



**Minia University
Faculty of Computers and Information**



إبرة أمان

For Faculty of Computers and information

[Graduation project [1,2]]

Submitted in partial fulfilment of requirements

For the award of

Bachelor's degree

Of the

Minia University- Faculty of Computers and Information

Supervisor

Dr: Ebtsam Abdel hakam

Eng: Ahmed Khairy



إبْرَةُ أَمَانٍ

Presented to Faculty of computers and information,
Minia University

In Partial Fulfilment
Of the Requirements for the Degree of
Bachelor of in Computers and Information

By:

Gamal Hussien Elsayed

gmalhhussein@gmail.com

Saeid Khaled Salah

saeidkhaled13@gmail.com

Eslam Ahmed Abo Ghanima

eslam.ahmed3045@gmail.com

Sohyla Gomaa Abd Elmoatamed

sohylaelqourashi@gmail.com

Doaa Ashraf Hassan

adoaa6322@gmail.com

Egypt, Minia

Fall 2025

Table of Contents

1 Introduction.....	7
1.1 Abstract.....	8
1.2 Introduction.....	9
1.3 Background.....	9
1.4 Problem statement.....	10
1.5 Required tools.....	11
2 Requirement analysis:.....	12
2.1 Work progress.....	13
3 Fesability Study.....	14
3.1 Technical Feasibility.....	14
3.2 Organizational Feasibility	15
4 Business Requirements identification:.....	16
4.1 Function requirements.....	17
4.2 NonFunctional requirements.....	21
5 System Design.....	23
5.1 Use case	24
5.2 Use case Format	27
5.3 Use case Digram.....	35
5.4 Sequance Digram.....	36
5.5 Class Digram.....	50
5.6 Activity Diagram.....	51
5.7 User interface design.....	62
5.8 Site Map.....	63
5.9 Story board.....	64
6 Conclusion.....	81
5.1 Conclusion:.....	81
5.2 Difficulty faced:.....	81
5.3 Future Works:.....	82
5.4 Reference.....	82

List of figures

Work progress	
Gant chart fiugre.....	13
Full System use case.....	
Use case figure.....	35
Sequance Diagram	
Figure 5.3.1.....	36
Figure 5.3.2.....	37
Figure 5.3.3.....	38
Figure 5.3.4.....	39
Figure 5.3.5.....	40
Figure 5.3.6.....	41
Figure 5.3.7.....	42
Figure 5.3.8.....	43
Figure 5.3.9.....	44
Figure 5.3.10.....	44
Figure 5.3.11.....	45
Figure 5.3.12.....	46
Figure 5.3.13.....	47
Figure 5.3.14.....	48
Figure 5.3.15.....	49
Class Diagram	
Class diagram figure	50
Activity Diagram.....	
Figure 5.7.1.....	51
Figure 5.7.2.....	52
Figure 5.7.3.....	53
Figure 5.7.4.....	54
Figure 5.7.5.....	55

Figure 5.7.6.....	56
Figure 5.7.7.....	57
Figure 5.7.8.....	58
Figure 5.7.9.....	59
Figure 5.7.10.....	60
Figure 5.7.11.....	61
User interface.....	
Site map figure.....	63
Story board.....	
Figure 5.9.1.....	64
Figure 5.9.2.....	65
Figure 5.9.3.....	66
Figure 5.9.4.....	67
Figure 5.9.5.....	68
Figure 5.9.6.....	69
Figure 5.9.7.....	70
Figure 5.9.8.....	71
Figure 5.9.9.....	72
Figure 5.9.10.....	73
Figure 5.9.11.....	74
Figure 5.9.12.....	75
Figure 5.9.13.....	76
Figure 5.9.14.....	77
Figure 5.9.15.....	78
Figure 5.9.16.....	79
Figure 5.9.17.....	80
Figure 5.9.18.....	81
Figure 5.9.19.....	80
Figure 5.9.20.....	83
Figure 5.9.21.....	84

Figure 5.9.22.....	85
Figure 5.9.23.....	86
Figure 5.9.24.....	87
Figure 5.9.25.....	88
Figure 5.9.26.....	89
Figure 5.9.27.....	90
Figure 5.9.28.....	91
Figure 5.9.29.....	92
Figure 5.9.30.....	93
Figure 5.9.31.....	94
Figure 5.9.32.....	95
Figure 5.9.33.....	96
Implementation.....	
Figure 6.....	97
Figure 6.1.....	98
Figure 6.2.....	99
Figure 6.3.....	100
Figure 6.4.....	100
Figure 6.5.....	101
Figure 6.6.....	102
Figure 6.7.....	103
Figure 6.8.....	104
Figure 6.9.....	105
Figure 6.10.....	106
Figure 6.11.....	107
Figure 6.12.....	108
Figure 6.13.....	109
Figure 6.14.....	110
Figure 6.15.....	111
Figure 6.16.....	112

List of Table

Use case Tables.....	
Parent use case table	24
Doctor use case table	25
Healthy care use case table	26
Ministry of Health admin use case table.....	27
Use case formates.....	
Register use case format.....	28
Login use case format.....	29
Child profile creation use case format.....	30
Vaccination schdule Display use case format.....	31
Book vaccine appointment use case format.....	32
Find best place use case format.....	33
Give feedback use case format.....	34

Introduction

1.1 Abstract

This project focuses on developing a web-based Child Vaccination System designed to streamline and enhance the vaccination process for children. The system enables parents to manage their children's health records, book vaccination appointments, and track vaccination status.

Doctor can access the platform to manage vaccination schedules, update children's health records, and address vaccine inventory issues. A built-in complaint management system allows parents to report side effects and receive timely responses. The system's user-friendly interface, real-time data management.

This innovative platform bridges the gap between parents and healthcare providers, improving accessibility, efficiency, and transparency in vaccination processes.

1.2 ACKNOWLEDGEMENT

We are very thankful to everyone who supported us complete this paper effectively. We would like to thank our advisor [Dr: Ebtsam Abdul Hakam](#) and [Eng: Ahmed Khairy](#) for her instruction and helping us to finish this project documentation successfully.

1.3 Introduction

Our graduation project is a comprehensive Child Vaccination Management System, designed to simplify and enhance the vaccination process for children. This system will serve as a centralized platform for parents, healthcare providers, and government authorities to ensure timely vaccinations and maintain accurate health records. By leveraging advanced technology and integration with external services, such as Google Maps and payment gateways, the system will deliver user-friendly features while adhering to healthcare standards and regulations.

1.4 Background

A child vaccinations system is a software application designed to help parents and healthcare providers track, manage, and schedule vaccinations for children. It serves as a centralized platform for storing vaccination records, accessing information about recommended vaccines.

Vaccination is a critical component of public health, aimed at preventing the spread of infectious diseases. However, managing vaccinations can be challenging for parents and healthcare providers due to inconsistent schedules, lack of reminders, and unavailability of reliable health records. Additionally, healthcare systems often face logistical issues like tracking vaccine inventory and distribution, as well as handling vaccination-related complaints.

In response to these challenges, our project focuses on digitalizing the vaccination management process, making it more accessible, secure, and efficient. By integrating functionalities such as child profile management, automated reminders, and vaccine inventory tracking, the system will bridge gaps in current vaccination processes and empower users with reliable tools and resources.

1.5 Problem statement

- Missed Vaccination Dates: Parents often fail to remember their child's vaccination schedules, leading to delays or missed doses.
- difficulty to find the vaccine recommend the best place for getting vaccine.
- Lack of Notifications: Parents are not consistently notified about upcoming vaccinations or their child's health status.
- Raising awareness of the importance of vaccinations lack of resources that help parents to know the importance of vaccine.
- difficult reaching the remote area to receive the vaccine using API google map for reaching the best place to get vaccine (+ check availability of vaccine).
- Inventory Management Issues: Healthcare facilities face challenges in managing and tracking vaccine stock levels, leading to shortages or delays.
- difficulty in contact with the doctor responsible for the vaccine (chat with doctor based on timetable description for side effects for parents after taking vaccine).
- Emergency preparedness and response to the spread of diseases. (reports -dashboard for display statistics).
- Data Integration Problems: Current systems lack integration with government databases and other healthcare platforms, making it difficult to share vaccination data seamlessly.
- Recording child's previous vaccination schedule customize for some pages based on child age.

- **1.6 Required tools:**

1.1 figma:

is a cloud-based design and prototyping tool used for creating user interfaces (UI) and user experiences (UX). It allows designers and developers to collaborate in real-time on web and mobile app designs. With features like vector editing, prototyping, design systems, and team libraries, Figma is widely used for building interactive and responsive UI mockups directly in the browser—no need for downloads.

1.2 Microsoft SQL Server:

It is a Relational Database Management System (RDMS) that uses the Structured Query Language SQL to access and manipulate stored data developed by Microsoft. It is a software product whose primary function is to store and retrieve data as requested by other software applications. Its scope includes data query and data update, schema creation and modification, and data access control. In SQL Server, the RDMS is integrated within the Visual Studio IDE to simplify data access from Windows applications. (About Microsoft SQL, 2012)

1.3 Microsoft Visual Studio:

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native codes together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework, and Microsoft Silverlight. (What is Visual Studio? 2012)

1.4 Microsoft Visual Studio Code:

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. We used it to develop frontend pages using language like typescript and angular framework

1.5 Entity Framework Core:

C# ORM Layer that help in contact with database making relational database using classes and relations

1.6 Hangfire:

C# Library used for handling background jobs to increasing performance like making send notification, emails and heavy jobs to be executed in background and not in sequential way

1.7 SignalR:

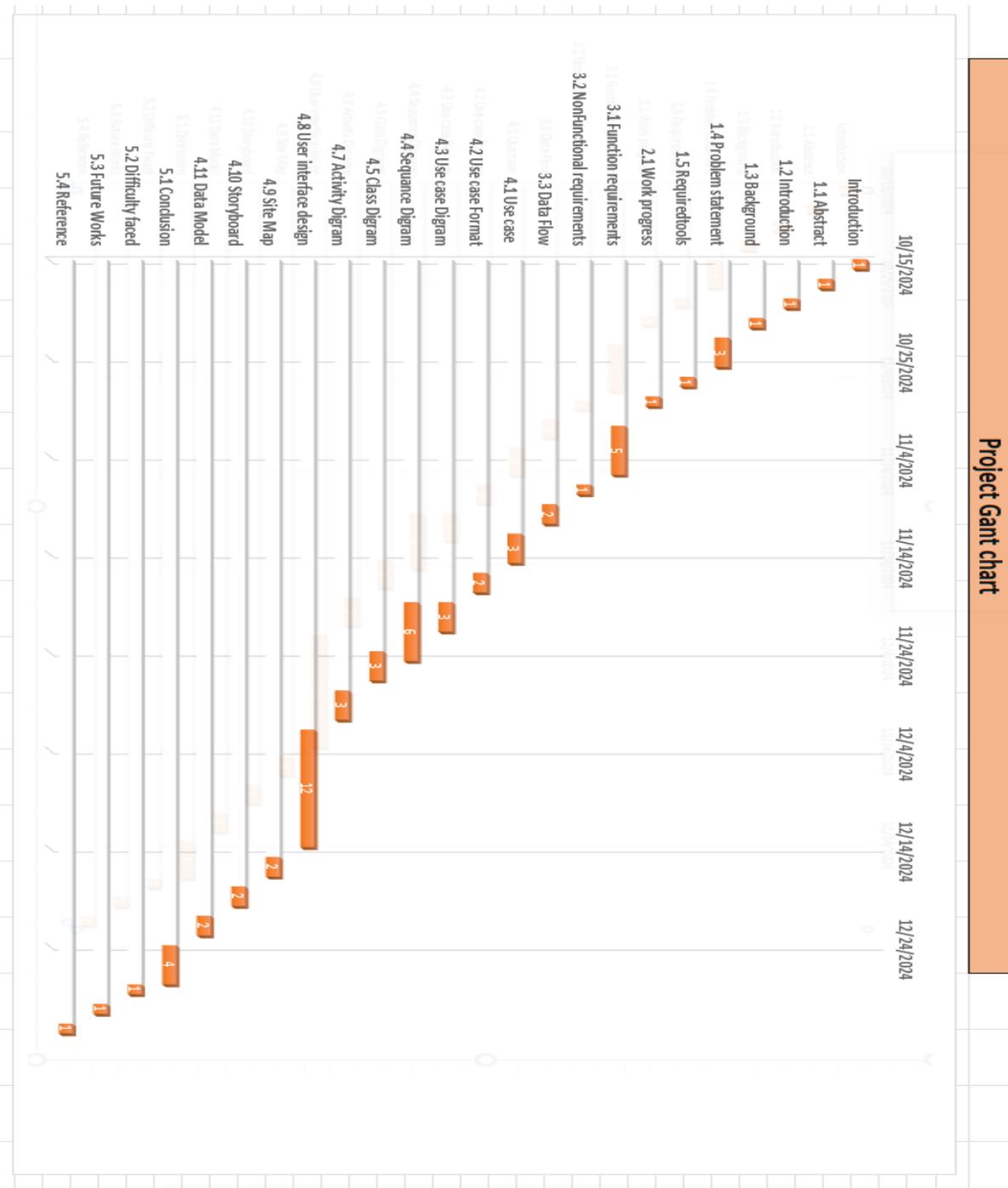
WebSocket way for handling real time applications and notifications helping interact with user in real time way

1.8 Postman:

An application for testing backend endpoint with Http methods and real time websocket application

Requirements analysis

Work progress:



Gant chart figure 2.1

FEASIBILITY STUDY:

Before proceeding with the project our team did a feasibility study that helped us make informed decisions about whether to proceed. It provided a clear understanding of potential challenges, risks, and opportunities associated with developing and implementing the child vaccination system. The findings for feasibility study guided us in determining that the project is viable and should move forward the highlights of

The feasibility study analysis is

❖ Technical feasibility:

- Familiarity with applications (Medium)

After doing some comprehensive analysis we find that our users can be with different levels of technology use so, we provide a user-friendly interface provide a seamless user experience for all users of the system, also this analysis helped all team members to understand all processes that will be performed on the system

- Familiarity with technology (High)

We have good experience in:

- 1) Asp.net core for backend
- 2) Angular for frontend
- 3) Figma and Ado premier for Design

- Project Size (Medium)

We have 5 teamwork members, who can work on the system and have 8 months to finish the project

Based on the technical feasibility findings, we can proceed with the implementation of the chosen technologies The compatibility and synergy between asp.net core and angular well-thought-out and cohesive technological stack that aligns effectively with the

project's objective. The combination of these technologies is likely to result in a secure, modular, and user-friendly child vaccination system that meets both technical and functional requirements. Continuous monitoring and adaptation to emerging technologies should be considered throughout the development process to ensure the system remains cutting-edge and aligned with industry standards.

❖ **Organizational Feasibility**

Project Sponsor

- A. Role: Oversee project alignment with organizational goals and secure necessary resources.
- B. Responsibilities: Provide strategic guidance, allocate resources, and resource provision.
- C. Commitment: Fully committed to child vaccination System, and actively involved in meetings, decision-making and resource provision.

