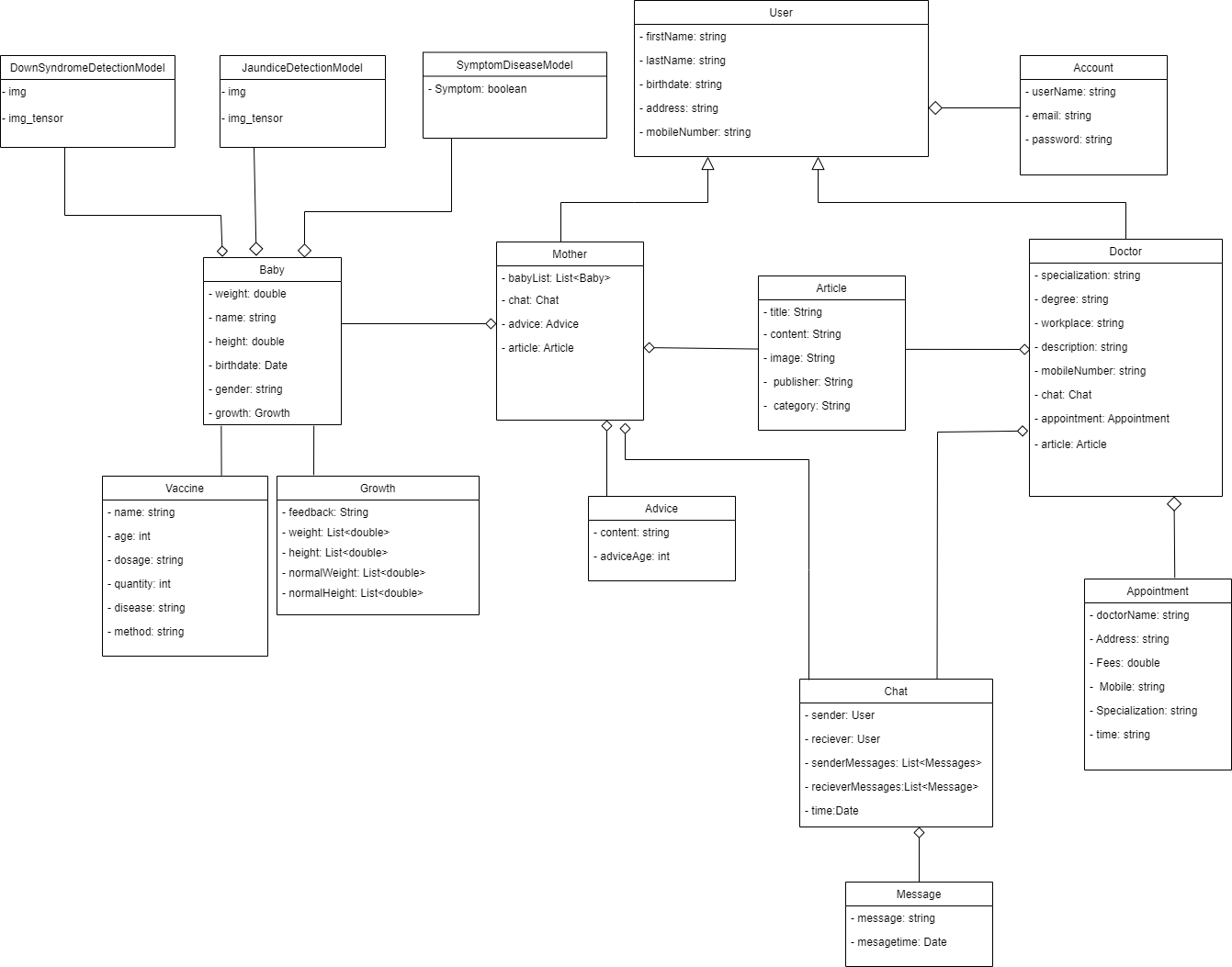


Figure 4. Domain Class Diagram

3.7 System Sequence Diagrams

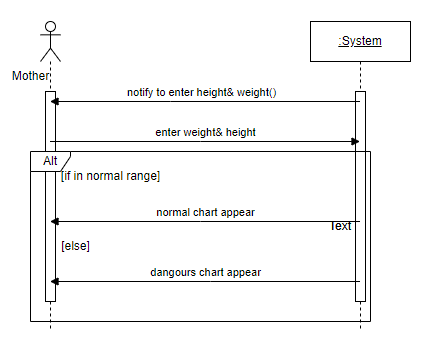


Figure 5. Follow up Growth System Seq. Diagram

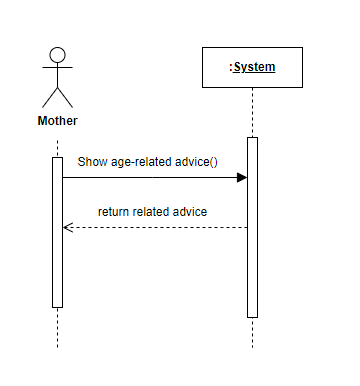


Figure 6. Show Age-Related Advice System Seq. Diagram

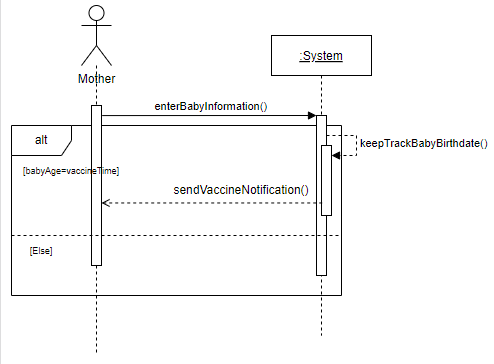


Figure 7. Enter Information System Seq. Diagram

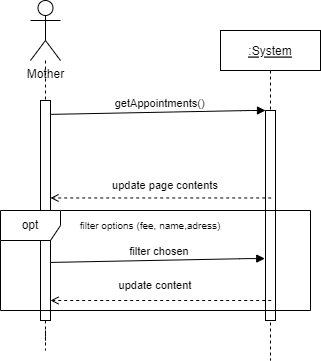


Figure 8. Filter doctor system Seq. Diagram

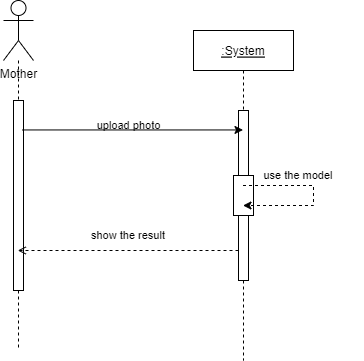


Figure 9. Down syndrome model system Seq. Diagram

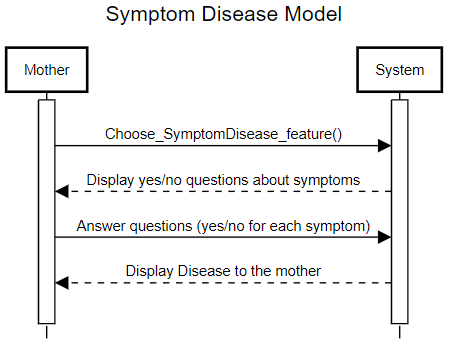


Figure 10. Symptom Disease model system Seq. Diagram

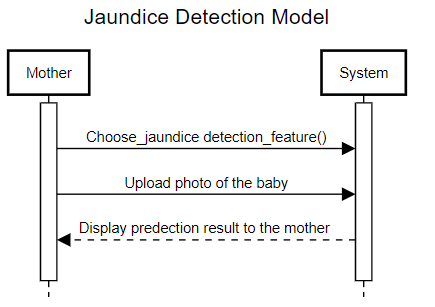


Figure 11. Jaundice Detection model system Seq. Diagram

3.8 System Scenarios

|  |  |  |
| --- | --- | --- |
| **Use case name** | Follow Up Growth | |
| **Scenario** | Follow up baby’s growth | |
| **Triggering Events** | When a mother enters new height and weight. | |
| **Brief Description** | A mother wants to know if her baby has appropriate weight and height according to his/her age, if no, dangerous warning will appear. | |
| **Actors** | Mother | |
| **Related Use cases** | Notify to enter weight and height. | |
| **Preconditions** | The mother must enter new height and weight. | |
| **Post conditions** | She knows if her baby’s growth rate is normal or not. | |
| **Flow of activities** | **User** | **System** |
|  | 1. Notify mother to enter weight and height. |
| 1. A mother enters a logic weight and height. |  |
|  | 1. Show the state of the growth normal, not normal but acceptable or dangers. |
| **Exceptional conditions** | 1. A mother enters weight and height out of range. |  |
|  | 1. Show error massage. |

Table 2. System Scenario 1

|  |  |  |
| --- | --- | --- |
| **Use case name** | Show age-related advice | |
| **Scenario** | Show advice based on the baby’s age. | |
| **Triggering Events** | When a mother wants to take advice regarding her baby depending on his age. | |
| **Brief Description** | After a mother has signed up and filled her baby’s age, she can ask the program to give her useful tips for her child. | |
| **Actors** | Mother | |
| **Related Use cases** | - | |
| **Preconditions** | The age must be stored when the mother has signed up. | |
| **Post conditions** | The program displays advice successfully. | |
| **Flow of activities** | **User** | **System** |
| 1. A mother signs up and fill out the information of her child. |  |
| 1. Want to take an advice. |  |
|  | 1. Retrieve the age from the database and search for an advice based on it. |
|  | 1. Display the advice to the mother. |
| **Exceptional conditions** | **The age is not in our scope.** | |
|  | 1- The system will display an error message that the age is not in the scope (in the specified range) and will not display any advice.  2- Ask the user to re-enter the age if it is within the scope. |