Project Management

- A. Methodology: Agile
- B. Communication: Regular meeting
- C. Quality assurance: Ensure high-quality deliverables
- D. Resource Management: Efficient allocation and utilization
- E. Updates: Regular updates to stakeholder

Business Requirements identification

Functional Requirements:

- **User registration:** Parents should be able to register for an account, user enters name, E-mail, phone number, National ID, country, city, address.
- **User login:** Secure login authentication for registered users, the system sends an email or SMS that contains a verification code.
- **Profile management:** Users should be able to update their profiles, the system allows users to do the updates that they want. such as if the user changes his address or his phone number and wants to update them.
- **Child profile creation:** Parents can create profiles for their children, parents enter the child's name, age, gender, national ID, Healthy status.
- **Update child profile:** Parents can update their child's Data.
- **Vaccination schedule display:** Display recommended vaccination schedules based on child's age and the healthy status the system displays to the user the time of the vaccines that suitable for the child based on his child profile.
- **Payment:** Parents should be able to pay for system services
- **User payment History:** The system should render a record of past payments in the parent's account for each child added in system.

- **Appointment booking:** Allow parents to book vaccine that system recommend to child based on this next vaccine in child vaccination table, track number of coming children for the hospital to track crowding then inform admin hospital.
- **Appointment cancellation:** Provide the option to cancel appointments providing a user interface to cancel appointments.
- **Appointment rescheduling:** Allow users to reschedule appointments if needed to rescheduling appointment providing user interface, reschedule options, select new date, check availability, confirmation prompt.
- **Find the best place:** System provides some of the nearest places to the user based on google map API or user chooses nearest one using drop down list provided in system.
- **Edit status of vaccine:** enable the health care provider to edit status of the vaccination for child (taken or not)
- **Tracking child Healthy:** Healthcare providers should be able to access the child's health record.
- **Immunization record tracking:** Keep track of each child's vaccination history design some cards for child previous vaccination that includes the date, time, the vaccine name, number of doses taken, location of Healthy unit.
- **Reminder notifications:** Send automated reminders for upcoming vaccination send SMS or email message contains the next vaccine for child, and the places where the vaccine is available.

- **Side Effects instruction:** Provide side effect of vaccine and helpful instructions for dealing with the side effects of the vaccine.
- **Report any ambiguous side effects:** Parents should be able to report any unexpected side effects after a vaccination.
- **Healthcare Provider Integration:** Enable integration between the system and healthcare provider. The system allows healthcare providers to receive, and review submitted data.
- **Vaccine Certification Printing:** Allows parents to print vaccination certificates for various purposes request for a certificate (e.g., for school enrolment, travel, or other official purposes), printable certificate containing vaccination details (child's name, vaccine types, dates administered).
- **Vaccine Inventory Management:** Manage and track vaccine inventory levels the system provides tools to monitor vaccine stock and update inventory records.
- **Vaccine distribution tracking:** Track the distribution of vaccines to different healthcare facilities, ensure that all healthcare facilities get the desirable amount of vaccine and send a report to admin site that they get the vaccine.
- **Report about low amount:** every healthy care can send report to the ministry of Health about shortage of vaccine and request amount.
- **Sync vaccine with database:** The system should be capable of syncing vaccination data with government databases when needed.

- **Vaccine education resources:** Provide educational materials on the importance of vaccination, develop a section within the application dedicated to providing educational materials on the importance of vaccination these materials can be videos or articles.
- **Update vaccination Schedule:** The system should include an updated vaccination schedule, modify the vaccination schedule as per new health guidelines
- **Statistics dashboard:** providing a Statistics section for admins to track the changes made by the admin under him.
- **Vaccine complaint:** providing a section help the user make complaint on system, set up a mechanism for users to track the status of their complaints and receive updates on their resolution, ensure that complaints are handled promptly and transparently, with clear communication to the user throughout the process.

Non-Functional Requirements:

Security

Providing a Log in system that detect who user is trying to access the data.

Ensure that all personal health information and sensitive data are encrypted during transmission and storage to comply with healthcare privacy regulations.

Performance

The system should be responsive and able to handle multiple user requests simultaneously without significant latency, especially during peak vaccination periods.

The system should be flexible in receiving requests and sending responses with multiple user requests.

Availability

refers to ensuring that the system is consistently accessible and operational for users whenever they need to access vaccination-related services.

Scalability

The system should be designed to accommodate an increasing number of users, healthcare providers, and vaccination data without compromising performance.

make the system available for adults not for children only in some cases such as virus C, B, and corona virus.

Reliability

Ensure high availability and minimal downtime of the system to prevent disruptions in vaccination scheduling and administration.

The system should accurately record, and track vaccination doses administered to patients with minimal risk of data loss or corruption.

Interoperability

The system should be able to integrate with existing healthcare information systems and standards to exchange data seamlessly between different platforms and providers.

Providing an API for integrating with payments and google maps for choosing the nearest location for vaccine.

Usability

Design an intuitive user interface that is easy to navigate, especially for parents with limited technical proficiency, and ensure accessibility for users with disabilities.

Maintainability

The system will be written in a way that will be easy to understand, with separate tasks for each function and available to enhance if we want to provide more features in future each function in the system should be independent in each other and if I changed anything in the system don't affect in the whole system.

Recovery

Implement regular data backups and establish procedures for data recovery in case of system failures or data loss.

System Design

Actor: Parent	System
User Registration: Register and create an account.	Display screen with input fiddles for register
User Login: access the parent count and have ability to interact with system	Display screen for parent for easy and secure login with various features
Profile Management: Update personal details like address or phone number.	Fetch the current profile details validate and save the updated information to the database.
Child Profile Management: Add, update, or delete a child's profile.	Screen for manage child record or update
Vaccination Schedule Display: View recommended schedules based on child's profile.	Display the schedule to the parent.
Appointment Booking: Book vaccination appointments.	show screen for reserve vaccine and show the available date for receiving.
Appointment Management: Cancel or reschedule appointments.	give options for cancel or chose another day to get vaccine.
Pay for services for each child added	Stripe API interface for manage payment with card, process the payment securely, save transaction details in the user's payment history.
Find Best Place: Select a preferred centre.	Use Google Maps API to fetch nearby locations, display the locations and save the user's choice.
Report side effects: Report any side effects observed in their child.	Collect the side effect report and store it in the database
Request certificate: Request a vaccination certificate.	Generate a printable vaccination certificate.
Submit Complaint: complaint about the system or services.	Collect and log the complaint Send feedback via email to ministry of health

Actor: Doctor	System
Add Schedule: add schedule for children who has a disease	user interface page for adding some unregular vaccine for those children.
Access a child's profile: enable to manage the child record based on child healthy	Fetch and display the child's health and vaccination history Fetch and display the child's health and vaccination history and doctor can update this profile
Handle Complaints: tracking the compliment of parent about the side effects of vaccine via chat page	provide chat page between parent and doctor to track child healthy

Actor: organizer	System
Edit Vaccine Status: Update the vaccination status of a child.	show interface for doctor via it he can update if child take this vaccine or not, save the updated vaccination status in the database.
Report Inventory Issues: report if any lack of vaccine amount.	Send the request to the Ministry of Health for needing or lack of some vaccine.

Actor: Admin of Ministry of health	System
Vaccine Inventory Management: Track and update vaccine stock levels.	Show a page for every health centre and amount of vaccine available in each one.
Vaccine Distribution Tracking: Monitor vaccine deliveries to healthcare centres	Provide page for health centre and show if received the amount of vaccine or not.
Handle Complaints: Address user complaints and provide resolutions.	Log and track complaints, provide updates to users about their complaints

Actor: governorate admin	System
Statistics dashboard: view statistic for the governorate he manages	Render the statistics for admin user to track the order and requested vaccines and the most requested city for vaccine
Vaccine Distribution Tracking: Monitor vaccine deliveries to cities	Provide page for governorate admin and show if received the amount of vaccine or not.
Vaccine order management: view all orders he requested to the admin of ministry of health	Provide a page for orders that enable accept order and track requested order
Adding city admins: page for adding city admins to manage the cities under this governorate	Provide page for manage and view all cities and manage city admins for this city

Actor: city admin	System
Statistics dashboard: view statistic for the governorate he manages	Render the statistics for admin user to track the order and requested vaccines and the most requested health care for vaccine
Vaccine Distribution Tracking: Monitor vaccine deliveries to healthcare centres	provide page for city admin and show if received the amount of vaccine or not.
Vaccine order management: view all orders he requested to the governorate admin	Provide a page for orders that enable accept order and track requested order
Adding doctors and organizers: page for adding doctor or organizer to specific healthcare	Page for view and manage all doctors in this city and in which health care
Handle complaint: page for complaint coming from the health care via specific user	Provide page for render all complaints from user and can contact with user to notify him that problem solved

Use case format

Use Case Name	Register
Primary actor	Parent
Description	This use case allows user to make new account in the system
Precondition	1. Connecting to the network 2. Open the website 3. Have an E-mail account
Basic Path	1. The user clicks on the “Register” button 2. The system displays the registration form 3. The user fill the required fields 4. The user clicks on the “Submit” button
Post condition	The system sends a confirmation email to the user's email address and displays the log the page.
Alternative Paths	-If the user enters invalid information, the system displays an error message. -If the user already has an account, the system displays a message asking the user to login

Use Case Name	Login
Primary actor	Doctor, Parent, Admin of Ministry of health, city admin, governorate admin, organizer
Description	Login to the system
Precondition	The user has a valid account
Basic Path	<ol style="list-style-type: none"> 1. The user clicks on the “Login” button 2. The system displays the login form 3. The user enters their email address and password 4. The user clicks on the “Login” button
Post condition	The system logs the user in and displays the home page of logged Actor.
Alternative Paths	If the user enters invalid information, the system displays an error message.

Use Case Name	Profile Management.
Primary actor	Parent
Description	Update User Profile
Precondition	<ol style="list-style-type: none"> 1. The user is authenticated and logged into the system. 2. The user's account exists in the database.
Basic Path	<ol style="list-style-type: none"> 1. The user navigates to the "Edit Profile" page. 2. The system retrieves the current profile information and displays it. 3. The user makes changes to the desired fields (e.g., phone number, address). 4. The user clicks the "Save" button. 5. The system validates the input data (e.g., format, completeness). 6. The system updates the profile information in the database. 7. A success message is shown to the user.
Post condition	The system updates user profile and displays the home page.
Alternative Paths	<p>Invalid Data:</p> <ol style="list-style-type: none"> 1. The user enters invalid data (e.g., an incorrect phone number format). 2. The system displays an error message and highlights the problematic fields. 3. The user corrects the input and resubmits. <p>Database Error:</p> <ol style="list-style-type: none"> 1. The system encounters an issue saving the data. 2. The system displays an error message and advises the user to retry or contact support.

Use Case Name	Child Profile Creation.
Primary actor	Parent
Description	Parent profile for his child
Precondition	The parent is authenticated and logged into the system.
Basic Path	<ol style="list-style-type: none"> 1. The user clicks on "add Child" button. 2. The system displays a form with fields for Child's Name, Age, Gender, National ID, weight, and Health Status. 3. The parent fills in the form with the child's details. 4. The parent submits the form by clicking the "add" button. 5. The system validates the entered information. 6. Ensures all required fields are completed. 7. Verifies the format of the National ID and Age. 8. The system saves the child profile in the database. 9. The system displays a success message and links the new profile to the parent's account.
Post condition	<ol style="list-style-type: none"> 1. The parent receives a confirmation message
Alternative Paths	<p>Invalid Data:</p> <ol style="list-style-type: none"> 1. The user enters incomplete or incorrect information (e.g., invalid National ID). 2. The system highlights the errors and displays an appropriate error message. 3. The user corrects the information and resubmits. <p>Database Error:</p> <ol style="list-style-type: none"> 1. The system encounters an issue while saving the profile. 2. The system displays an error message and suggests the parent retry later or contact support.

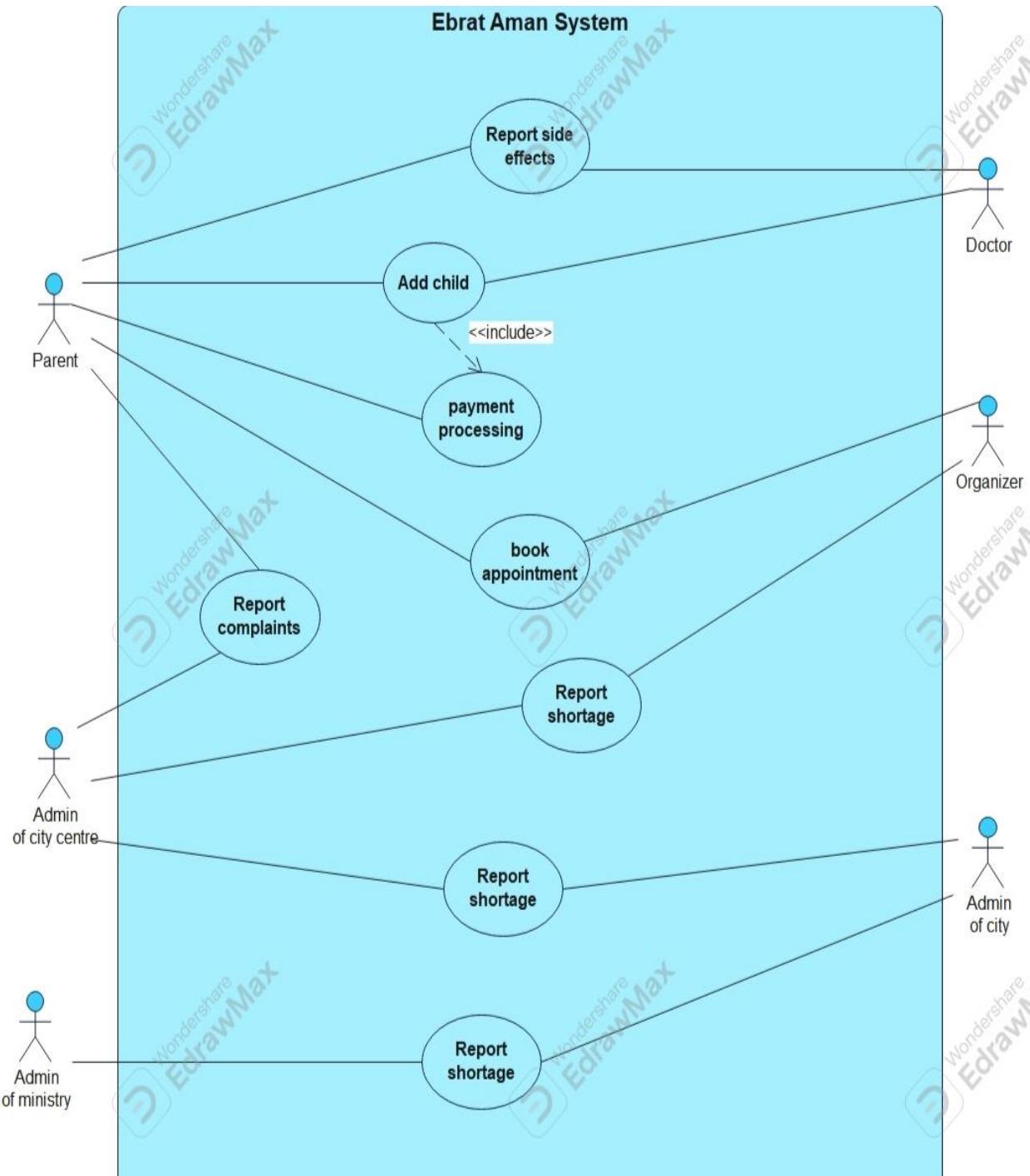
Use Case Name	Vaccination Schedule Display
Primary actor	Parent
Description	the system displays vaccination schedule to parent of child.
Precondition	<ol style="list-style-type: none"> 1. The parent is authenticated and logged into the system. 2. The child profile exists with accurate age, health status, and other necessary details. 3. The schedule is based on the child's profile data and is up to date with the latest medical guidelines
Basic Path	<ol style="list-style-type: none"> 1. The parent click view button to display the schedule 2. The system displays vaccination schedule for added child in a user-friendly format.
Post condition	1. The child's vaccination schedule is displayed to the parent.
Alternative Paths	<p>Incomplete Profile Data:</p> <ol style="list-style-type: none"> 1. The parent updates the profile of child the system displays new vaccination schedule based on new data. <p>No Vaccinations Needed:</p> <p>If the child's age or health status does not require any vaccinations, the system displays a message indicating no current vaccinations are recommended.</p>

Use Case Name	Book Vaccination Appointment
Primary actor	Parent
Description	parent wants to book a vaccination appointment for their child at a convenient time.
Precondition	<ol style="list-style-type: none"> 1. The parent is authenticated and logged into the system. 2. The child's vaccination schedule exists and includes the next recommended vaccine. 3. The hospital has availability for appointments.
Basic Path	<ol style="list-style-type: none"> 1. The parent navigates to the child's vaccination schedule and selects the next recommended vaccine. 2. The system retrieves available appointment slots from the hospital database. 3. The system displays available dates and times for the selected vaccine. 4. The parent selects a preferred date and time. 5. The system confirms availability and updates the hospital's appointment log. 6. The system sends a booking confirmation to the parent, including: <ol style="list-style-type: none"> 1. Appointment date and time 2. Hospital details additional instructions (e.g., required documents). 9. The system updates the hospital's record with the total number of children expected for the selected date to monitor crowding. 10. The system notifies the hospital administrator of any crowding risks (e.g., if bookings exceed capacity).
Post condition	<ol style="list-style-type: none"> 1. The vaccination appointment is successfully booked. 2. The parent receives confirmation and appointment details.
Alternative Paths	<p>Parent Cancels Booking:</p> <ol style="list-style-type: none"> 1. The parent decides to cancel the booking after confirmation. 2. The system updates the hospital's appointment log to free the slot

	<p>and reduce the count.</p> <p>System Error:</p> <p>The system encounters an issue while saving the appointment and system display error message to user.</p>
--	---

Use Case Name	Vaccine complaint
Primary actor	Parent, Admin of city
Description	This use case allows user to make complaint on system.
Precondition	The user is authenticated and logged into the system.
Basic Path	<ol style="list-style-type: none"> 1. The user clicks the Report a complaint from dashboard. 2. Users write his complaint. 3. Admin of city track the complaint and alert the complainant to solve the problem
Post condition	The system will display complaint to Admin of city
Alternative Paths	User can add more than one complaint