Table 3. System Scenario 2

Table 4. System Scenario 3

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case name** | Send notification for vaccine | | |
| **Scenario** | The system sends a notification to mothers for vaccination. | | |
| **Triggering Events** | When baby’s age matches with specific vaccine time. | | |
| **Brief Description** | The system keeps track of a baby’s birth date. When it finds that the baby’s age matches a specific vaccine time, it sends a notification to the mother to remind her. | | |
| **Actors** | System | | |
| **Related Use cases** | - | | |
| **Preconditions** | The mother must have account with her child’s info. | | |
| **Post conditions** | A notification of a vaccine is sent to a mother. | | |
| **Flow of activities** | **Mother** | **System** | |
| 1. Signs up with her baby info. |  | |
|  | 1. Keeps track of baby’s birth date | |
|  | 1. Sends a notification to the mother when her baby’s age matches a specific vaccine time. | |
| **Exceptional conditions** | **The notification can’t be sent** | | |
| 1. The mother doesn’t sign up with her baby birthdate | |  |
|  | | 1. The system cannot send the notification to a mother. |

|  |  |  |
| --- | --- | --- |
| **Use case name** | Filter Doctor | |
| **Scenario** | Filter Doctor based on mother choice | |
| **Triggering Events** | When a mother views the doctors’ list and wants to book an appointment. | |
| **Brief Description** | The mother enters the page of the doctors’ list and wants to book an appointment based on her filtration option. | |
| **Actors** | Mother | |
| **Related Use cases** | Filter appointment. | |
| **Preconditions** | - | |
| **Post conditions** | She knows the appropriate appointments for her of the desired doctors. | |
| **Flow of activities** | **User** | **System** |
| 1-The mother wants to view the available appointments. |  |
|  | 2- Show the available appointments. |
| 3-She choose the filtration option (Fee, Name or Location) and enter any other input related to her choice. |  |
|  | 4- Show the appointments based on the mother’s choice. |
| **Exceptional conditions** | 1- A mother enters her desired option. |  |
|  | 2- No available appointments. |

Table 5. System Scenario 4

|  |  |  |
| --- | --- | --- |
| **Use case name** | Down Syndrome model | |
| **Scenario** | Predict if the baby has down syndrome. | |
| **Triggering Events** | When a mother enters her baby’s image and wants to check it with the models provided with our app. | |
| **Brief Description** | When a mother enters her baby’s image and wants to check it with the models provided with our app to know he has the down syndrome or not. | |
| **Actors** | Mother | |
| **Related Use cases** | - | |
| **Preconditions** | The mother enters the image of her baby. | |
| **Post conditions** | She knows if her baby has down syndrome or not. | |
| **Flow of activities** | **User** | **System** |
| 1-The mother enters her baby’s image. |  |
|  | 2- Send the image to the predictor model.  3- Show response if the baby has down syndrome or not. |
| **Exceptional conditions** | 1- A mother enters the image. |  |
|  | 2- Show wrong response. |

Table 6. System Scenario 5

|  |  |  |
| --- | --- | --- |
| **Use case name** | Jaundice Detection Model | |
| **Scenario** | Predict if the baby has Jaundice or not. | |
| **Triggering Events** | When a mother uploads a baby’s photo to check for jaundice using the models provided in the app. | |
| **Brief Description** | When a mother uploads a baby’s photo, the app analyzes the image to detect if the baby has jaundice or not. | |
| **Actors** | Mother | |
| **Related Use cases** | - | |
| **Preconditions** | The mother uploaded a clear photo of the baby. | |
| **Post conditions** | The application provides a result indicating whether the baby has jaundice or not. | |
| **Flow of activities** | **User** | **System** |
| 1-The mother enters her baby’s photo. |  |
|  | 2- Send the image to the predictor model.  3- Show response if the baby has jaundice or not. |
| **Exceptional conditions** | 1- A mother enters the image. |  |
|  | 2- Show wrong response. |

Table 7. System Scenario 6

|  |  |  |
| --- | --- | --- |
| **Use case name** | Symptom Disease Model | |
| **Scenario** | Predict the disease of the baby based on the symptoms entered by the mother. | |
| **Triggering Events** | When a mother wants to check her baby’s symptoms using the models provided in the app to know the expected disease. | |
| **Brief Description** | When a mother answers yes or no questions about her baby’s symptoms, the app analyzes the answers to predict potential diseases or conditions. | |
| **Actors** | Mother | |
| **Related Use cases** | - | |
| **Preconditions** | The mother has observed symptoms in the baby and answer yes or no questions. | |
| **Post conditions** | The application provides disease based on the symptoms entered. | |
| **Flow of activities** | **User** | **System** |
|  | 1- Displays a series of yes/no questions about the baby's symptoms. |
| 2- Answers the questions by selecting yes or no for each symptom. | 3- Collects the responses and processes them  4- Analyzes the symptoms and generates a recommendation for potential diseases or conditions. |
| **Exceptional conditions** | 1- The mother answers the questions by selecting yes or no for each symptom. |  |
|  | 2- System encounters an error and displays an error message to the mother. |