Use case Diagram



Use case figure 2

Sequence Diagram

User Registration Sequence Diagram

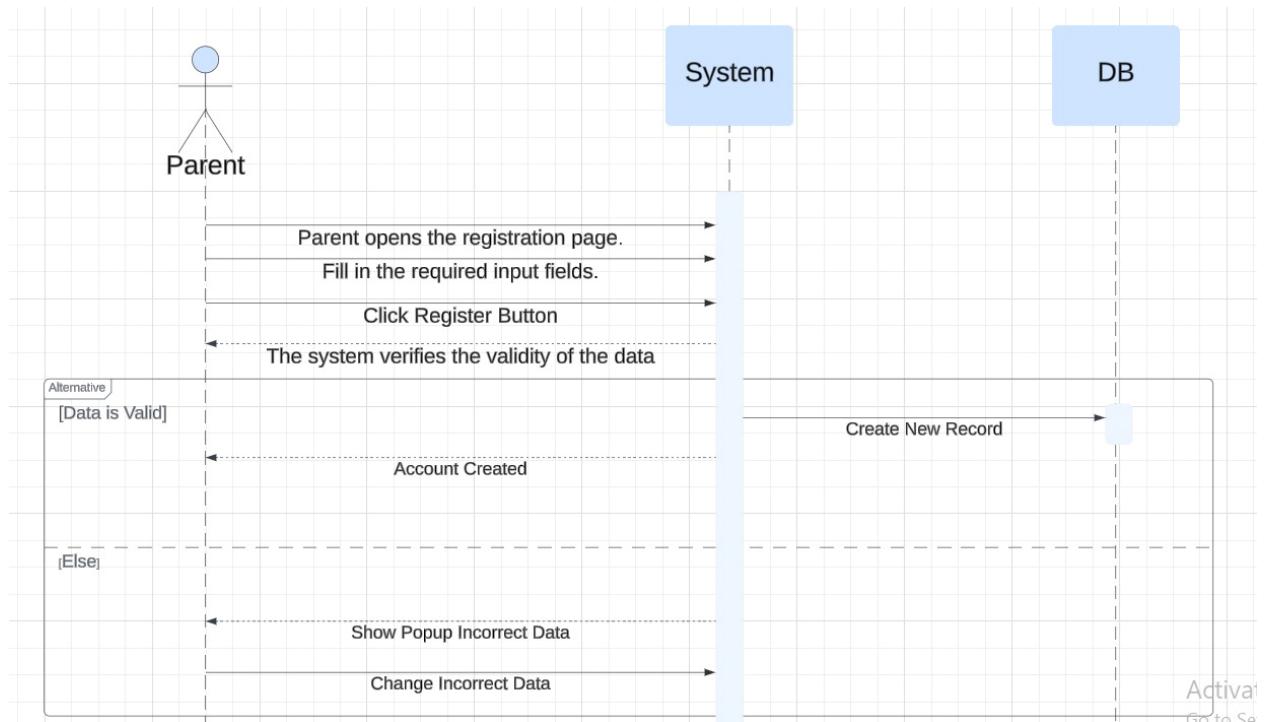


Figure 5.3.1

User login Sequence Diagram

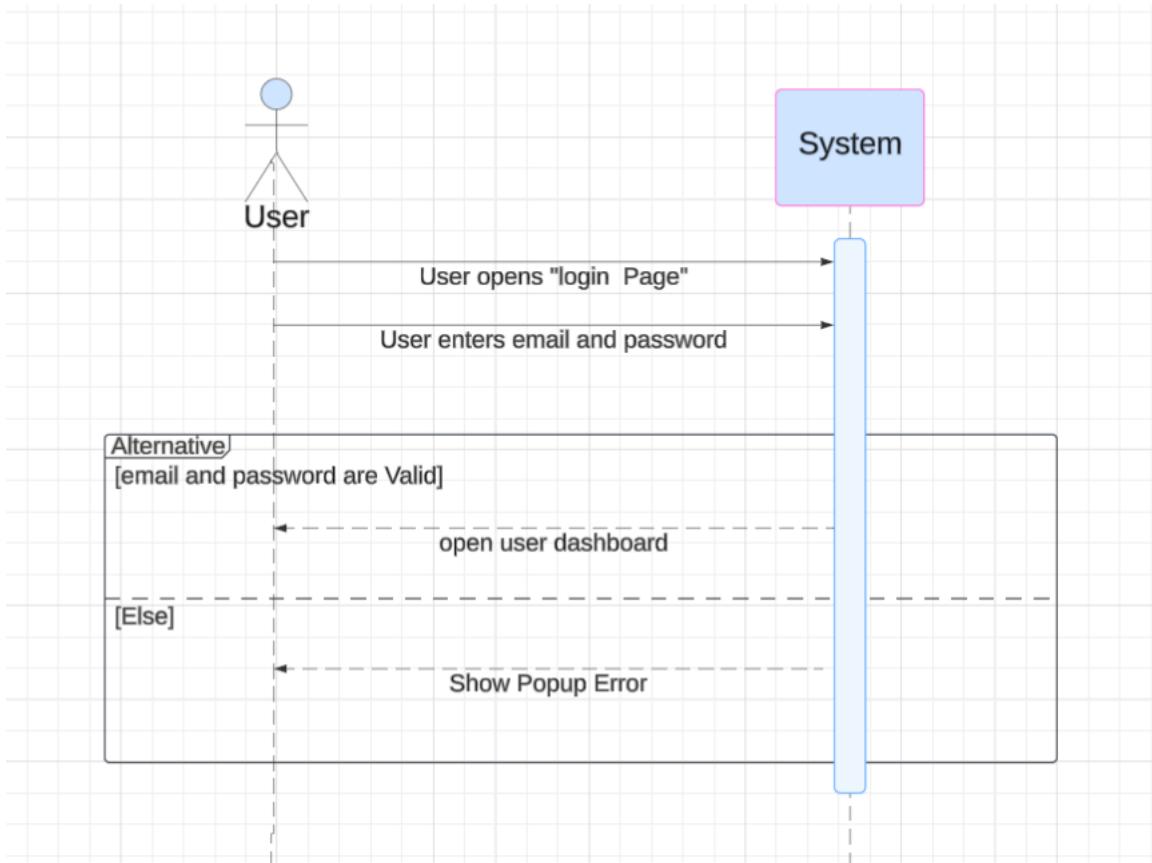


Figure 5.3.2

Profile Management Sequence

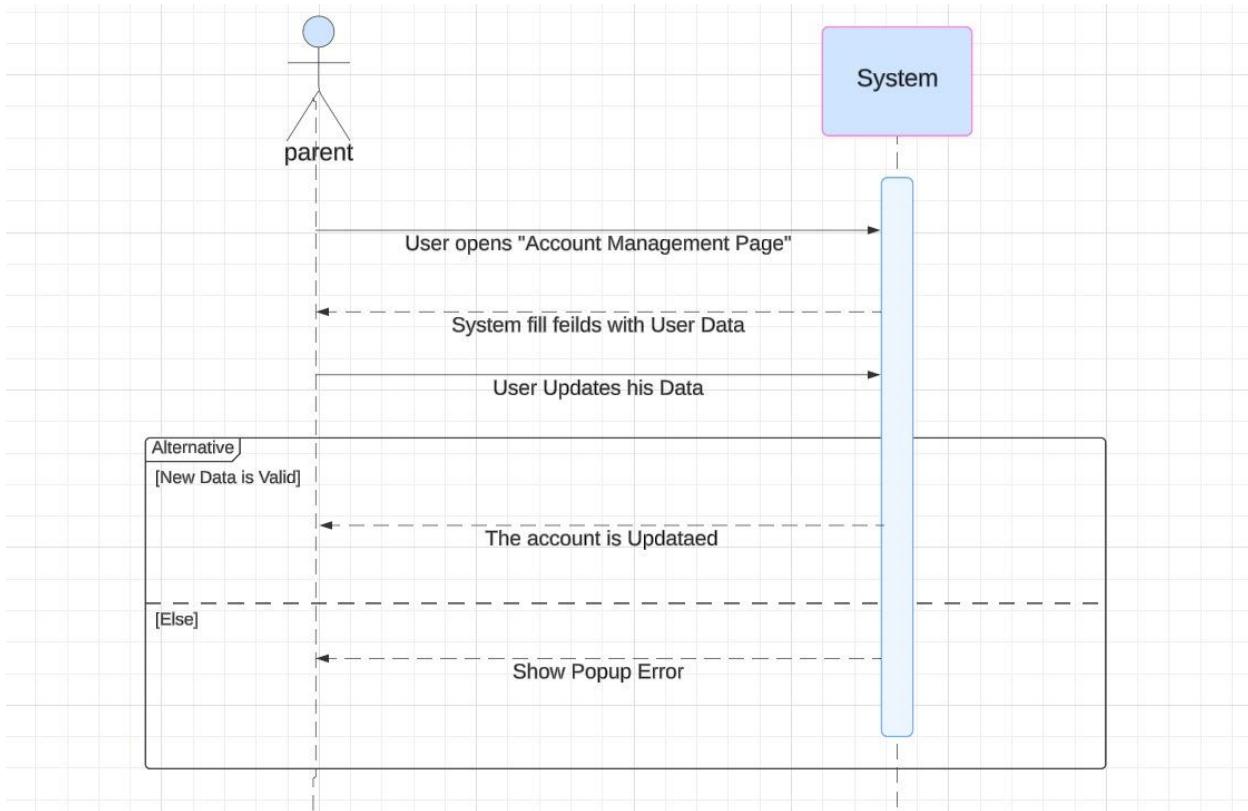


Figure 5.3.3

Add Child Profile Management Sequence Diagram

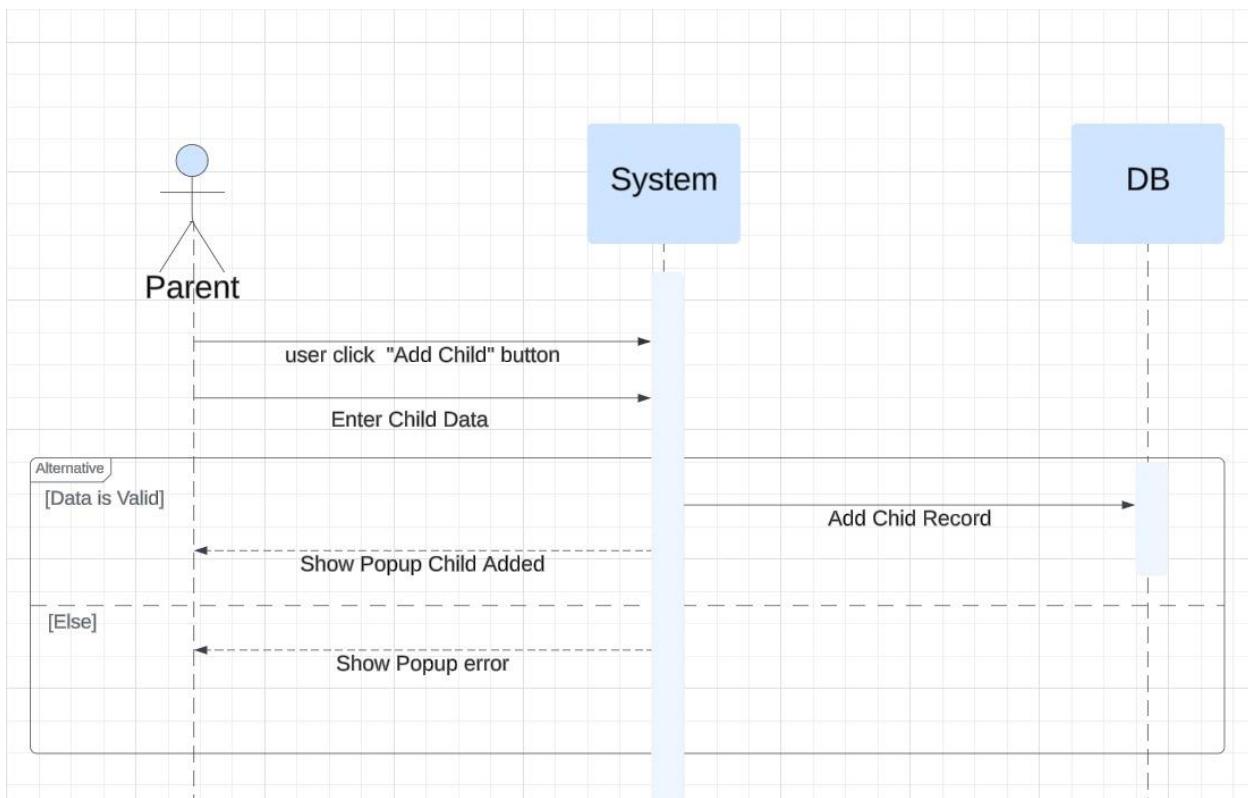


Figure 5.3.4

Vaccination Schedule Display Sequence Diagram

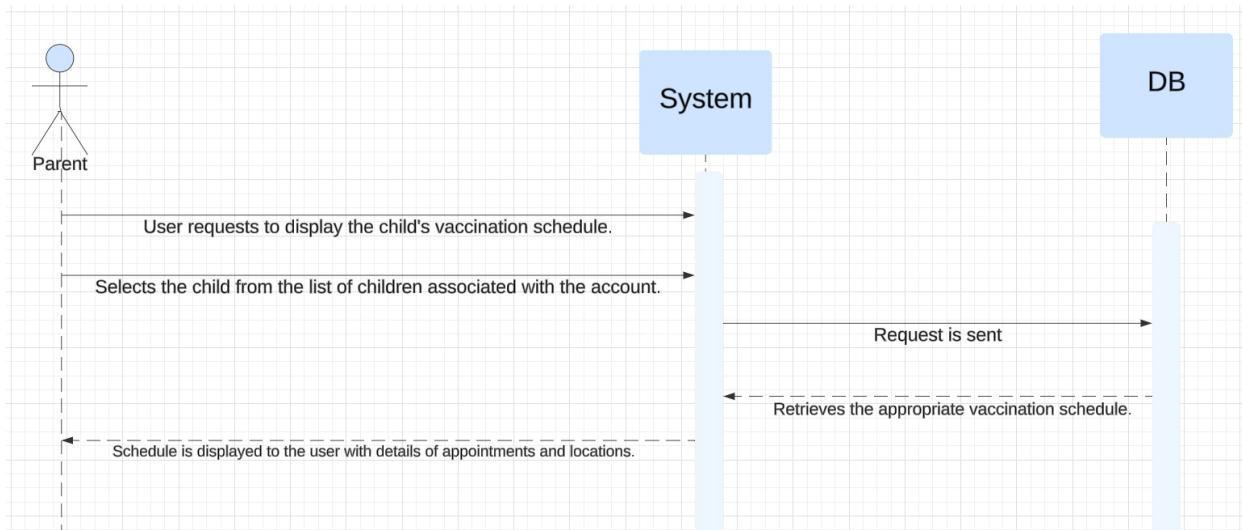


Figure 5.3.5

Appointment Booking Sequence Diagram

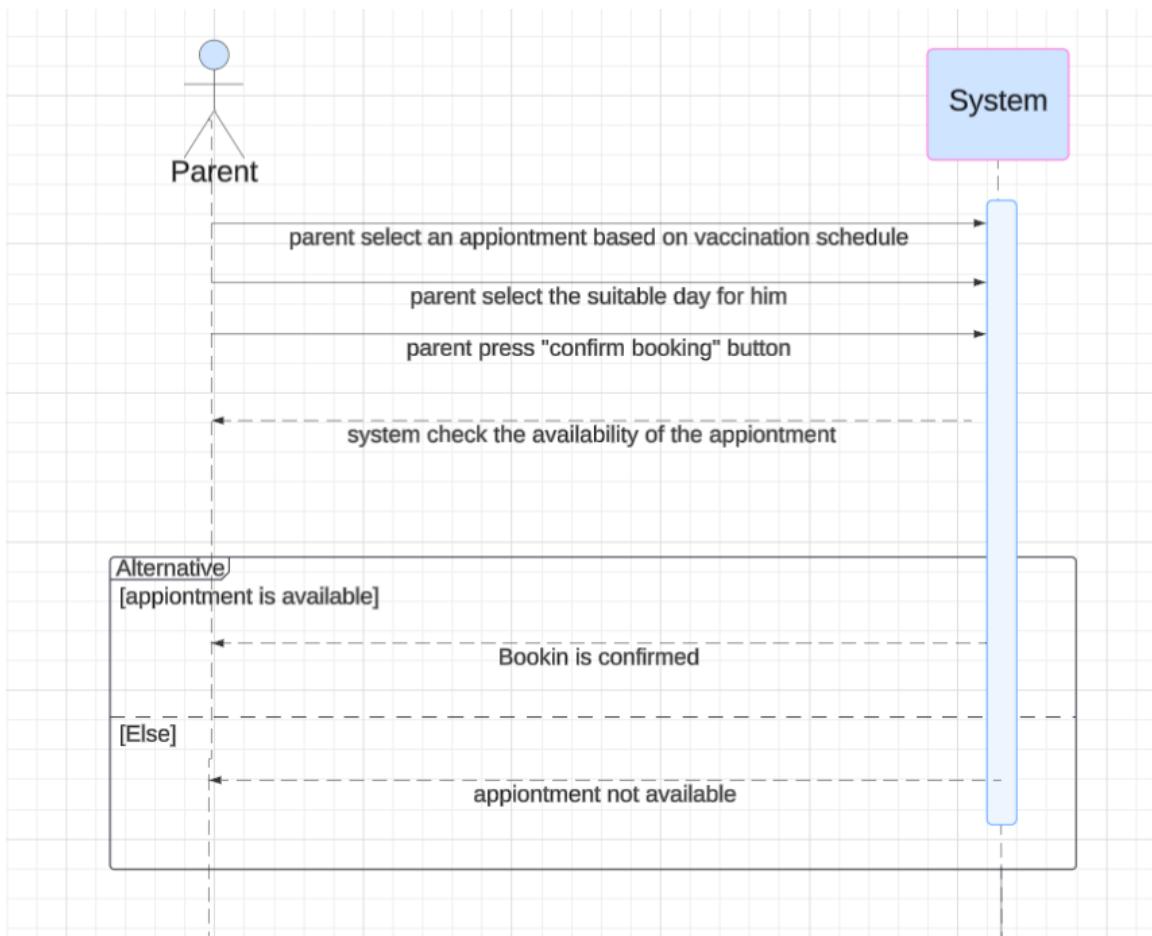


Figure 5.3.6

Appointment Management Sequence Diagram

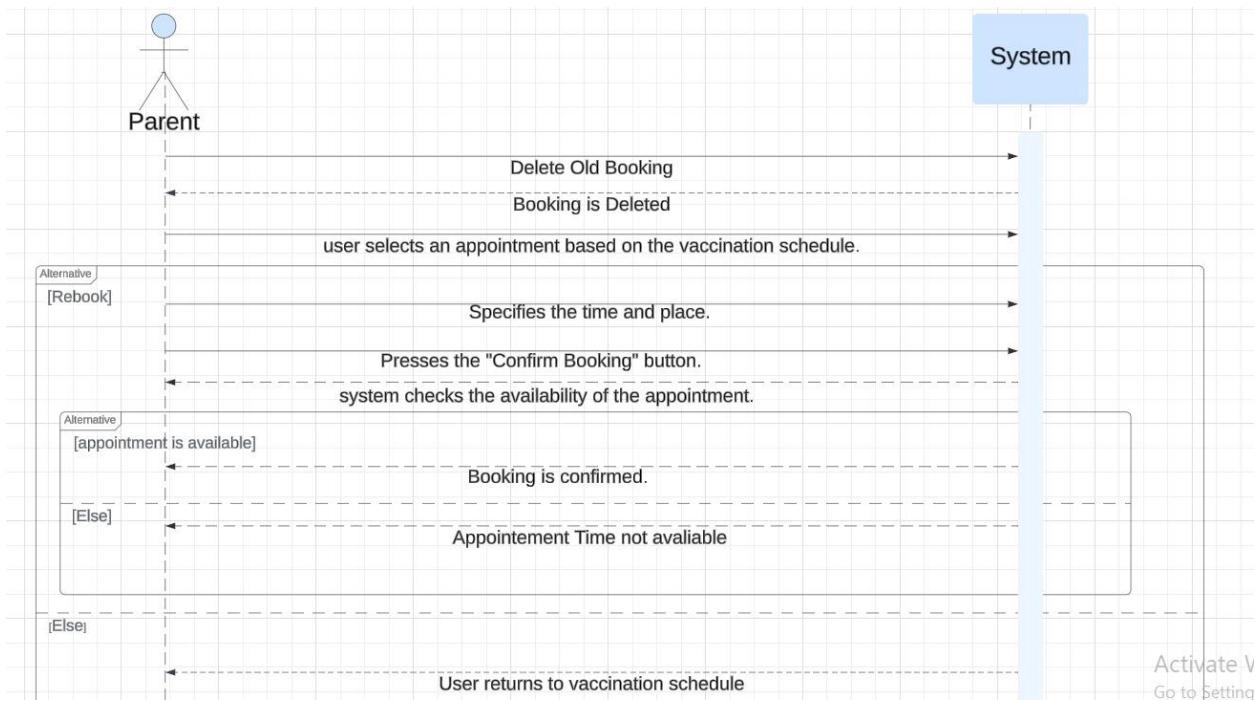


Figure 5.3.7

Report Side effect Diagram

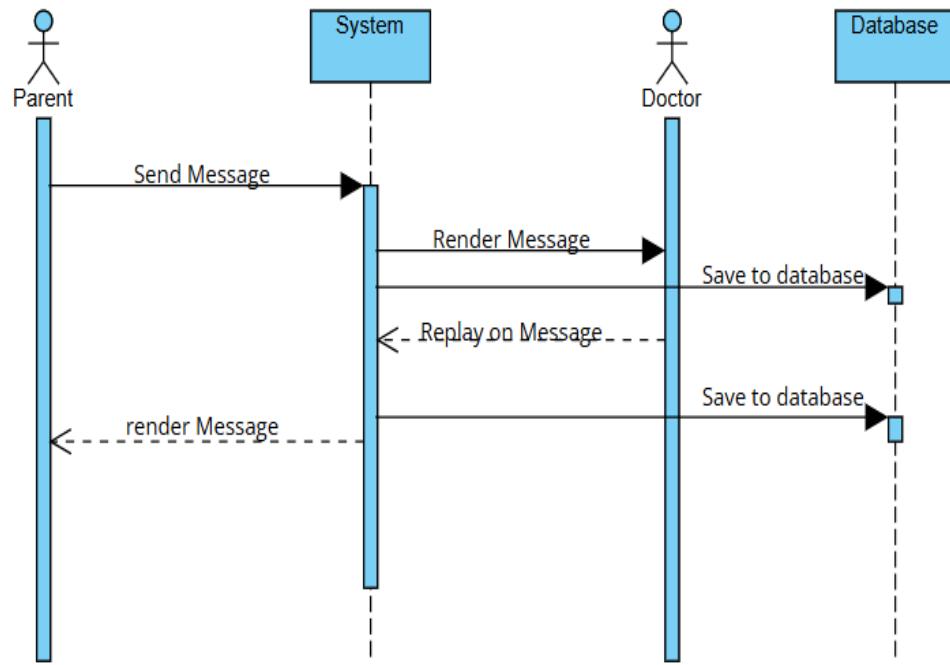


Figure 5.3.8

Request certificate printing Diagram

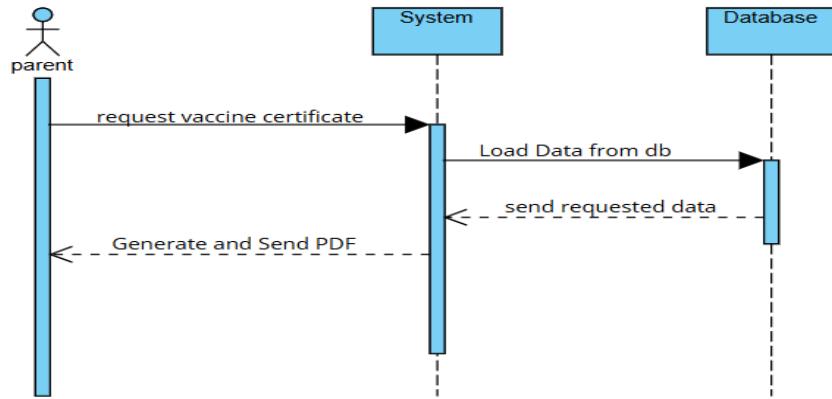


Figure 5.3.9

Make complaint on System Diagram

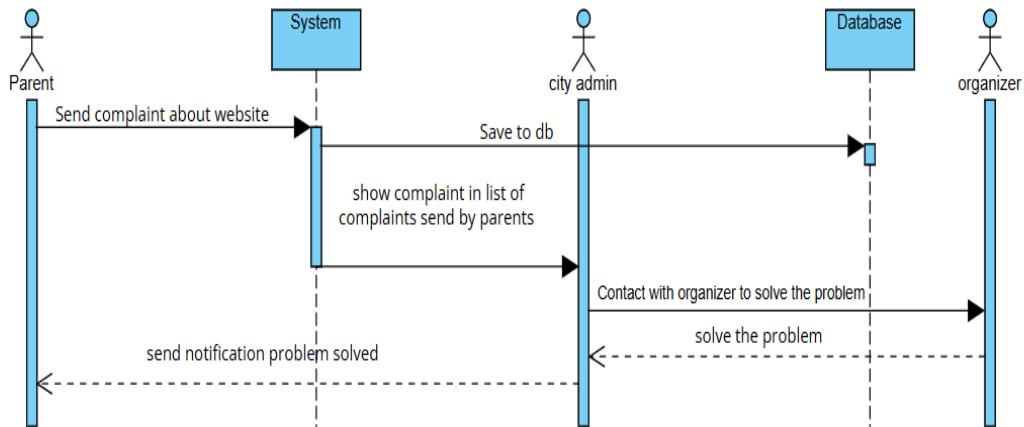


Figure 5.3.10