Table 8. System Scenario 7

# Chapter 4 Design and Implementation

4.1 Design Class Diagram

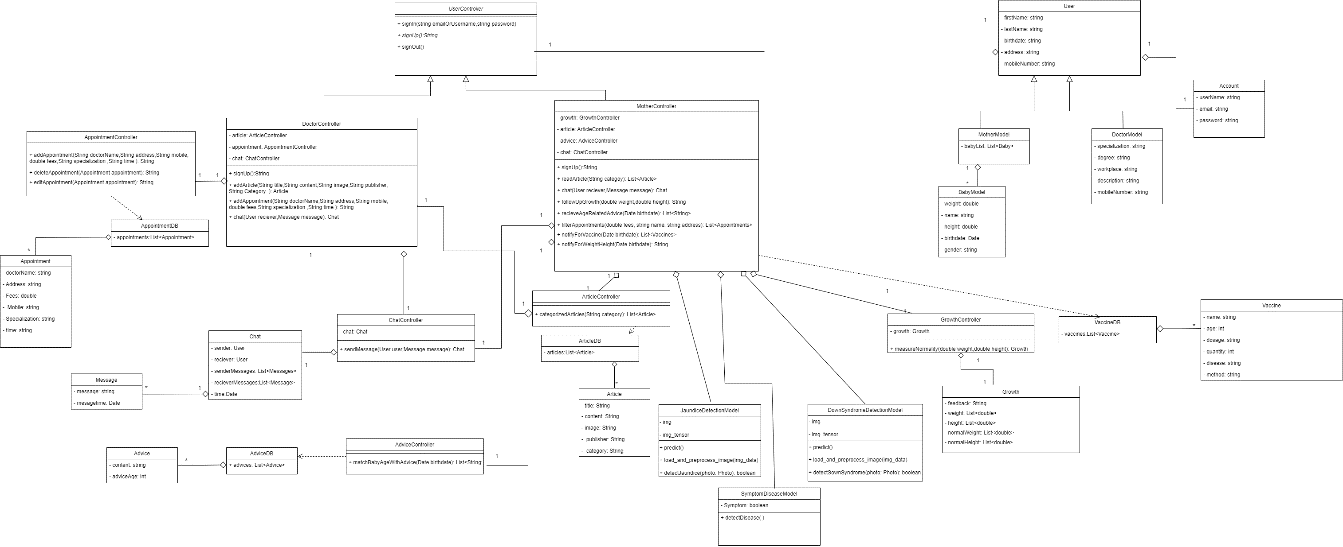


Figure 12. Design Class Diagram

4.2 Design Sequence Diagrams

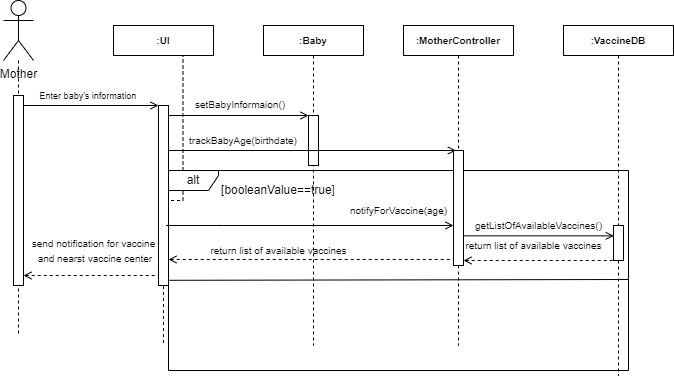
****

Figure 13. Enter Information Design Seq. Diagram

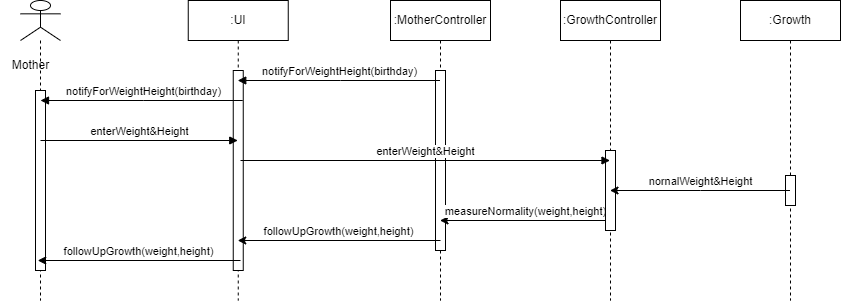
****

Figure 14. Follow up Growth Design Seq. Diagram

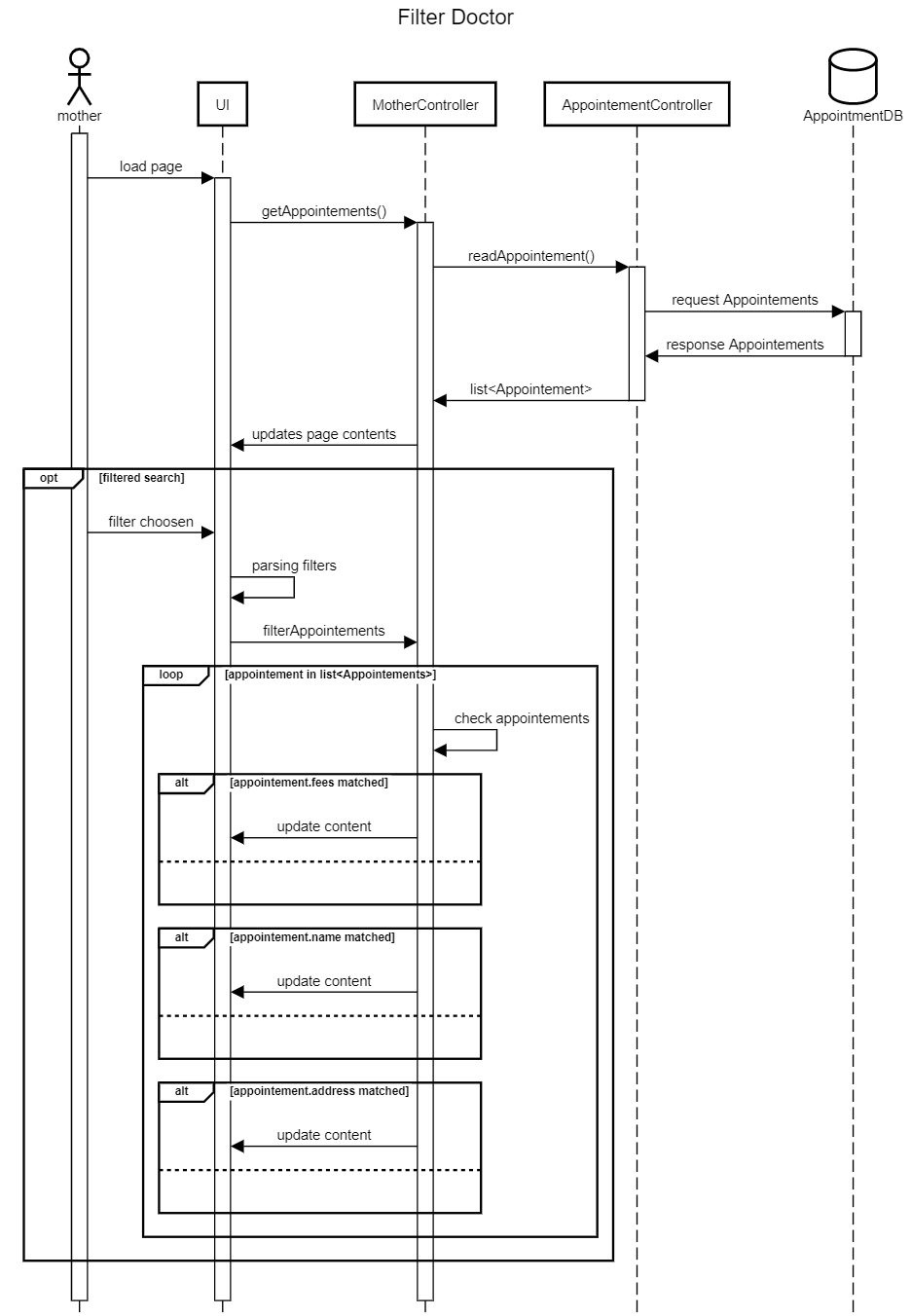
****

Figure 15. Filter Doctor Design Seq. Diagram

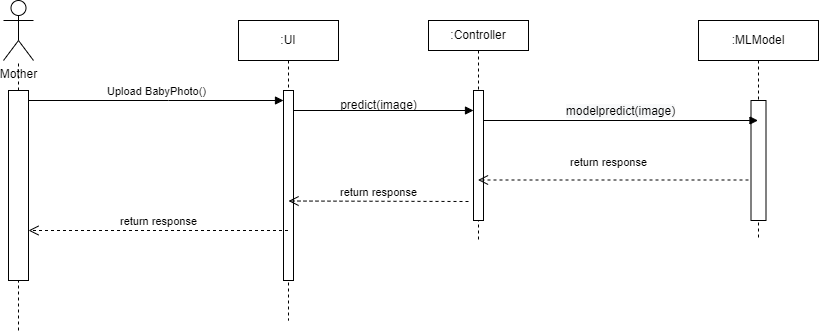
****

Figure 16. Down syndrome model Design Seq. Diagram

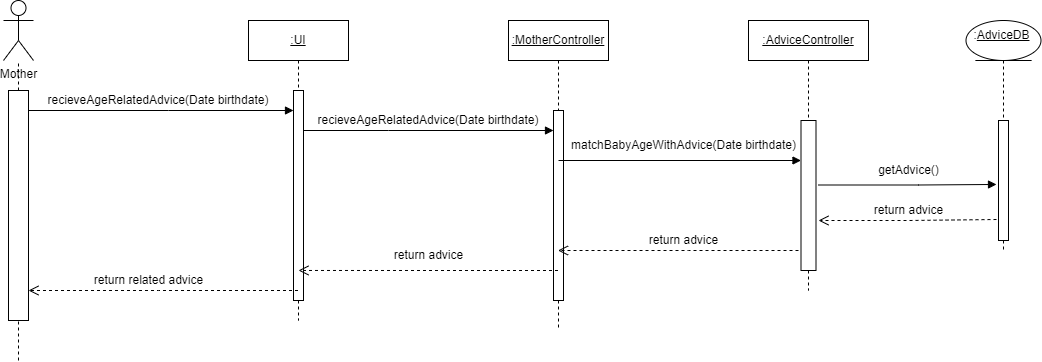
****

Figure 17. Show Age-Related Advice Design Seq. Diagram

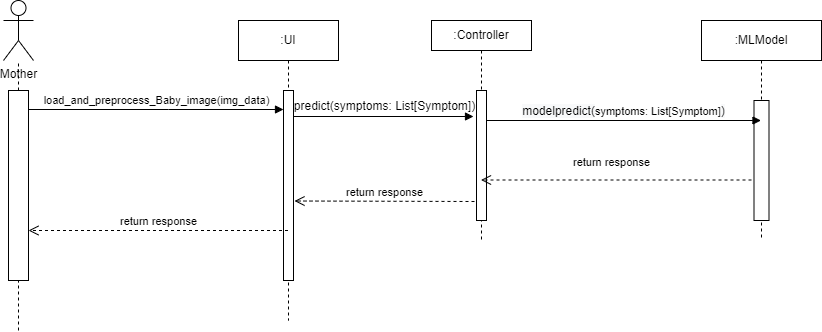


Figure 18. Symptom Disease model Design Seq. Diagram

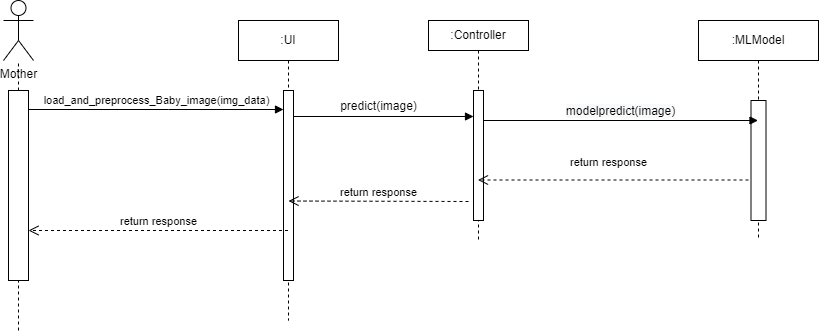
****

Figure 19. Jaundice Detection model Design Seq. Diagram

4.3 Main Techniques, Technologies, and Application

The technologies and techniques and their respective applications that are used to build the system are:

**Node.js:** It is a toolkit that provides features that enable us to build Restful API, it provides serialization, authentication, authorization as well as security.

Application: Backend of the system.

**Flutter:** An open-source framework developed by Google. It is used for building multiplatform applications from the same code. Therefore, we could use it to develop an application that runs on both android and IOS with the same code. It can also be used for websites’ user interface development.

Application: Frontend for mobile.

**MongoDB:** It is an open source [NoSQL](https://www.techtarget.com/searchdatamanagement/definition/NoSQL-Not-Only-SQL) database management program. It supports application features such as JSON, XML as well as various data types. It is suitable for large systems or systems that have large dataset also, it is secure, reliable and is fast for both read and write operations.

Application: Database management and queries.

4.4 System Architecture

A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system.

**User interface**

The UI layer was built using the Flutter framework for a responsive and cross-platform user experience.

**Application layer**

Implement Node.js services for user authentication and profile management. Create a service to manage health metrics tracking (weight, height, and vaccination schedules)

**Data management**

MongoDB is employed as the database system for storing user information, articles and advice, vaccine information and other relevant data.

**Communication and integration**

API requests and responses, connecting the Flutter frontend with the Node.js backend services.



A diagram of a server

Description automatically generated



Figure 20. Data flow

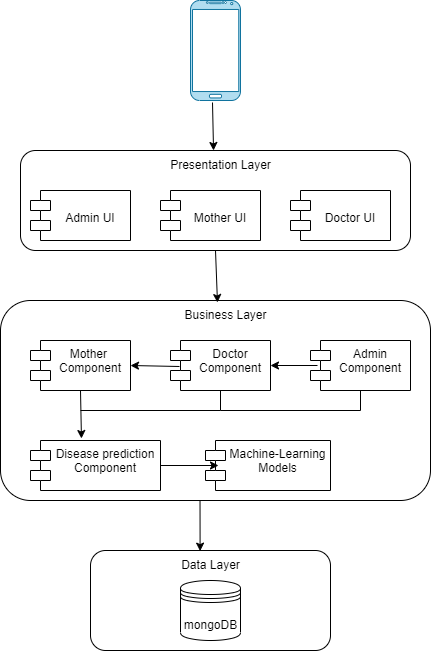
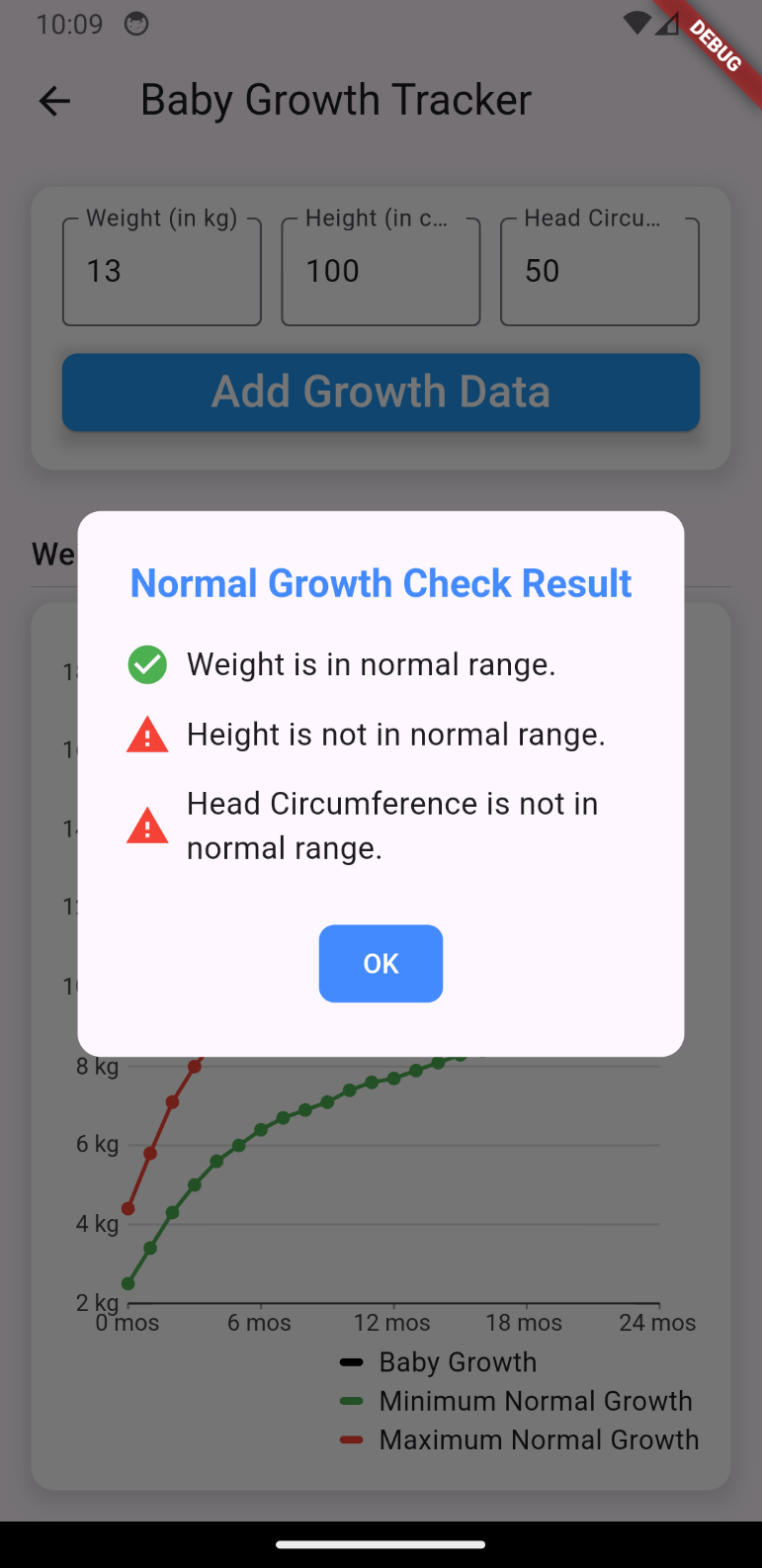


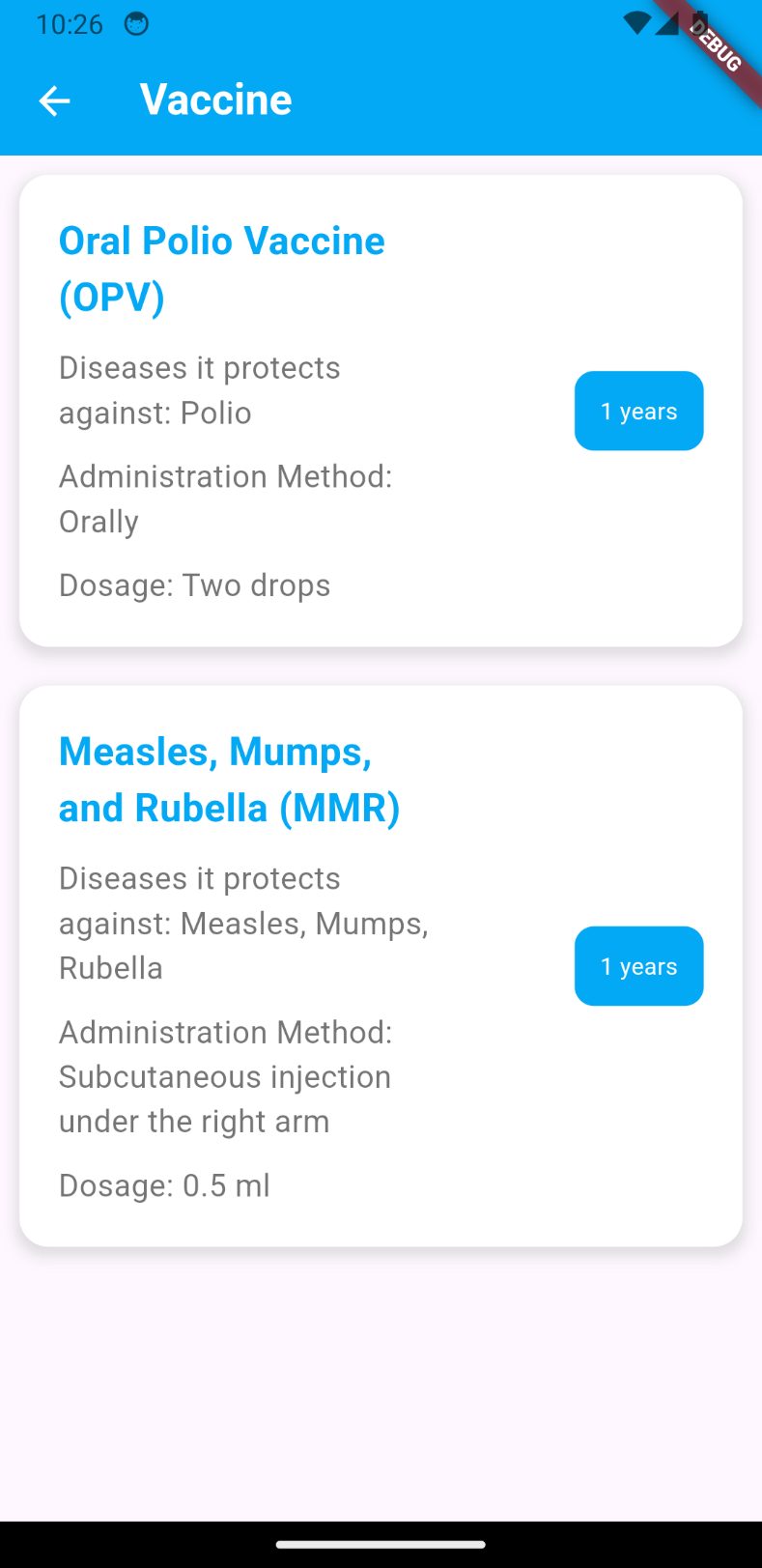


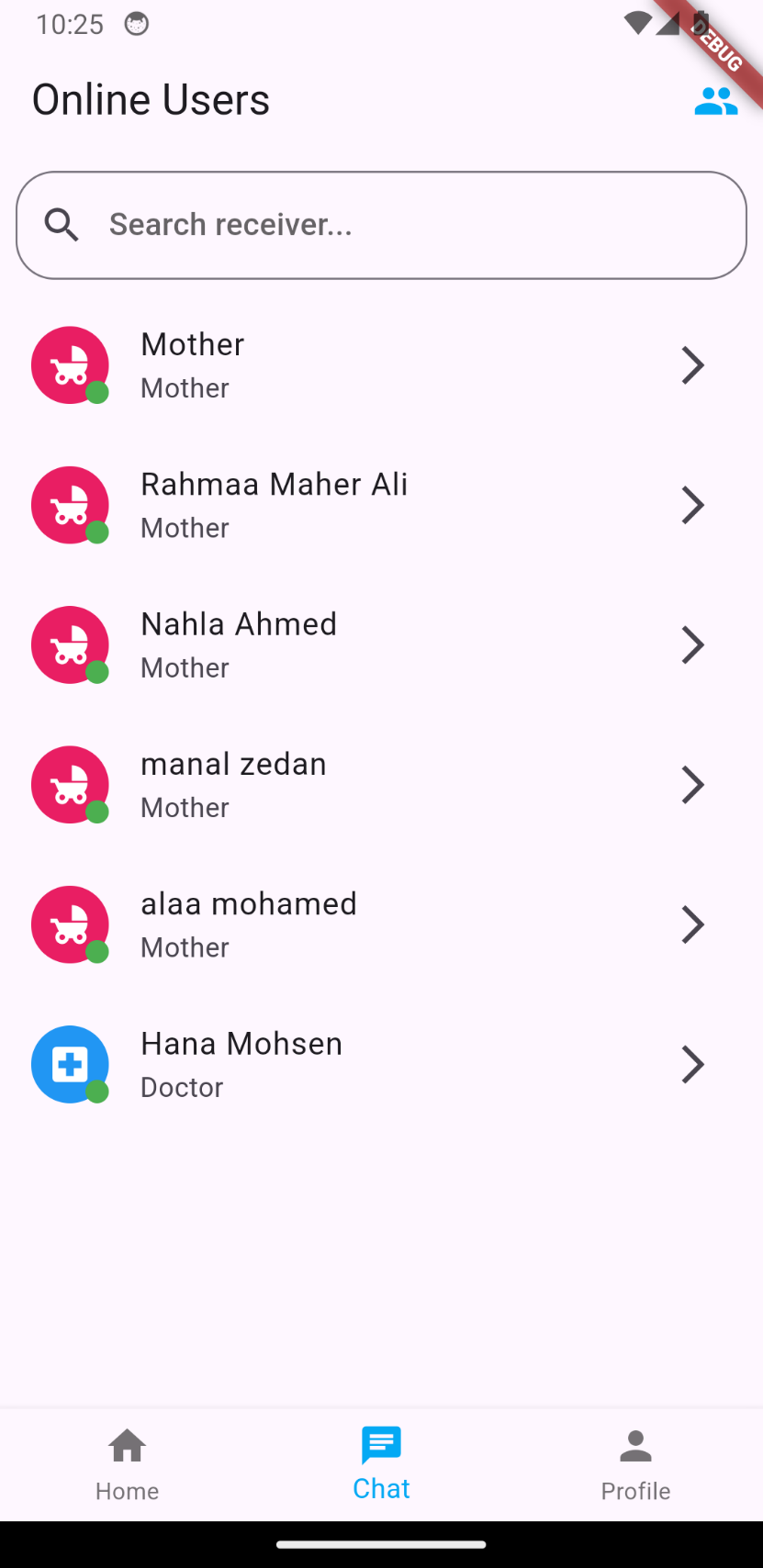
Figure 21. System Overview

4.5 Prototype









# Chapter 5: Testing and Evaluation

5.1 Testing and Evaluation

|  |  |
| --- | --- |
| **Test Case**  **Description** | The administrator wants to add advice so he adds its title, content and age. |
| **Test Scenario** | The system returns if the advice is created successfully or not. |
| **Post** | http://10.0.2.2:9000/api/v1/advice |
| **Test Data** | {      "title": "Stay Hydrated",      "content": "Drink at least 8 glasses of water a day.",      "age": 3  } |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {  "status": "success",  "data": {  "title": "Stay Hydrated",  "content": "Drink at least 8 glasses of water a day.",  "age": 3,  "\_id": "668710eab9aec21c0ac08061"  }  } |
| **Processing Time**  **and Status Code** | Time: 154 ms  Status Code: 201 Created |
| **Pass/Fail** | Pass |

In this section, we will be providing API testing results using postman application, we will also be evaluating the performance of each API according to the time it takes to process the API request and return the expected results.