Adding Vaccine Schedule Diagram

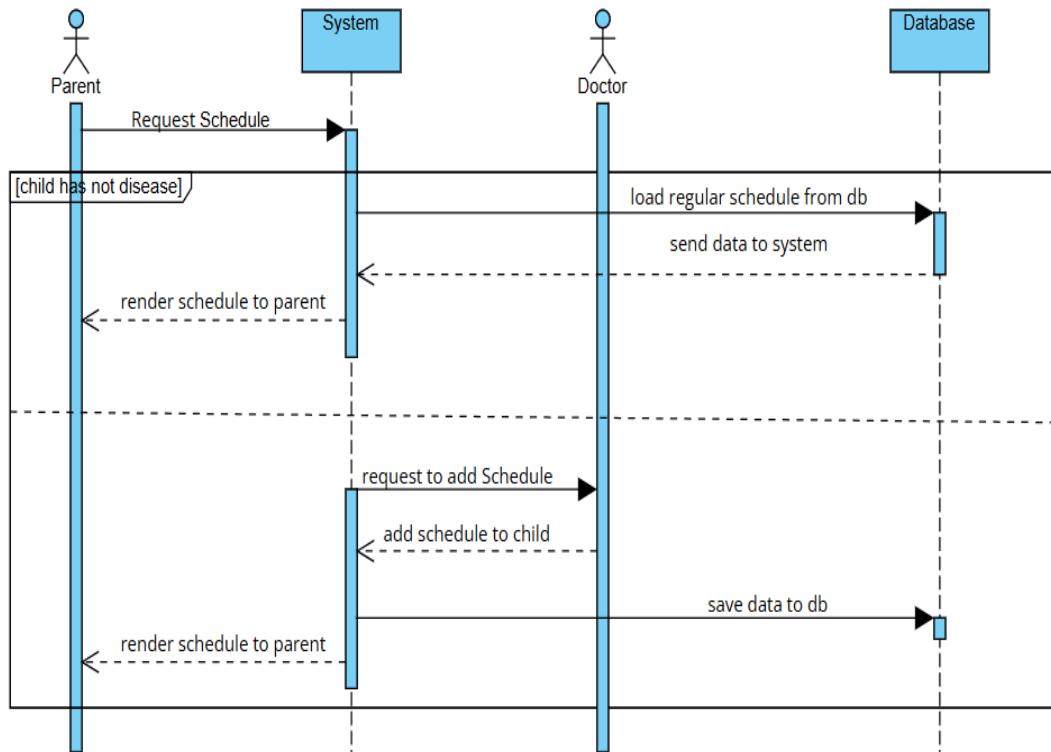


Figure 5.3.11

Access a child's profile Diagram

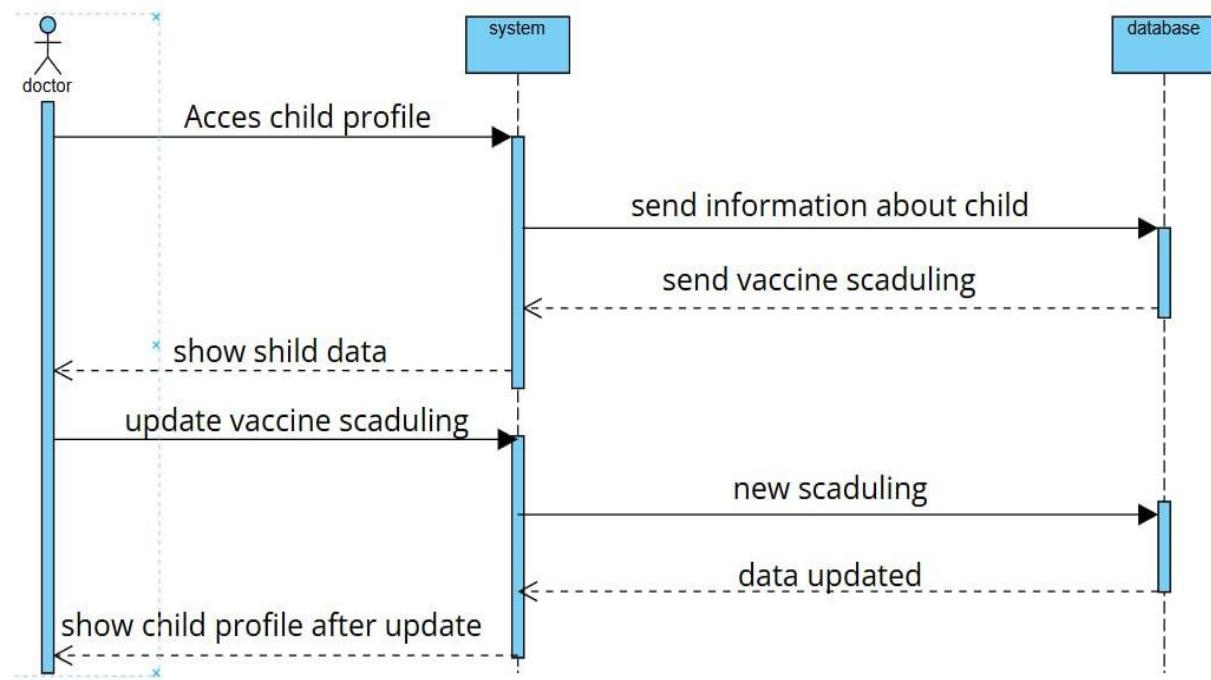


Figure 5.3.12

Update vaccine status Diagram

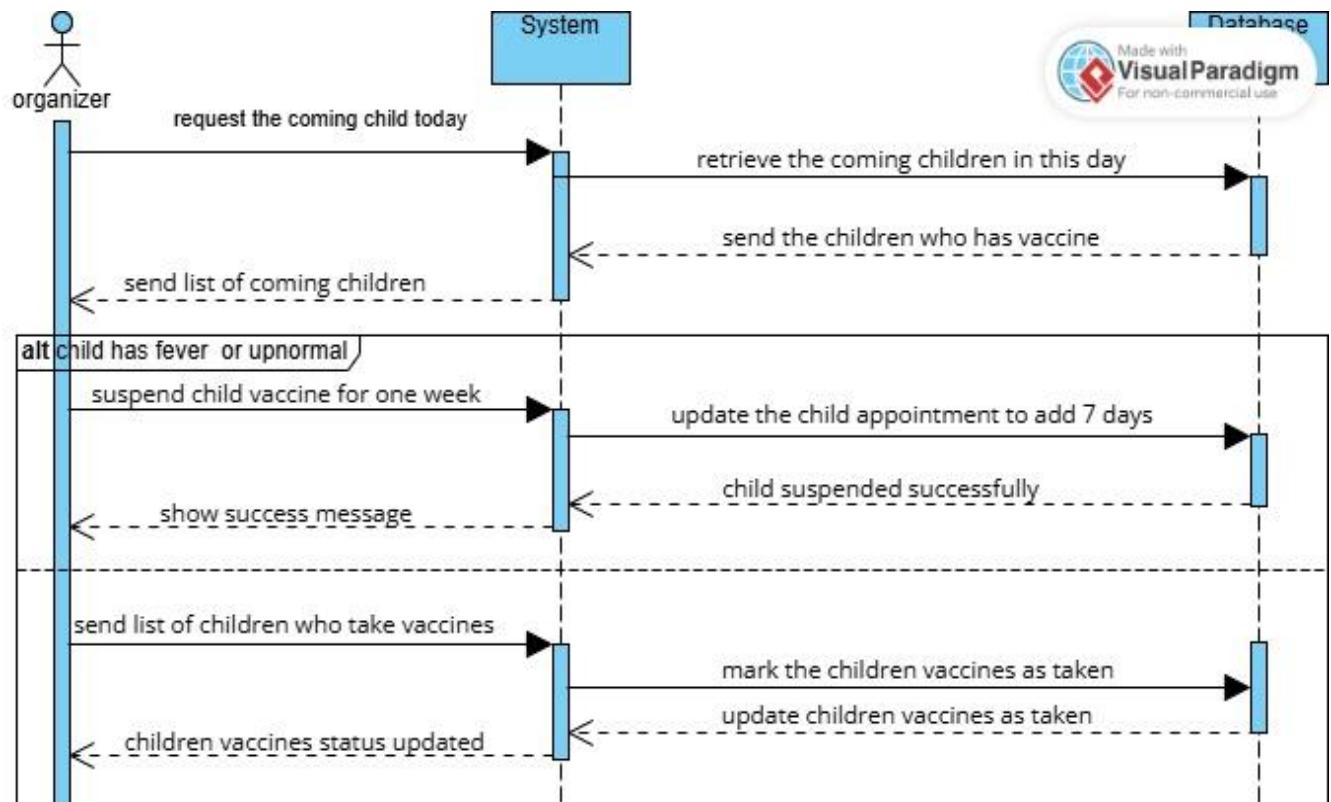


Figure 5.3.13

Vaccine Inventory Management Diagram

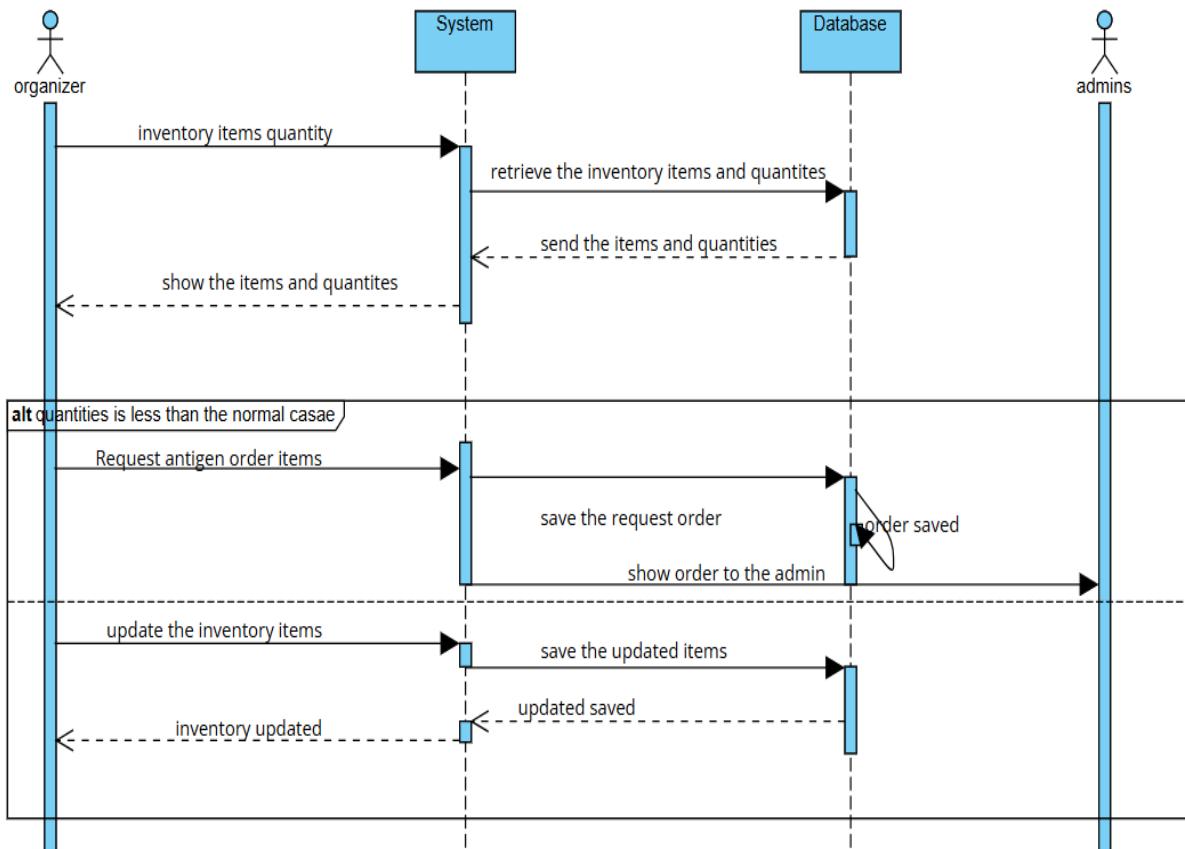


Figure 5.3.14

Vaccine Distribution Tracking Diagram

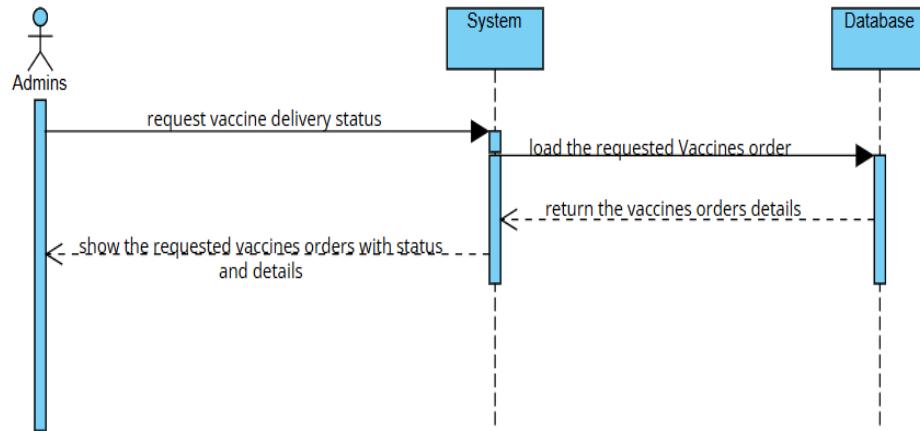
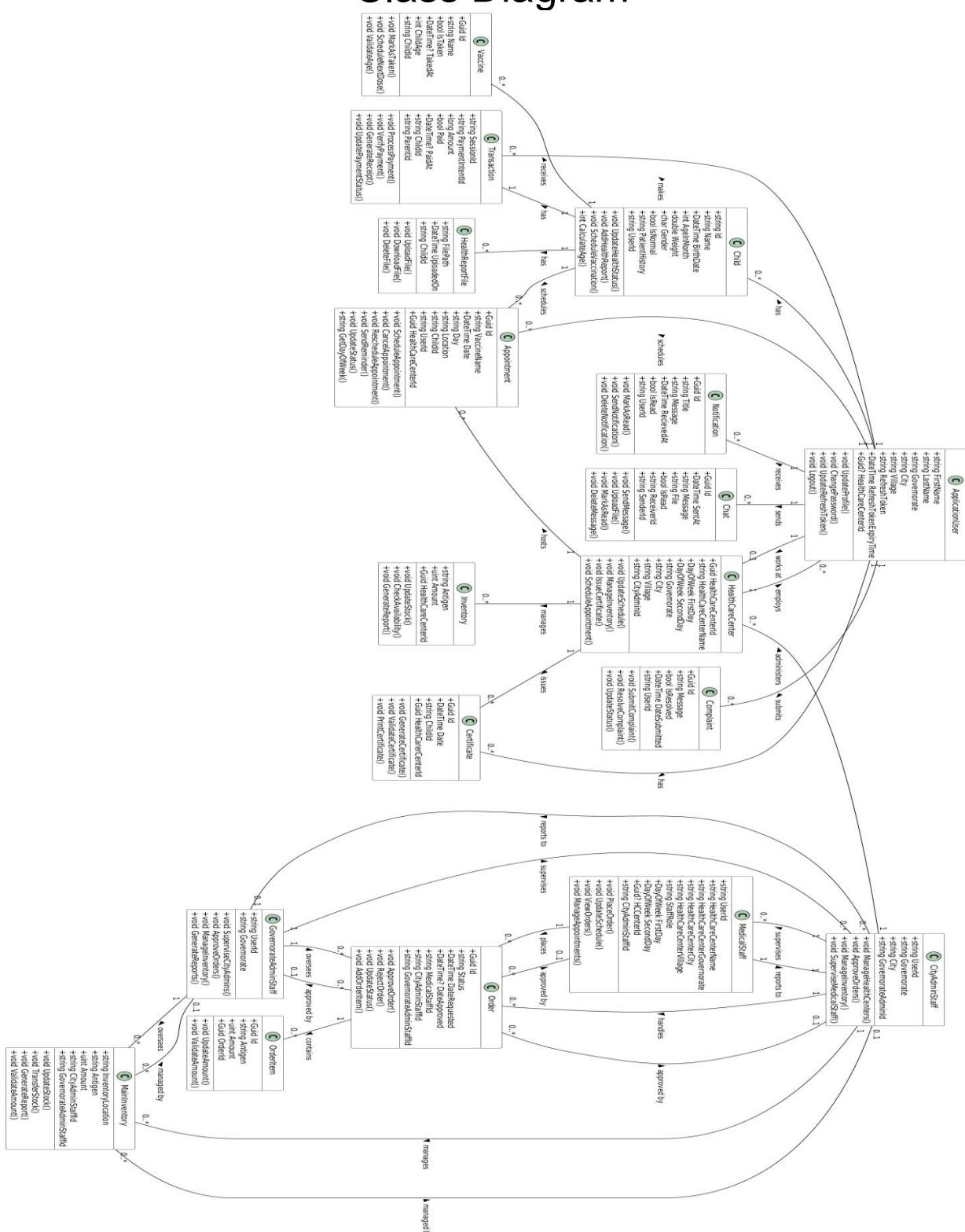