**1- Create Advice**

Table 9. Testing result 1

**2- Get Advice**

|  |  |
| --- | --- |
| **Test Case**  **Description** | The user wants to get all advice stored at the system. |
| **Test Scenario** | The system returns all the advice stored. |
| **Get** | http://10.0.2.2:9000/api/v1/advice |
| **Test Data** | No collections are sent in the request body. |
| **Expected Result** | A success message and the data should be returned. (PS: we showed here only 2 advice from the 128 that came in the results.) |
| **Actual Result** | {      "status": "success",      "results": 128,      "data": [          {              "\_id": "6679704747ba12f17df54c29",              "title": "Stay Hydrated",              "content": "Drink at least 8 glasses of water a day.",              "age": 3          },          {              "\_id": "6679712647ba12f17df54c2b",              "title": "Feeding",              "content": "Ensure your baby is feeding well, whether breastfeeding or formula-feeding. Look for signs of hunger and fullness.",              "age": 1          }  } |
| **Processing Time**  **and Status Code** | Time: 673 ms  Status Code: 200 OK |
| **Pass/Fail** | Pass |

Table 10. Testing result 2

**3- Get Advice by age**

|  |  |
| --- | --- |
| **Test Case**  **Description** | The user wants to get advice for a specific age. |
| **Test Scenario** | The system returns all the advice stored for this specific age sent at the parameters. |
| **Get** | http://10.0.2.2:9000/api/v1/advice/age/1 |
| **Test Data** | No collections are sent in the request body. |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {    "success": **true**,      "data": [          {              "\_id": "6679712647ba12f17df54c2b",              "title": "Feeding",              "content": "Ensure your baby is feeding well, whether breastfeeding or formula-feeding. Look for signs of hunger and fullness.",              "age": 1          },          {              "\_id": "6679716047ba12f17df54c2d",              "title": "Sleeping",              "content": "Newborns typically sleep 16-18 hours a day. Establish a sleep routine with a quiet environment.",              "age": 1  }  } |
| **Processing Time**  **and Status Code** | Time: 140 ms  Status Code: 200 OK |
| **Pass/Fail** | Pass |

Table 11. Testing result 3

|  |  |
| --- | --- |
| **Test Case**  **Description** | The doctor wants to add appointment that contains his address, fee and date. |
| **Test Scenario** | The system returns if the appointment is added successfully or not. |
| **Post** | http://10.0.2.2:9000/api/v1/appointment |
| **Test Data** | {    "address": "123 Main St Maadi",    "fee": 700,    "date": "2024-07-29T10:30:00.000Z",    "doctor": "65f871a6795b1ec90a7a544f"  } |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {  "success": true,  "data": {  "address": "123 Main St Maadi",  "fee": 700,  "date": "2024-07-29T10:30:00.000Z",  "doctor": "65f871a6795b1ec90a7a544f",  "\_id": "6687174a9eecffe794e732e2"  }  } |
| **Processing Time**  **and Status Code** | Time: 188 ms  Status Code: 201 Created |
| **Pass/Fail** | Pass |

**4- Create appointment**

Table 9. Testing result 1

Table 12. Testing result 4

|  |  |
| --- | --- |
| **Test Case**  **Description** | The mother wants to get appointments for a specific filtration she chooses from (address, fee, and doctor name). |
| **Test Scenario** | The system returns all the available appointments related to this specific filtration sent at the body. |
| **Post** | http://10.0.2.2:9000/api/v1/appointment/search |
| **Test Data** | {    "address": "123 Main St Maadi",    "doctorName": "Hanaa”  } |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {    "success": **true**,      "data": [          {              "\_id": "6687174a9eecffe794e732e2",              "address": "123 Main St Maadi",              "fee": 700,              "date": "2024-07-29T10:30:00.000Z",              "doctor": {                  "\_id": "65f871a6795b1ec90a7a544f",                  "firstName": "Hanaa",                  "lastName": "Hany",                  "phone": "01155464646",                  "specialization": "nutrition",                  "degree": "Specialist",                  "workPlace": "9st madyyy elmarghany" }  } |
| **Processing Time**  **and Status Code** | Time: 299 ms  Status Code: 200 OK |
| **Pass/Fail** | Pass |

**5- Get Filtered Appointments**

Table 13. Testing result 5

|  |  |
| --- | --- |
| **Test Case**  **Description** | The User browse the articles. |
| **Test Scenario** | The system allows the user to browse through articles available on the platform. Each article is categorized and contains relevant content based on the chosen category. |
| **Get** | http://10.0.2.2:9000/api/v1/article/arabic ?category=BreadFeeding |
| **Test Data** | No collections are sent in the request body. |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {      "status": "success",      "data": [          {              "\_id": "668925a309a7b81ea8f0648e",              "title": "كيف تطعم طفلك",              "category": "الرضاعة الطبيعية",              "content": "أثناء إطعام طفلك ، قم بإيقاف تشغيل التلفزيون واترك كل ما عليك هو انتباه طفلك ، امسك طفلك أو الجلوس مع طفلك ، والتحدث مع طفلك ، فإن التغذية وقت خاص لكي يستمتع اثنان منكم بالتعرف على بعضهما البعض.",              "date": "2024-07-06T11:08:19.832Z",          },      ]  } |
| **Processing Time**  **and Status Code** | Time: 10.67 s  Status Code: 200 OK |
| **Pass/Fail** | Pass |

**6- Get Articles in Arabic**

Table 14. Testing result 6

|  |  |
| --- | --- |
| **Test Case**  **Description** | The user wants to get all advice stored at the system for a specific age. |
| **Test Scenario** | The system returns all the advice stored for this specific age sent at the parameters. |
| **Get** | http://10.0.2.2:9000/api/v1/advice/arabic/age/1 |
| **Test Data** | No collections are sent in the request body. |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {      "success": **true**,      "data": [          {              "\_id": "6679712647ba12f17df54c2b",              "title": "تغذية",              "content": "تأكد من أن طفلك يتغذى بشكل جيد ، سواء كان الرضاعة الطبيعية أو الرضاعة الحية.ابحث عن علامات الجوع والامتلاء.",              "age": 1          },          {              "\_id": "6679716047ba12f17df54c2d",              "title": "نائم",              "content": "ينام حديثي الولادة عادة 16-18 ساعة في اليوم.إنشاء روتين النوم مع بيئة هادئة.",              "age": 1          },          {              "\_id": "6679719c47ba12f17df54c2f",              "title": "صحة",              "content": "جدولة وحضور أول زيارة لطب الأطفال.مراقبة اليرقان وظروف حديثي الولادة الأخرى.",              "age": 1          }      ]  } |
| **Processing Time**  **and Status Code** | Time: 3.74 s  Status Code: 200 OK |
| **Pass/Fail** | Pass |

**7- Get Advice in Arabic**

Table 15. Testing result 7

**8- Tracks baby's growth**

|  |  |
| --- | --- |
| **Test Case**  **Description** | Mother tracks baby's growth. The system returns if the baby's growth within the normal range or not. |
| **Test Scenario** | The system evaluates the baby growth parameters (gender, age, weight, height and head circumference) to determine if they are within normal ranges. |
| **Post** | http://10.0.2.2:9000/api/v1/normalGrowth/check |
| **Test Data** | {      "gender": "male",      "ageInMonths": "12",      "weight":7,      "height":70,      "headCircumference":15  } |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {      "success": **true**,      "data": {          "weight": "Weight is not in normal range",          "height": "Height is not in normal range",          "headCircumference": "Head Circumference is not in normal range"      }  } |
| **Processing Time**  **and Status Code** | Time: 260 ms  Status Code: 201 Created |
| **Pass/Fail** | Pass |

Table 16. Testing result 8

**9- Upload Article**

|  |  |
| --- | --- |
| **Test Case**  **Description** | Doctor adds article to a specific category. |
| **Test Scenario** | The application allows a doctor to add an article to the platform. |
| **Post** | http:// 10.0.2.2:9000/api/v1/article |
| **Test Data** | {  "title": "How to feed your baby",  "category": "Breastfeeding",  "content": "While feeding your baby, turn off the TV and give your baby all your attention, Hold your baby or sit with your baby, and talk to your baby, Feeding is a special time for the two of you to enjoy getting to know each other."  } |
| **Expected Result** | A success message should be returned, and article should be added. |
| **Actual Result** | {  "success": true,  "data": {  "title": "How to feed your baby",  "category": "Breastfeeding",  "content": "While feeding your baby, turn off the TV and give your baby all your attention, Hold your baby or sit with your baby, and talk to your baby, Feeding is a special time for the two of you to enjoy getting to know each other.",  "token": "",  "\_id": "668925a309a7b81ea8f0648e",  "date": "2024-07-06T11:08:19.832Z",  "createdAt": "2024-07-06T11:08:19.839Z",  "updatedAt": "2024-07-06T11:08:19.839Z",  "\_\_v": 0  }  } |
| **Processing Time**  **and Status Code** | Time: 250 ms  Status Code: 201 Created |
| **Pass/Fail** | Pass |

Table 17. Testing result 9

**10- Get Articles**

|  |  |
| --- | --- |
| **Test Case**  **Description** | Mother browse articles. |
| **Test Scenario** | The system allows the mother to browse through articles available on the platform. Each article is categorized and contains relevant content based on the chosen category. |
| **Get** | http:// 10.0.2.2:9000/api/v1/article /english?category= BreastFeeding |
| **Test Data** | No collections are sent in the request body. |
| **Expected Result** | A success message and the articles should be returned. |
| **Actual Result** | {  "success": true,  "data": {  "title": "How to feed your baby",  "category": "Breastfeeding",  "content": "While feeding your baby, turn off the TV and give your baby all your attention, Hold your baby or sit with your baby, and talk to your baby, Feeding is a special time for the two of you to enjoy getting to know each other.",  "token": "",  "\_id": "668925a309a7b81ea8f0648e",  "date": "2024-07-06T11:08:19.832Z",  "createdAt": "2024-07-06T11:08:19.839Z",  "updatedAt": "2024-07-06T11:08:19.839Z",  "\_\_v": 0  }  } |
| **Processing Time**  **and Status Code** | Time: 194 ms  Status Code: 200 OK |
| **Pass/Fail** | Pass |

Table 18. Testing result 10

**11- Get all vaccine centers**

|  |  |
| --- | --- |
| **Test Case**  **Description** | Get all vaccine centers stored. |
| **Test Scenario** | The system retrieves vaccine center information based on the search criteria. |
| **Get** | http://10.0.2.2:9000/api/v1/vaccineCenter/ |
| **Test Data** | No collections are sent in the request body. |
| **Expected Result** | A success message and the vaccine centers should be returned. |
| **Actual Result** | {  "status": "success",  "results": 36,  "data": {  " vaccineCenters ": [  {  "\_id": "667de51c32629f9c6017d5f6",  "name": "Health Office - Abou El Seoud",  "address": "1 El Fostat St. off Salah Salem, Misr El Kadima, Cairo.",  "phone": "02-2741-6358",  }  ]  }  } |
| **Processing Time**  **and Status Code** | Time: 160 ms  Status Code: 200 OK |
| **Pass/Fail** | Pass |

Table 19. Testing result 11

**12- Get Vaccine Centers in a City**

|  |  |
| --- | --- |
| **Test Case**  **Description** | Search for vaccine center by city. |
| **Test Scenario** | The system retrieves vaccine center information based on the city. |
| **Get** | http://10.0.2.2:9000/api/v1/vaccineCenter/inCity/?address=Misr El Kadima |
| **Test Data** | No collections are sent in the request body. |
| **Expected Result** | A success message and the vaccine centers in the determined city should be returned. |
| **Actual Result** | {      "status": "success",      "results": 1,      "data": {          "vaccineCenters": [              {                  "\_id": "667de51c32629f9c6017d5f6",                  "name": "Health Office - Abou El Seoud",                  "address": "1 El Fostat St. off Salah Salem, Misr El Kadima, Cairo.",                  "phone": "02-2741-6358",              }          ]      }  } |
| **Processing Time**  **and Status Code** | Time: 165 ms  Status Code: 200 OK |
| **Pass/Fail** | Pass |

Table 20. Testing result 12

|  |  |
| --- | --- |
| **Test Case**  **Description** | The mother or the doctor wants to send a message, providing the sender, receiver, and the message content. |
| **Test Scenario** | The system returns whether the message is sent successfully or not. |
| **Post** | http://10.0.2.2:9000/api/v1/chat/send |
| **Test Data** | {      "sender": "66881a1046b4c707e5b32a47",      "senderModel": "Mother",      "receiver": "666ca13e1396ea29382310f6",      "receiverModel": "Doctor",      "message": "Hello, how are you?"  } |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {      "success": **true**,      "data": {          "sender": "66881a1046b4c707e5b32a47",          "senderModel": "Mother",          "receiver": "666ca13e1396ea29382310f6",          "receiverModel": "Doctor",          "message": "Hello, how are you?",          "isRead": **false**,          "\_id": "66881c1b46b4c707e5b32a4c",          "timestamp": "2024-07-05T16:15:23.111Z",      }  } |
| **Processing Time**  **and Status Code** | Time: 190 ms  Status Code: 201 Created |
| **Pass/Fail** | Pass |

**13- Chat (send messages)**

Table 21. Testing result 13

**14- Chat (get chat history)**

|  |  |
| --- | --- |
| **Test Case**  **Description** | Fetch chat history between a Mother and a Doctor. |
| **Test Scenario** | The system retrieves chat history based on the user's ID, user model, contact's ID, and contact model. |
| **Get** | http:// 10.0.2.2:9000/api/v1/chat/history?userId=66881a1046b4c707e5b32a47&  userModel=Mother&contactId=666ca13e1396ea29382310f6&contactModel=Doctor |
| **Test Data** | No collections are sent in the request body. |
| **Expected Result** | A success message. |
| **Actual Result** | {      "success": **true**,      "chatHistory": [          {              "\_id": "66881ae446b4c707e5b32a49",              "sender": "66881a1046b4c707e5b32a47",              "senderModel": "Mother",              "receiver": "666ca13e1396ea29382310f6",              "receiverModel": "Doctor",              "message": "Hello, how are you?",              "isRead": **false**,              "timestamp": "2024-07-05T16:10:12.826Z",              "createdAt": "2024-07-05T16:10:12.838Z",              "updatedAt": "2024-07-05T16:10:12.838Z",              "\_\_v": 0          }  } |
| **Processing Time**  **and Status Code** | Time: 322 ms  Status Code: 200 OK |
| **Pass/Fail** | Pass |

Table 22. Testing result 14

**15- Add Monthly Measurement for Baby**

|  |  |
| --- | --- |
| **Test Case**  **Description** | The user wants to add monthly measurements for a baby. |
| **Test Scenario** | The system should store and return the added monthly measurement for the baby. |
| **Post** | http://10.0.2.2:9000/api/v1/baby/measurements/6681f73e8f5bf49890d037ad |
| **Test Data** | {  "month": 0,  "height": 46.5,  "weight": 3,  "headCircumference": 33  } |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {  "status": "success",  "data": {  "\_id": "6681f73e8f5bf49890d037ad",  "firstname": "Omar",  "lastname": "Hassan",  "birthdate": "2024-03-20T00:00:00.000Z",  "gender": "Male",  "measurements": [  {  "month": 0,  "height": 46.5,  "weight": 3,  "headCircumference": 33,  "\_id": "66880331934f588c9d844794"  }  ],  }  } |
| **Processing Time**  **and Status Code** | Time: 635 ms  Status Code: 201 Created |
| **Pass/Fail** | Pass |

Table 23. Testing result 15

**16- Get All Monthly Measurements for Baby**

|  |  |
| --- | --- |
| **Test Case**  **Description** | The user wants to get all monthly measurements stored for a baby. |
| **Test Scenario** | The system returns all the monthly measurements stored. |
| **Get** | http://10.0.2.2:9000/api/v1/baby/measurements/6681f73e8f5bf49890d037ad |
| **Test Data** | No collections are sent in the request body. |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {  "status": "success",  "data": [  {  "month": 0,  "height": 46.5,  "weight": 3,  "headCircumference": 33,  "\_id": "66880331934f588c9d844794"  }  ]  } |
| **Processing Time**  **and Status Code** | Time: 313 ms  Status Code: 200 OK |
| **Pass/Fail** | Pass |

Table 24. Testing result 16

**17- Get All Doctors**

|  |  |
| --- | --- |
| **Test Case**  **Description** | The user wants to get all doctors stored in the system. |
| **Test Scenario** | The system returns all the doctors stored. |
| **Get** | http://10.0.2.2:9000/api/v1/doctor/all |
| **Test Data** | No collections are sent in the request body. |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {  "status": "success",  "data": [  {  "\_id": "65f871a6795b1ec90a7a544f",  "firstName": "Hanaa",  "lastName": "Hany",  "phone": "01155464646",  "specialization": "nutrition",  "degree": "Specialist",  "workPlace": "9st madyyy elmarghany",  "email": menna7@gmail.com,  "online": false,  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  eyJlbWFpbCI6Im1lbm5hN0BnbWFpbC5jb20iLCJpZCI6IjY1Zjg3MWE2Nzk1YjFlYzkwYTdhNTQ0ZiIsImlhdCI6MTcxMDc4MDg3Mn0.RJrPDVse9NxeQmB9RPcAfTHpooXu4ZpkQAaZ9UOiQpY"  },  {  "\_id": "666ca13e1396ea29382310f6",  "firstName": "Nora",  "lastName": "NaDer",  "phone": "0122456789",  "specialization": "Pediatrics",  "degree": "MD",  "workPlace": "City Hospital",  "description": "Experienced pediatrician with over 10 years of practice.",  "email": "noor@gmail.com",  "online": false,  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  eyJlbWFpbCI6Im5vb3JAZ21haWwuY29tIiwiaWQiOiI2NjZjYTEzZTEzOTZlYTI5MzgyMzEwZjYiLCJpYXQiOjE3MjAxMjkzNzR9.L5-CpUU-SmkcMVtfKun0BppJJQzmjaoNo7DoRpENLlE"  },  ]  } |
| **Processing Time**  **and Status Code** | Time: 232 ms  Status Code: 200 OK |
| **Pass/Fail** | Pass |

Table 25. Testing result 17

**18-** **Get Online Doctors**

|  |  |
| --- | --- |
| **Test Case**  **Description** | The user wants to get all online doctors stored in the system. |
| **Test Scenario** | The system returns all the online doctors stored. |
| **Get** | http://10.0.2.2:9000/api/v1/doctor/online |
| **Test Data** | No collections are sent in the request body. |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {  "success": true,  "data": [  {  "\_id": "668596e4b564756cbfce4d76",  "firstName": "Hana",  "lastName": "Mohsen",  "phone": "01565464644",  "specialization": "nutrition",  "degree": "Specialist",  "workPlace": "9st maadi elmarghany",  "online": true,  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  eyJlbWFpbCI6ImhhbmE0QGdtYWlsLmNvbSIsImlkIjoiNjY4NTk2ZTRiNTY0  NzU2Y2JmY2U0ZDc2IiwiaWF0IjoxNzIwMTU0MTM3fQ.QwqN89YMVT4Zr\_hss-Bnbba29lo2YDSbMHab6KKUBVY",  }  ]  } |
| **Processing Time**  **and Status Code** | Time: 177 ms  Status Code: 200 OK |
| **Pass/Fail** | Pass |