Figure 5.3.15

Class Diagram



Class Diagram figure 5.5.1

Activity Diagram

User Registration

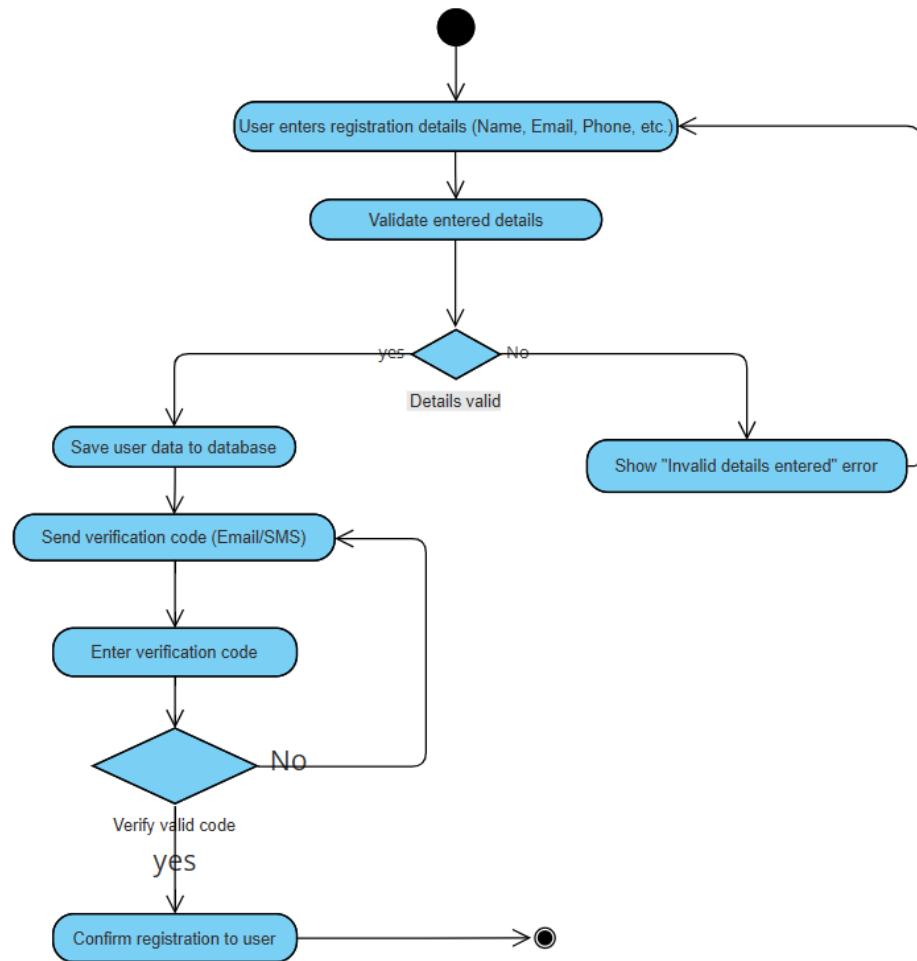


Figure 5.7.1

User login

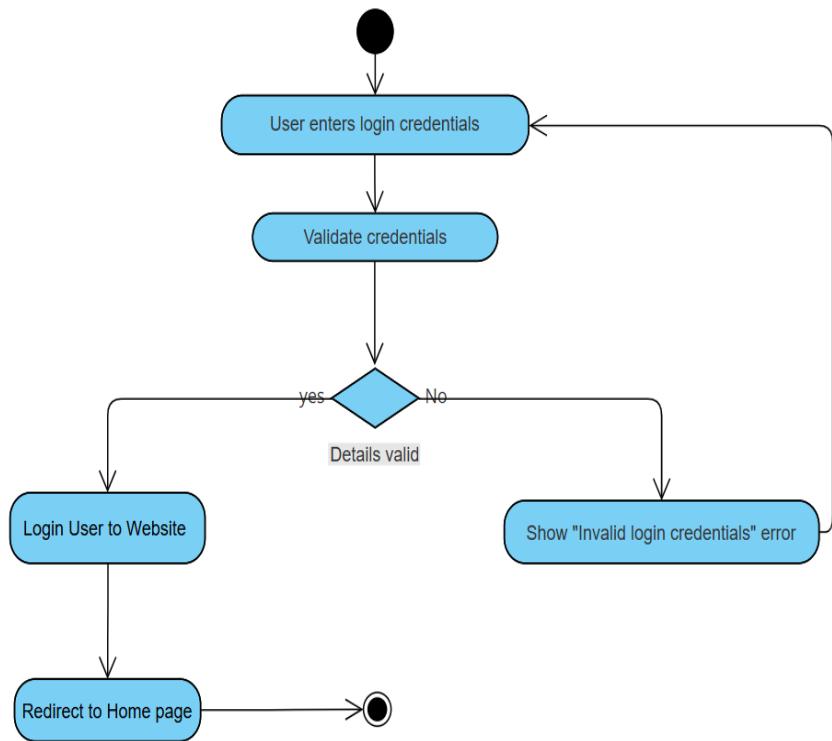


Figure 5.7.2

Adding Child and Schedule

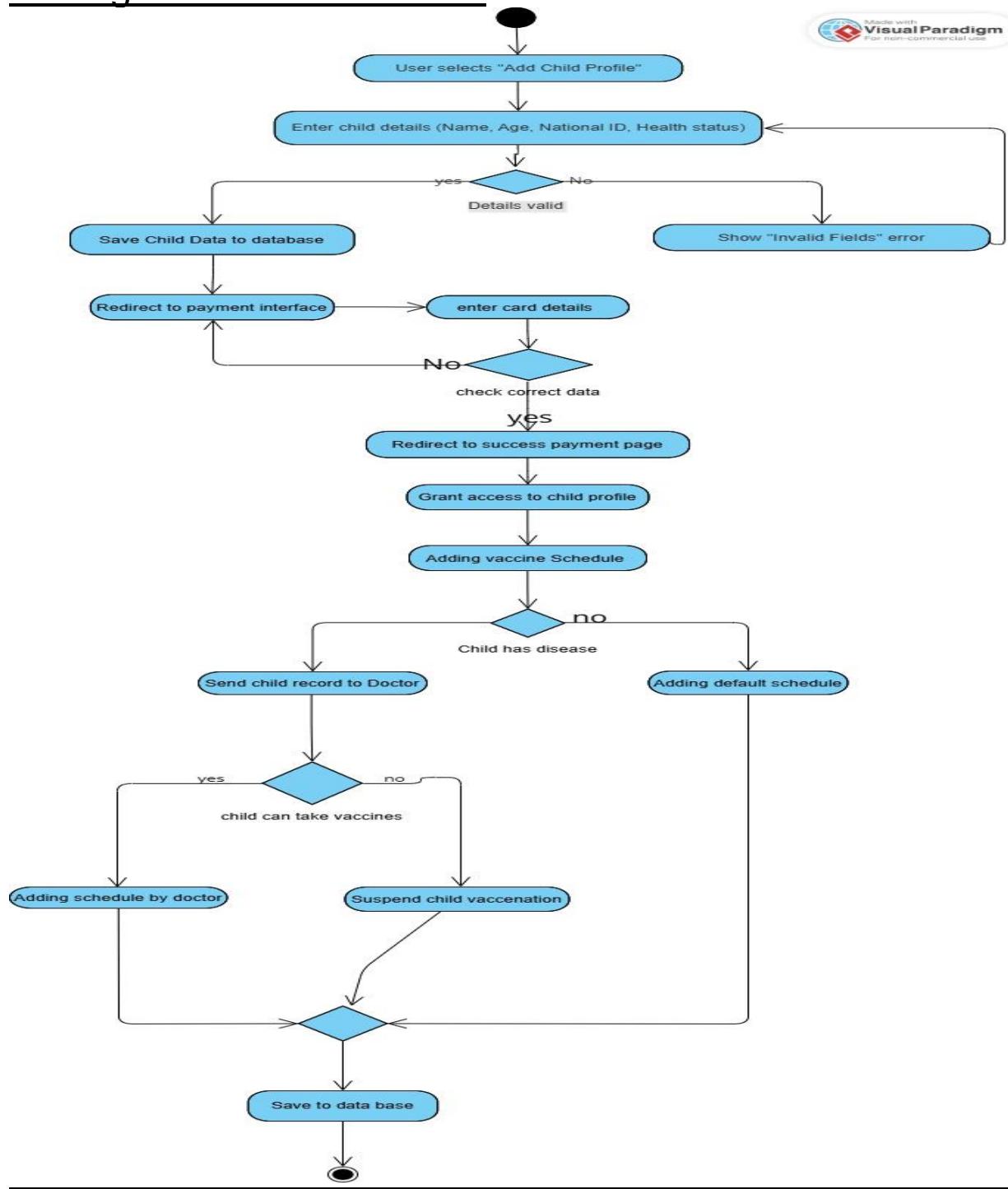


Figure 5.7.3

Update profile

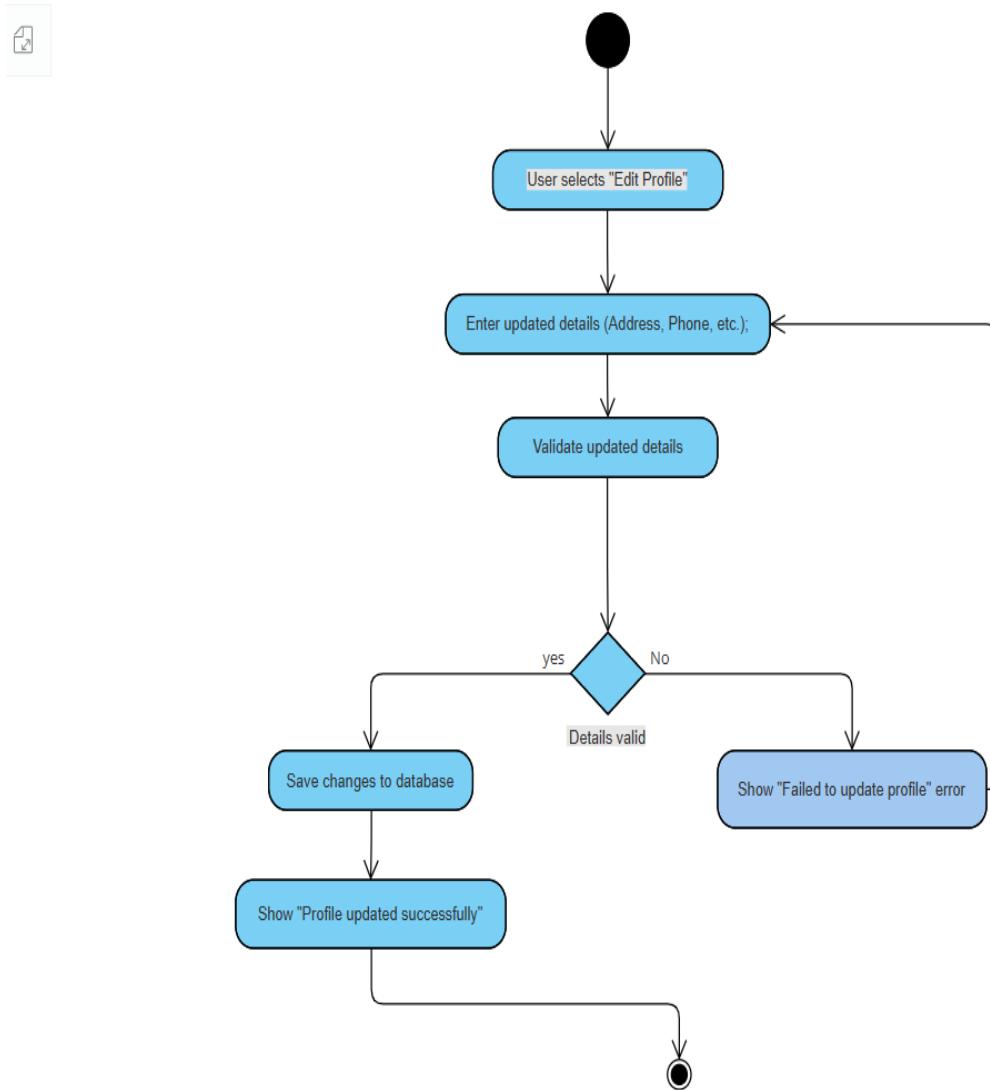


Figure 5.7.4

Book vaccine

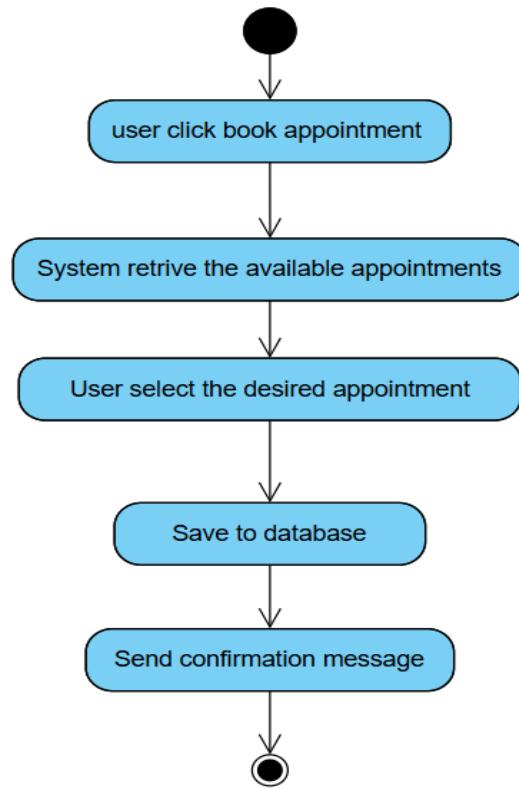
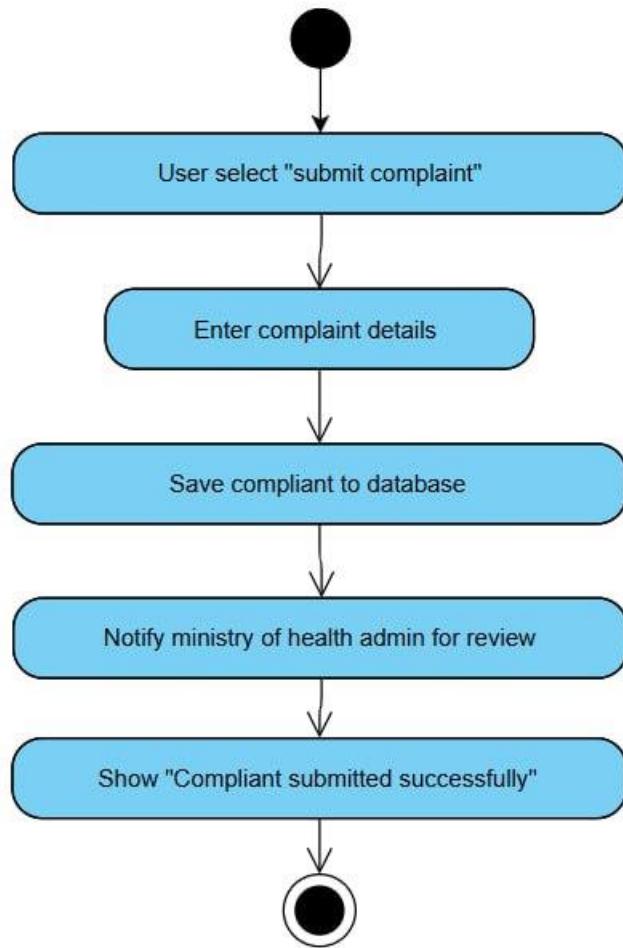
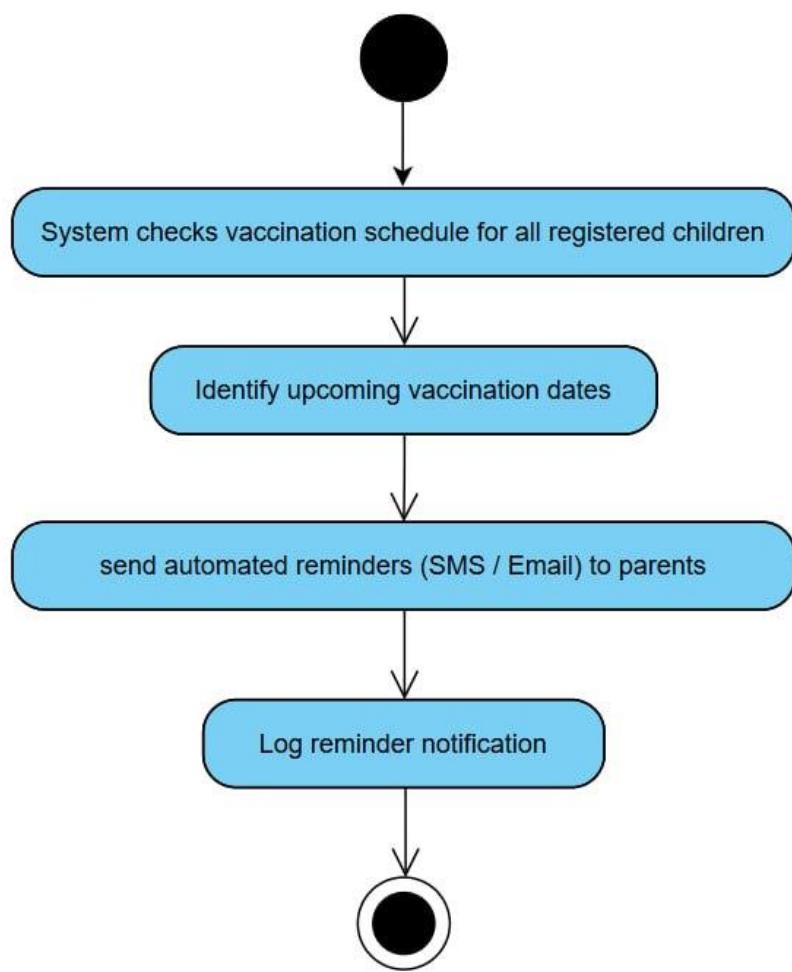