Table 26. Testing result 18

**19-** **Register Doctor**

|  |  |
| --- | --- |
| **Test Case**  **Description** | The doctor wants to register in the system. |
| **Test Scenario** | The system registers the new doctor and returns the doctor details. |
| **Post** | http://10.0.2.2:9000/api/v1/doctor/ |
| **Test Data** | {  "firstName":"Aziz",  "lastName":"Hamdy",  "phone":"01565464644",  "specialization":"Pediatrics",  "degree":"Specialist",  "workPlace":"9st madyyy elmarghany",  "email":"aziz33@gmail.com",  "password":"1234567"  } |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {  "status": "success",  "data": {  "firstName": "Aziz",  "lastName": "Hamdy",  "phone": "01565464644",  "specialization": "Pediatrics",  "degree": "Specialist",  "workPlace": "9st madyyy elmarghany",  "email": "aziz33@gmail.com",  "password": "$2a$10$O2pSayJ2qFXrlku2ZzAj8OoBQ4t2xpDO/VhzjuC2qLtwH5NuqZaD.",  "online": false,  "\_id": "668809dc934f588c9d84479a",  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  eyJlbWFpbCI6ImF6aXozM0BnbWFpbC5jb20iLCJpZCI6IjY2ODgwOWRjOTM0ZjU4OGM5ZDg0ND  c5YSIsImlhdCI6MTcyMDE5MTQ1Mn0.L7GU\_c2OMVMzdsEgwSPEWwhBEAiifLsb0RYjeukr2Mw",  }  } |
| **Processing Time**  **and Status Code** | Time: 371 ms  Status Code: 201 Created |
| **Pass/Fail** | Pass |

Table 27. Testing result 19

**20-** **Create Mother**

Table 28. Testing result 20

|  |  |
| --- | --- |
| **Test Case**  **Description** | The mother wants to create profile in the system. |
| **Test Scenario** | The system returns if the mother profile is created successfully or not. |
| **Post** | http://10.0.2.2:9000/api/v1/mother |
| **Test Data** | {  "name":"manal zedan",  "email":"manal23@gmail.com",  "password":"19990",  "address":"23st Maadi"  } |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {  "success": true,  "data": {  "name": "manal zedan",  "email": "manal23@gmail.com",  "password": "$2a$07$lh9aHsmJxVOKv/enpVXyfOc53tdxtOiOoE8ZBPy8CvDkzQ5Y5F2F2",  "address": "23st Maadi",  "babies": [],  "currentIndex": 0,  "online": false,  "\_id": "66881103934f588c9d84479d",  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  eyJlbWFpbCI6Im1hbmFsMjNAZ21haWwuY29tIiwiaWQiOiI2Njg4MTEw  MzkzNGY1ODhjOWQ4NDQ3OWQiLCJpYXQiOjE3MjAxOTMyODN9.6TYErEAz-RCmGdAX7RiRINcg-tCEDVZF9AucOs1swpo",  }  } |
| **Processing Time**  **and Status Code** | Time: 208 ms  Status Code: 201 Created |
| **Pass/Fail** | Pass |

**21- Get All Mothers**

|  |  |
| --- | --- |
| **Test Case**  **Description** | The user wants to retrieve all mother profiles stored in the system. |
| **Test Scenario** | The system returns all mother profiles stored. |
| **Get** | http://10.0.2.2:9000/api/v1/mother/all |
| **Test Data** | No collections are sent in the request body. |
| **Expected Result** | A success message and the data should be returned. (PS: we showed here only 2 advice from the 128 that came in the results.) |
| **Actual Result** | {  "success": true,  "data": [  {  "\_id": "6681c9bc3fc27bd36856fc0c",  "name": "Wafaa Magdy",  "email": "wafaa@gmail.com",  "password": "$2a$07$CS8/fGFA0i0Fj2UMCAUtK.1zx8FQE491KfvWGyGjlKCrQJedap1xm",  "address": "13st Mahran Kamel,Maadi,Cairo",  "babies": [],  "online": false,  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  eyJlbWFpbCI6IndhZmFhQGdtYWlsLmNvbSIsImlkIjoiNjY4MWM5YmMzZm  MyN2JkMzY4NTZmYzBjIiwiaWF0IjoxNzE5NzgxODIwfQ.t5kFqNpHkzuvM6gBja\_w  lQ7ZEEQbp89gyW4tTrUpNk",  },  {  "\_id": "668214f2cbc623477dbc7912",  "name": "Safaa Osama",  "email": "Safaa@gmail.com",  "password": "$2a$07$nG92Oj8PoJ7uVeFyr7y0CuMQEkGrSbK60w9RY4UsA3sYDzQm8YVQ6",  "address": "9st Ahmed Badran,Maadi,Cairo",  "babies": [  {  "\_id": "668214f2cbc623477dbc7913",  "firstname": "Omar",  "lastname": "Hassan",  "birthdate": "2024-03-20T00:00:00.000Z",  "gender": "Male",  "measurements": [],  "createdAt": "2024-07-01T02:31:14.358Z",  "updatedAt": "2024-07-01T02:31:14.358Z",  "\_\_v": 0  }  ],  "currentIndex": 0,  "online": false,  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  eyJlbWFpbCI6IlNhZmFhQGdtYWlsLmNvbSIsImlkIjoiNjY4MjE0ZjJjYm  M2MjM0NzdkYmM3OTEyIiwiaWF0IjoxNzE5ODAxMDc0fQ.FeiD67X4zs  PlYhb1GcF8FAgwivbj62WpeMfo\_h1LaE4",  }  ]  } |
| **Processing Time**  **and Status Code** | Time: 415 ms  Status Code: 200 OK |
| **Pass/Fail** | Pass |

Table 29. Testing result 21

**22- Get Online Mothers**

|  |  |
| --- | --- |
| **Test Case**  **Description** | The user wants to retrieve all online mother profiles stored in the system. |
| **Test Scenario** | The system returns all online mother profiles stored. |
| **Get** | http://10.0.2.2:9000/api/v1/mother/online |
| **Test Data** | No collections are sent in the request body. |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {  {  "success": true,  "data": [  {  "\_id": "66817562cffc038a9744e099",  "name": "Mother",  "email": "walaa@example.com",  "password": "$2a$10$TJbV/Va.af4DqRzMJW6cU.Z9wWY8c6l2kYfuskSk6iLfhOcJbitje",  "babies": [],  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  eyJlbWFpbCI6IndhbGFhQGV4YW1wbGUuY29tIiwiaWQiOiI2NjgxNzU2Mm  NmZmMwMzhhOTc0NGUwOTkiLCJpYXQiOjE3MTk3NjAyMjYsImV4cCI6MT  cxOTc2MzgyNn0.iIddHMXPw-oF6LZWbtuh07BvVCS76pn-lDonh2c-9FA",  "online": true  },  {  "\_id": "66821c1c0dfc9e211611edd6",  "name": "RaHmAA MaheR",  "email": "rahma@gmail.com",  "password": "$2a$10$HeoUq.BLX6RCXW411B735.LMuuth1BMcCbhtOJZG/lBUjGuRJ65k6",  "address": "15st Ezz El-deen Dahshan , Maadi,Cairo",  "babies": [  "66821c3d0dfc9e211611edd9",  "66821d230dfc9e211611ede3",  "66876b7fe94898c29bcbf179"  ],  "currentIndex": 1,  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  eyJlbWFpbCI6InJhaG1hQGdtYWlsLmNvbSIsImlkIjoiNjY4MjFjM  WMwZGZjOWUyMTE2MTFlZGQ2IiwiaWF0IjoxNzIwMTkwNjU4fQ.  R9MAESJrOvG48IQmYU1SZVgje7n5U8y5695-u0yWPbo",  "online": true  }  ]  } |
| **Processing Time**  **and Status Code** | Time: 142 ms  Status Code: 200 OK |
| **Pass/Fail** | Pass |

Table 30. Testing result 22

**23-** **Add Baby to Mother**

|  |  |
| --- | --- |
| **Test Case**  **Description** | The mother wants to add a baby to her profile. |
| **Test Scenario** | The system returns if the baby is successfully added to the mother's profile. |
| **Post** | http://10.0.2.2:9000/api/v1/mother/66881103934f588c9d84479d/babies |
| **Test Data** | {  "firstname":"laila",  "lastname":"ahmed",  "birthdate": "2024-05-10",  "gender": "female"  } |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {  "success": true,  "data": {  "\_id": "66881103934f588c9d84479d",  "name": "manal zedan",  "email": "manal23@gmail.com",  "password": "$2a$07$lh9aHsmJxVOKv/enpVXyfOc53tdxtOiOoE8ZBPy8CvDkzQ5Y5F2F2",  "address": "23st Maadi",  "babies": [  {  "\_id": "668815df934f588c9d8447d3",  "firstname": "laila",  "lastname": "ahmed",  "birthdate": "2024-05-10T00:00:00.000Z",  "gender": "female",  "measurements": [],  }  ],  "currentIndex": 0,  "online": false,  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  eyJlbWFpbCI6Im1hbmFsMjNAZ21haWwuY29tIiwiaWQiOiI2Njg4MTEwMzkzNGY1ODhjOWQ4NDQ3OWQiLCJpYXQiOjE3MjAxOTMyODN9.6TYErEAz-RCmGdAX7RiRINcg-tCEDVZF9AucOs1swpo" } } |
| **Processing Time**  **and Status Code** | Time: 836 ms  Status Code: 201 Created |
| **Pass/Fail** | Pass |

Table 31. Testing result 23

**24-** **Get All Babies for Mother**

Table 32. Testing result 24

|  |  |
| --- | --- |
| **Test Case**  **Description** | The user wants to retrieve all babies associated with a specific mother stored in the system. |
| **Test Scenario** | The system returns all the babies associated with the specified mother. |
| **Get** | http://10.0.2.2:9000/api/v1/mother/66881103934f588c9d84479d/babies |
| **Test Data** | No collections are sent in the request body. |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {  "success": true,  "data": [  {  "\_id": "668815df934f588c9d8447d3",  "firstname": "laila",  "lastname": "ahmed",  "birthdate": "2024-05-10T00:00:00.000Z",  "gender": "female",  "measurements": [],  }  ]  } |
| **Processing Time**  **and Status Code** | Time: 605 ms  Status Code: 200 OK |
| **Pass/Fail** | Pass |

**25- Update Mother Online Status**

|  |  |
| --- | --- |
| **Test Case**  **Description** | The user wants to update the online status of a mother in the system. |
| **Test Scenario** | The system updates the online status of the specified mother. |
| **Put** | http://10.0.2.2:9000/api/v1/mother/66881103934f588c9d84479d/status |
| **Test Data** | {  "online":true  } |
| **Expected Result** | A success message and the data should be returned. |
| **Actual Result** | {  "success": true,  "data": "Mother online status updated successfully",  "mother": {  "\_id": "66881103934f588c9d84479d",  "name": "manal zedan",  "email": "manal23@gmail.com",  "password": "$2a$07$lh9aHsmJxVOKv/enpVXyfOc53tdxtOiOoE8ZBPy8CvDkzQ5Y5F2F2",  "address": "23st Maadi",  "babies": [  "668815df934f588c9d8447d3"  ],  "currentIndex": 0,  "online": true,  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  eyJlbWFpbCI6Im1hbmFsMjNAZ21haWwuY29tIiwiaWQiOiI2Njg4  MTEwMzkzNGY1ODhjOWQ4NDQ3OWQiLCJpYXQiOjE3MjAxOTMy  ODN9.6TYErEAz-RCmGdAX7RiRINcg-tCEDVZF9AucOs1swpo"  }  } |
| **Processing Time**  **and Status Code** | Time: 977 ms  Status Code: 200 OK |
| **Pass/Fail** | Pass |

Table 33. Testing result 25

5.2 Development

This section will include the developing tools we used and some segments of the code to show the logic of the complicated APIs.

### **5.2.1 Tools Used for Development**

* Frontend tools:  
   - Visual Studio Code.
* Backend tools:

- Visual Studio Code.

- Postman.

-Mongo DB compass.

### 5.2.2 Screens and Code Segments

// Function to get advices in Arabic

// Node.js Server

exports.getAdvicesInArabic = async (req, res) => {

    const age = req.params.age; // Get the age from the request parameters

    try {

        // Fetch all advices (assuming stored in the original language)

        const advices = await Advice.find({ age: age }); // Filter by age

        // Translate each advice's title, content, and possibly category to Arabic

        const translatedAdvices = await Promise.all(advices.map(async (advice) => {

            let translatedTitle = advice.title;

            let translatedContent = advice.content;

            // Translate title

            if (advice.title) {

                translatedTitle = await translate(advice.title, { to: 'ar' }).catch(error => {

                    console.error(`Error translating title: ${error.message}`);

                    return advice.title; // Fallback to original title

                });

            }

            // Translate content

            if (advice.content) {

                translatedContent = await translate(advice.content, { to: 'ar' }).catch(error => {

                    console.error(`Error translating content: ${error.message}`);

                    return advice.content; // Fallback to original content

                });

            }

            // Return translated advice object

            return {

                ...advice.toObject(),

                title: translatedTitle,

                content: translatedContent

            };

        }));

        res.status(200).json({ success: true, data: translatedAdvices });

    } catch (error) {

        res.status(500).json({ success: false, error: error.message });

    }

A screenshot of a phone

Description automatically generated};

// get appointments with filtration

const getAppointments = async(req, res) => {

    try {

        const { doctorName, address, fee } = req.body;

        let query = {};

        // Check if doctorName is provided

        if (doctorName) {

            // Find the doctor whose name matches the provided name

            const doctor = await Doctor.findOne({ firstName: doctorName });

            if (doctor) {

                // If doctor found, filter appointments by doctor ID

                query.doctor = doctor.\_id;

            } else {

                // If doctor not found, return empty result

                return res.status(200).json({ success: true, data: [] });

            }

        }

        // Check if address is provided

        if (address) {

            query.address = { $regex: address, $options: 'i' };

        }

        // Check if fee is provided

        if (fee) {

            // Parse fee value

            const feeValue = parseInt(fee);

            // Check if fee is a valid number

            if (!isNaN(feeValue)) {

                // Update query to find appointments with fee less than or equal to the specified value

                query.fee = { $lte: feeValue };

            } else {

                return res.status(400).json({ success: false, error: 'Invalid fee value' });

            }

        }

        // Find appointments matching the query and populate the doctor information

        const appointments = await Appointment.find(query).populate('doctor');

        res.status(200).json({ success: true, data: appointments });

    } catch (err) {

        res.status(500).json({ success: false, error: err.message });

    }

};

A screenshot of a computer

Description automatically generated

// Get Articles in Arabic

exports.getArticlesInArabic = async(req, res) => {

    try {

        const category = req.query.category;

        const articles = category ? await ArticleModel.find({ category }) : await ArticleModel.find();

        // Translate each article's title, content, and category to Arabic

        const translatedArticles = await Promise.all(articles.map(async(article) => {

            let translatedTitle = article.title;

            let translatedContent = article.content;

            let translatedCategory = article.category;

            // Check and translate title

            if (article.title) {

                translatedTitle = await translate(article.title, { to: 'ar' }).catch(error => {

                    console.error(`Error translating title: ${error.message}`);

                    return article.title; // Fallback to original title

                });

            }

            // Check and translate content

            if (article.content) {

                translatedContent = await translate(article.content, { to: 'ar' }).catch(error => {

                    console.error(`Error translating content: ${error.message}`);

                    return article.content; // Fallback to original content

                });

            }

            // Check and translate category

            if (article.category) {

                translatedCategory = await translate(article.category, { to: 'ar' }).catch(error => {

                    console.error(`Error translating category: ${error.message}`);

                    return article.category; // Fallback to original category

                });

            }

            // Return translated article object

            return {

                ...article.toObject(),

                title: translatedTitle,

                content: translatedContent,

                category: translatedCategory

            };

        }));

        res.status(200).json({ success: true, data: translatedArticles });

    } catch (error) {

        console.error(`Error fetching or translating articles: ${error.message}`);

        res.status(500).json({ success: false, error: error.message });

    }

};

// update doctor's online status

exports.updateDoctorStatus = asyncHandler(async (req, res, next) => {

    const doctorId = req.params.id; // Assuming id is passed as a route parameter

    // Check if doctorId is provided

    if (!doctorId) {

        return next(new ApiError('Doctor ID is required in the route parameters', 400));

    }

    const { online } = req.body; // Assuming onlineStatus is passed in the request body

    // Check if onlineStatus is provided

    if (online == undefined) {

        return next(new ApiError('Online status is required in the request body', 400));

    }

    try {

        // Update the doctor's online status based on doctorId

        const updatedDoctor = await Doctor.findByIdAndUpdate(doctorId, { online: online }, { new: true });

        // If no doctor found, return error

        if (!updatedDoctor) {

            return next(new ApiError(`No doctor found with ID ${doctorId}`, 404));

        }

        // If updated successfully, return success message

        res.status(200).json({

            success: true,

            data: "Doctor online status updated successfully",

            doctor: updatedDoctor // Optionally return the updated doctor data

        });

    } catch (err) {

        // If an error occurs, pass it to the error handler middleware

        return next(err);

    }

});

A screenshot of a phone

Description automatically generated

exports.getAllMonthlyMeasurements = asyncHandler(async (req, res, next) => {

    const { id } = req.params;

    const baby = await Baby.findById(id);

    if (!baby) {

        return next(new ApiError(`No baby found with id ${id}`, 404));

    }

    res.status(200).json({

        status: 'success',

        data: baby.measurements

    });

});

A screenshot of a phone

Description automatically generated

# Chapter 6: Conclusions and Future Work

6.1 Conclusions

According to the features and functionalities designed and tested, and by addressing the initial problem statement, Mommy Mentor Application has successfully provided:

✓ **Growth Tracking:** A user-friendly tool for recording and monitoring baby's weight and height, allowing mothers to easily observe growth patterns and identify any deviations.

✓ **Vaccination Reminders:** Display the upcoming vaccine appointments, including information on where to go for vaccinations and the vaccines needed according to the baby's age, ensuring that critical immunizations are not missed.

✓ **Educational Resources:** A rich library of articles and videos that educates and empowers mothers with knowledge on various aspects of baby care, including general nutrition, breastfeeding, diseases, and care for abnormal babies.

✓ **Doctor Search:** Mothers can search for a doctor by name, area, and price, ensuring they find the best care tailored to their needs and preferences.

✓ **Age-Based Advice:** The app provides tailored advice for mothers based on their child's age, ensuring relevant and timely guidance.

✓ **Advanced machine-learning models that:**

Detect jaundice in babies from baby’s photo, enabling early detection and intervention.

Identify potential signs of Down syndrome, facilitating early awareness and consultation.

Recommend actions based on the symptoms entered by mothers, providing timely advice on baby care.

In conclusion, Mommy Mentor Application has passed the validation tests and meets the requirements expected from it, addressing the initial problem statement effectively. Mommy Mentor stands as a comprehensive solution to the challenges faced by new mothers. By integrating user-friendly growth tracking, reliable vaccination reminders, a wealth of educational resources, an efficient doctor search feature, and age-based advice, the app ensures that mothers are well-equipped with the necessary tools and knowledge for effective baby care. Moreover, the incorporation of advanced machine learning models for detecting jaundice and Down syndrome, as well as providing symptom-based recommendations, highlights the app’s commitment to leveraging technology for early detection and timely intervention.

6.2 Recommendation and Future Work

To further enhance the functionality and value of our mobile application, the following features are proposed for future development:

**Sleep Tracker:**

Implement a feature where mothers can log their baby is sleeping times. The app will analyze the data and provide feedback on whether the sleep patterns are normal; helping mothers ensure their baby gets adequate rest.

**Feeding Tracker:**

Introduce a feeding tracker that allows mothers to record the times of each meal and get reminders for the next meal. The app will assess if the feeding schedule is normal and provide recommendations if adjustments are needed. Additionally, it will offer guidance on what the baby can eat at different stages of development.

**Movement Analysis:**

Develop an advanced feature that utilizes camera footage to analyze the baby's movements. This tool will help detect any abnormalities or developmental issues, allowing for early intervention and support.

**Personalized Recommendations:**

Build a recommendation system to suggest relevant articles, videos, and resources based on the mother's preferences and past interactions. This will help mothers discover new information tailored to their needs.

**Health Monitoring Integration:**

Integrate with wearable devices to track the baby's vital signs such as heart rate and sleep quality. This data can provide deeper insights into the baby's health and well-being.

**Emergency Alerts and First Aid:**

Develop an emergency alert system that provides quick access to first aid information and emergency contact numbers. This ensures mothers can respond promptly to any urgent health issues.

**Multi-language Support:**

Expand the application’s accessibility by offering multi-language support, ensuring that users who do not understand English can benefit from the app’s features.

**Mobile Wallet development:**

Develop a mobile wallet feature that allows users to pay for medical consultations, book appointments with doctors, and manage their transactions directly within the app, increasing convenience and engagement.

**Community Support Forums:**

Create a platform within the app for mothers to connect, share experiences, ask questions, and offer support to one another. This community feature will foster a sense of belonging and provide an additional layer of support for new mothers.

These future enhancements will build on the existing foundation, making the app an even more comprehensive and indispensable tool for parenting. By continuously evolving and expanding its features, the app will remain a valuable resource for mothers, helping them provide the best care for their babies.