Figure 5.7.5



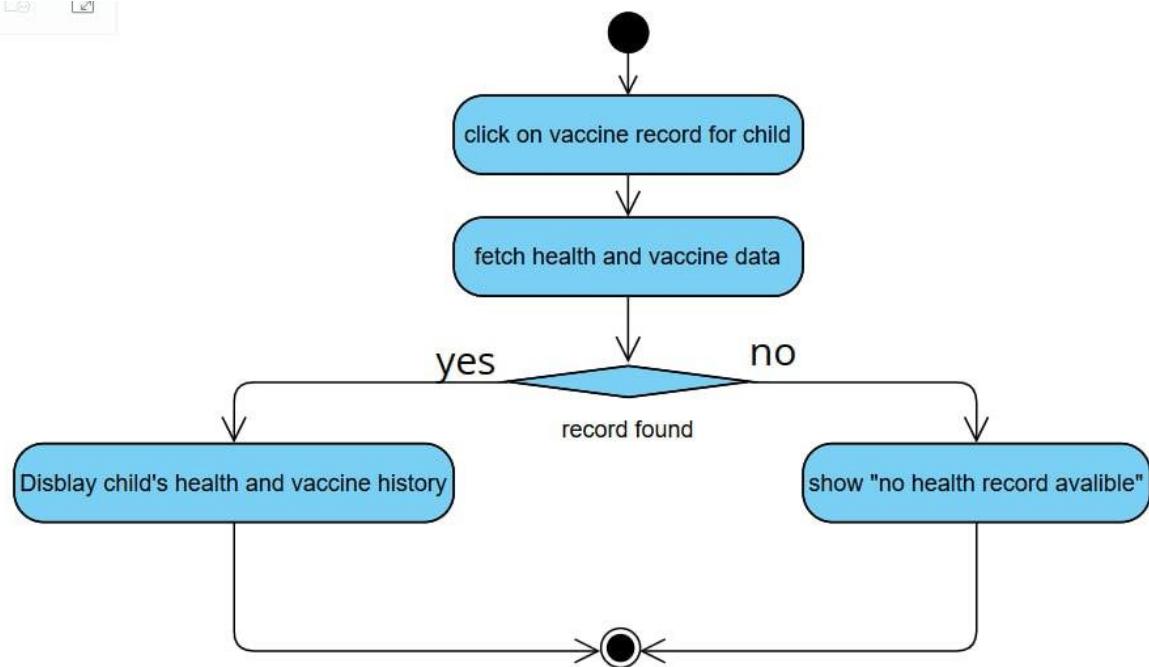
Handle Complaints

Figure 5.7.6



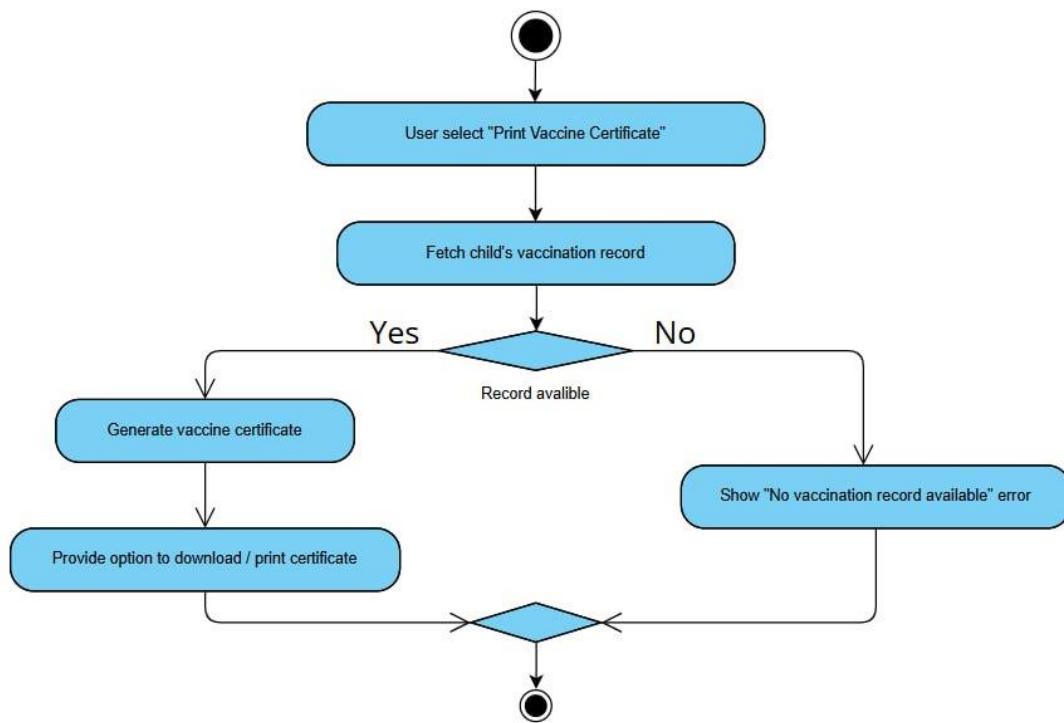
Send Reminder Notifications

Figure 5.7.7



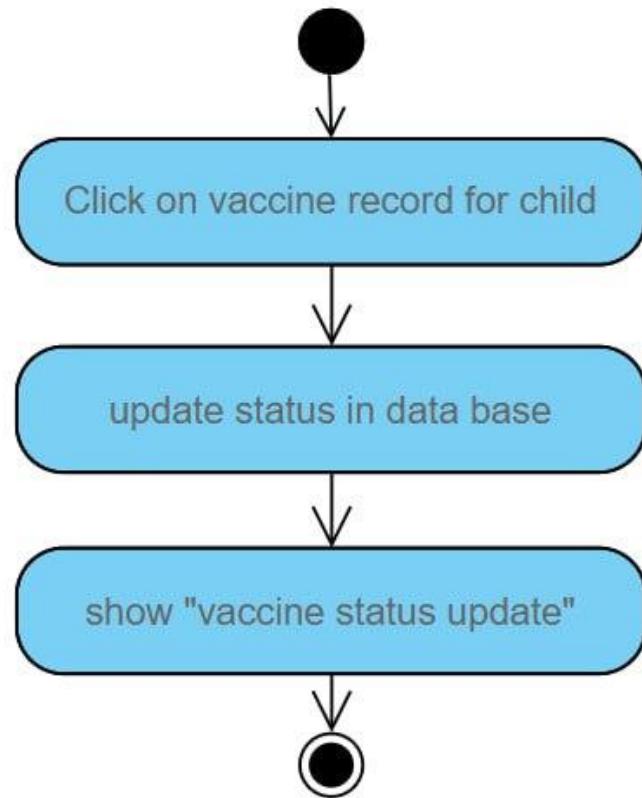
Track child's health

Figure 5.7.8



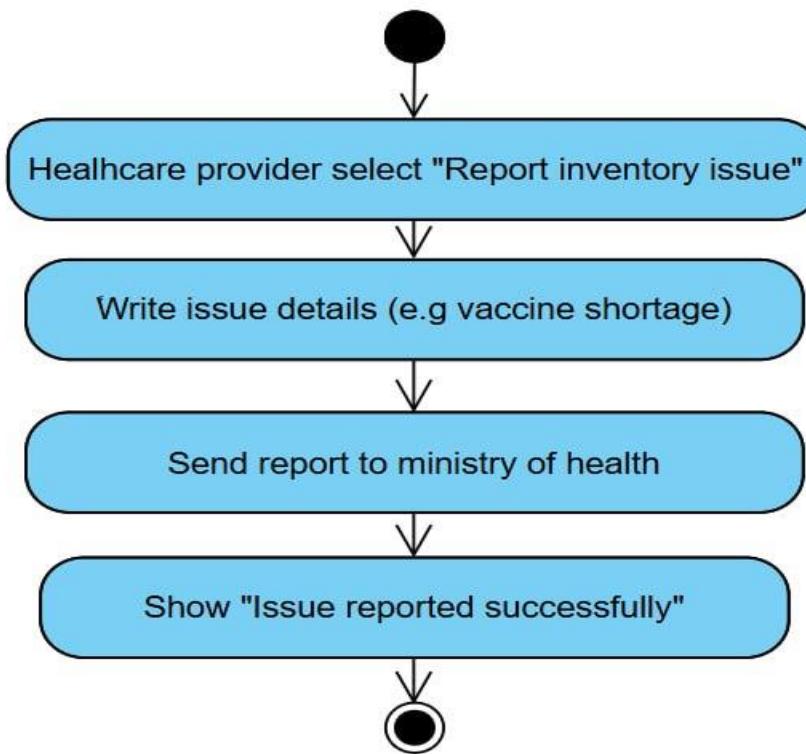
Vaccine Certification Printing

Figure 5.7.9



Edit vaccine status

Figure 5.7.10

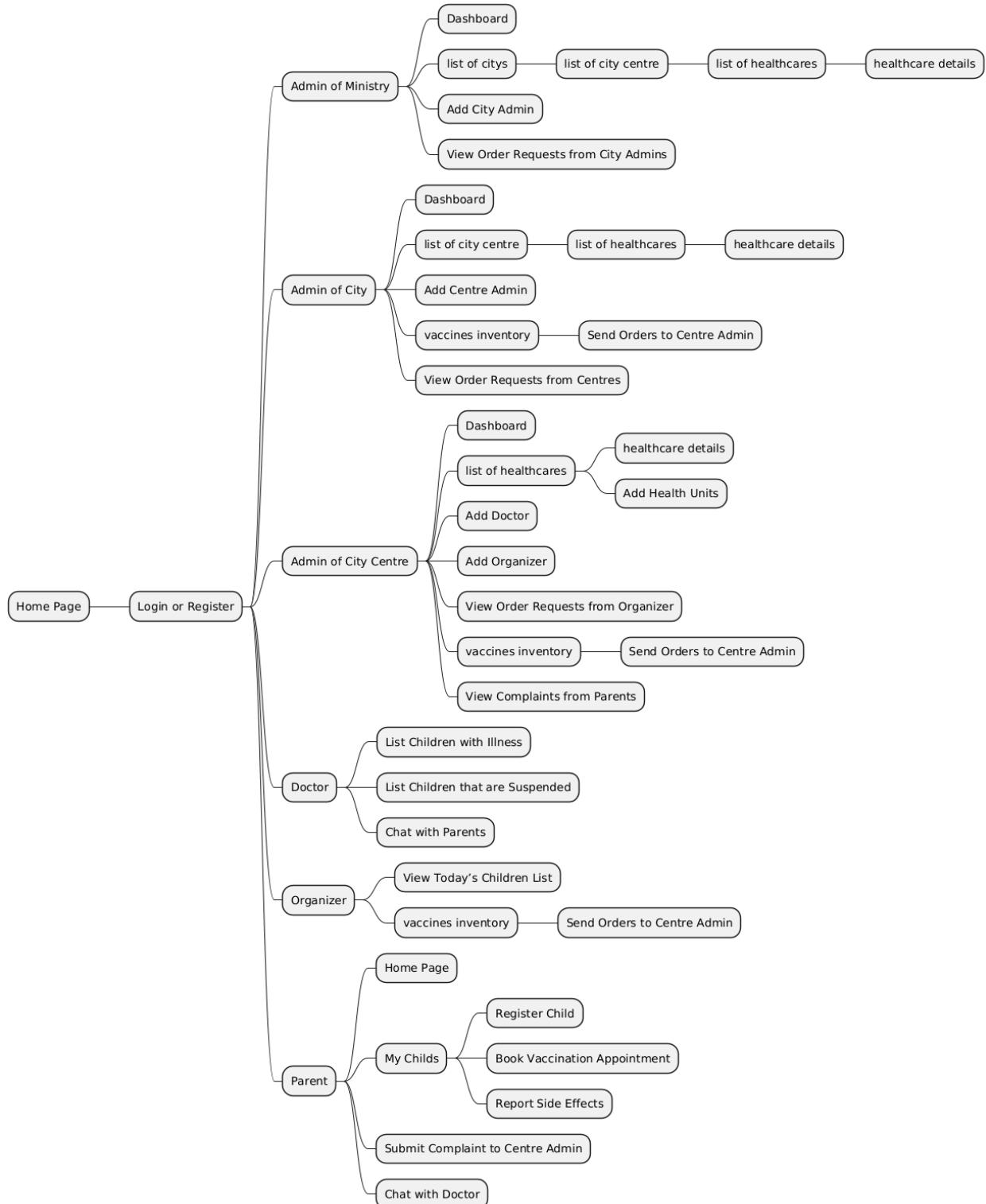


Report Inventory Issues

Figure 5.7.11

User Interface

Site Map:



Site map figure 5.8

Story board:

إنشاء حساب تسجيل دخول تواصل معنا الصفحة الرئيسية جدول التطعيمات إبرة آمان



موقع إدارة تطعيمات الأطفال
يؤمن إبرة آمان رعاية صحية ومتابعة لتطعيمات أطفالك ومتاح للجميع عبر الإنترنت بكل سهولة

عرض جدول التطعيمات

خدماتنا

البحث عن أقرب وحدة
يمكنك إثبات أمر بودرة صحية لك من بين الآلاف الوحدات حول الجمهورية

الاستشارة الطبية
إنسانة حانية مع أطبائنا الموثوق بهم والحصول على أفضل النصائح

متابعة التطعيمات
يمكنك متابعة وجدول لتطعيمات أطفالك عبر موقعنا

إدارة عملية التوزيع
إذاعة عملية توزيع التطعيمات في جميع أنحاء الجمهورية

التطبيقات
سريع عند دخول الماء من أحد أصالتك وتنسيبه أيضًا قبل الماء الذي تم دخوله

هل تريدين إدارة تطعيمات أطفالك بشكل منظم وسهل؟

إنشاء حساب

تابعنا على:  تواصل معنا على: hello@ebrataman.com

الصفحة الرئيسية جدول التطعيمات تواصل معنا

إبرة آمان
إدارة تطعيمات الأطفال بكل سهولة ويسر

Figure 5.9.1 about website

As the user enters the URL for the portal, he is directed to the home page. The home page is divided into a link bar that includes links to the following pages:

- "جدول التطعيمات"
- "من نحن"
- "تواصل معنا"

This link bar will appear in all pages of the portal. Then comes the header section of the page. The body is an introduction to the portal, and an image. The homepage will include a link to the login page and Register Page. The user may glance at the statistics section to gauge the platform's credibility and activity.

إربة أمان

إنشاء حساب

برجاء إدخال البيانات التالية لـإنشاء الحساب

الإسم الثاني	الإسم الأول
<input type="text"/> أدخل الإسم الثاني	<input type="text"/> أدخل الإسم الأول
البريد الإلكتروني	
<input type="text"/> someone@example.com	
أدخل بريد الكتروني صحيح	
رقم الهاتف	
<input type="text"/> +20 123 456 7890	
أدخل رقم هاتف صحيح	
كلمة المرور	
<input type="password"/> *****	
! يجب إدخال كلمة مرور لا تقل عن 8 أحرف ويجب أن تحتوي على حرف ورقم على الأقل	
تأكيد كلمة المرور	
<input type="password"/> *****	
! كلمات المرور غير متطابقة ، حاول مرة أخرى	
<input type="button" value="التالي"/>	

إربة أمان

إنشاء حساب

حدد موقعك وإختار الوحدة الصناعية المناسبة لك

<input type="button" value="اختر مركز"/> إختار مركز	<input type="button" value="اختر صناعية"/> إختار الصناعية
<input type="button" value="اختر الوحدة الصناعية"/> إختار الوحدة الصناعية	
<input type="button" value="إنشاء حساب"/> إنشاء حساب	

Figure 5.9.2 register

The Register page in which the user will enter his information and can choose the location he is exists in

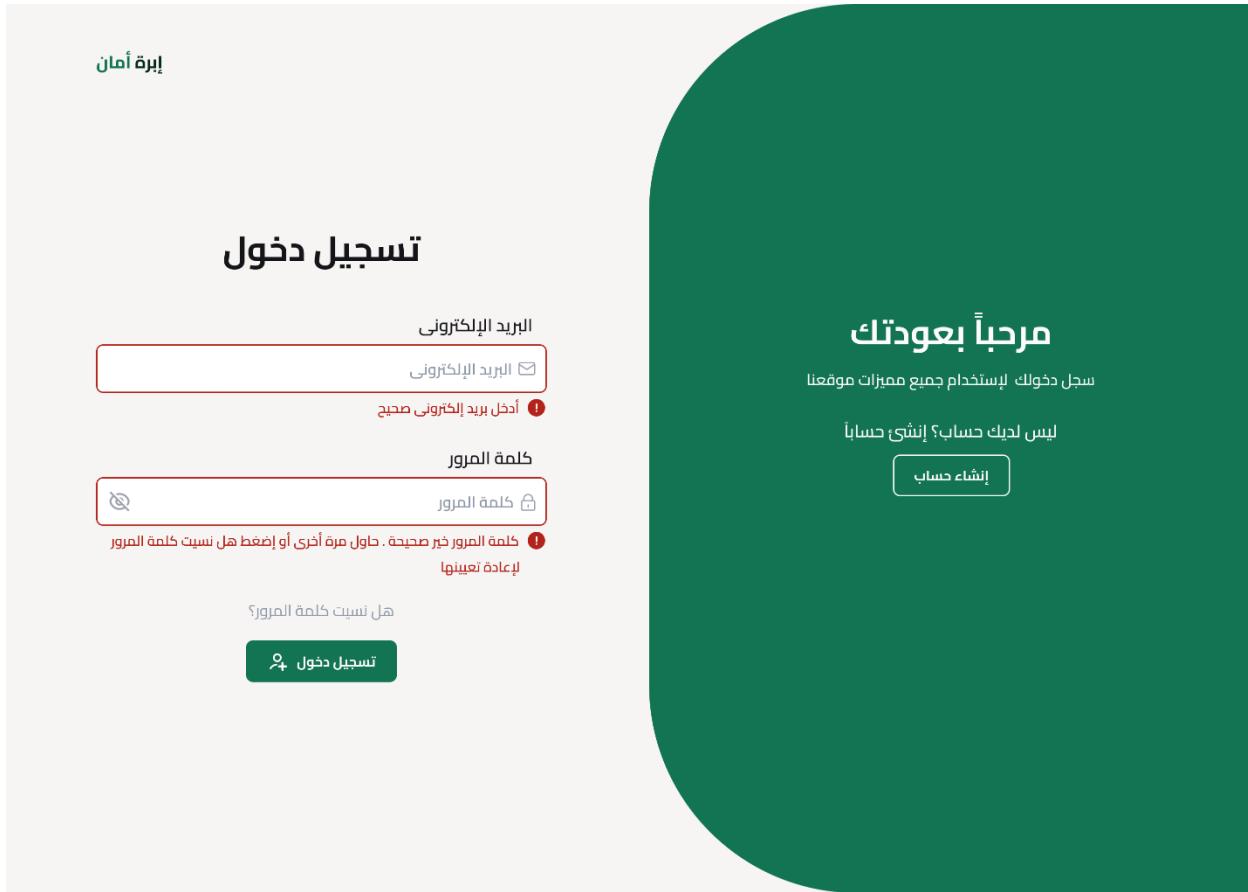


Figure 5.9.3 login

The login in page will include a login plug-in, in which the user will enter his Email and password. The user can directed to another form if he forgot his password or click create account



Figure 5.9.4 parent home page

The System supports 4 types of Users:

- Parent
- Doctor
- Admin of Healthcare
- Admin of Ministry of Health

If the user was a Parent, he will be directed to the Home Page, the home page is divided into a navbar that includes links to the following pages:

- "الرئيسية"
- "أطفالى"
- "الشكاوى"

This navbar will appear in all pages of the portal

Then Parents can overview of upcoming vaccinations for their children.

إضافة طفل

يرجاء إدخال البيانات التالية لإضافة طفل

الإسم

اسلام

الرقم القومي

30309012404196

تاريخ الميلاد

2025 / ابريل / 10

نوع

ذكر

الوزن

8 كغم

التالي

أهلاً وسهلاً

+1 رسائل

+5 إشعارات

إبلاغ عن شكوى

الرسائل

الإشعارات

الرئيسية

أهلاً وسهلاً

اسلام محمد
ah@example.com

Figure 5.9.5 add child page

parents can add a new child to their profile by filling in required details.

إضافة طفل

يرجاء إدخال البيانات التالية لـإضافة طفل

التاريخ المرضي (اختياري)

الإضافة المرضي

هل تزيد إضافة صور تحاليل أو إشعاعات [إضافة صور](#)

هل حصل الطفل على لقاحات سابقة؟ [إضافة لقاحات](#)

إضافة طفل

رجوع

إبرة أمان

الرئيسية

أطفال

إبلاغ عن شكوى

+1 الرسائل

+5 الإشعارات

اسلام محمد ah@example.com

Figure 5.9.6 child health record

Parents fill in medical history (if any) and confirm if the child has taken previous vaccinations, then complete the registration process and then the parent directed to payment page.

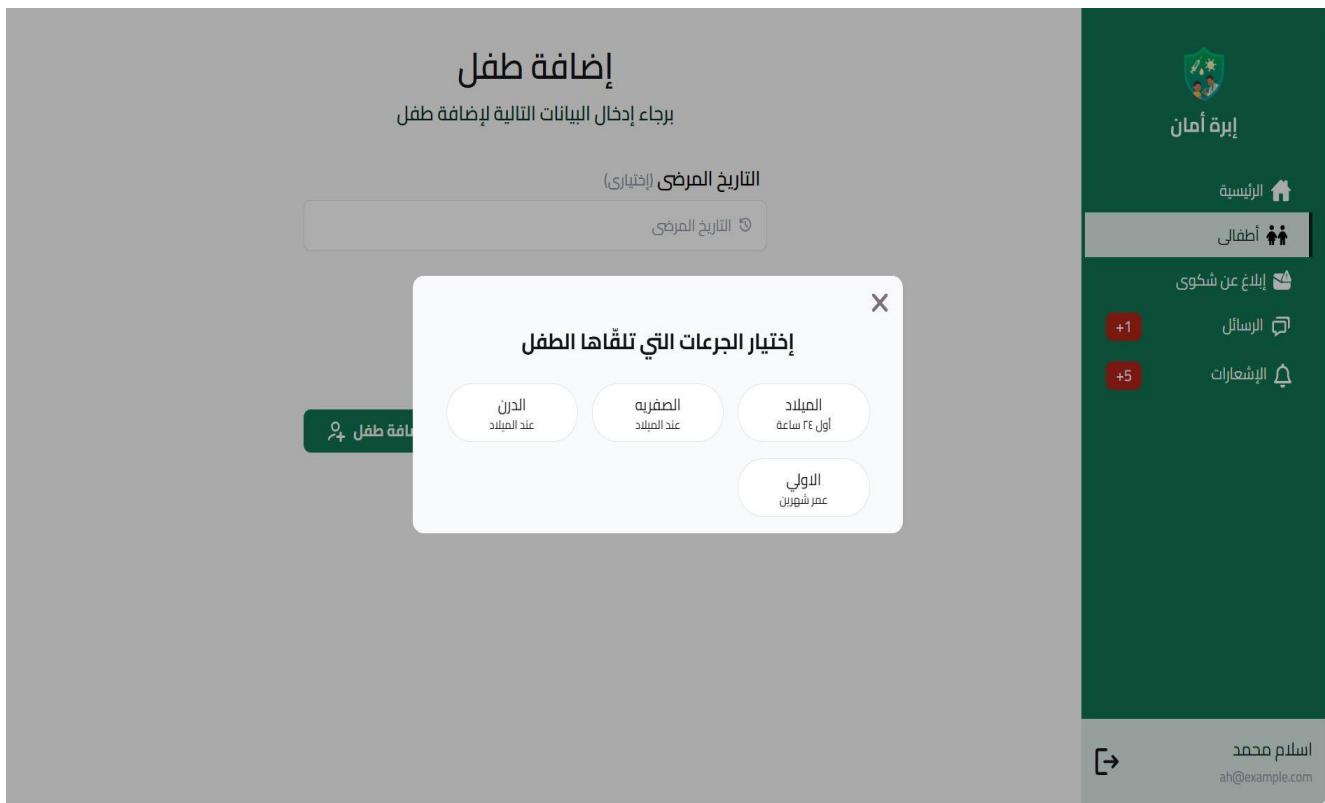


Figure 5.9.7 child taked vaccines

This screen allows the parent to select the vaccines their child has already received before registering on the platform. It displays a checklist of common vaccines, and the parent can mark the ones that apply.

The information provided helps create an accurate medical history for the child and ensures that future vaccination schedules avoid duplicate doses.

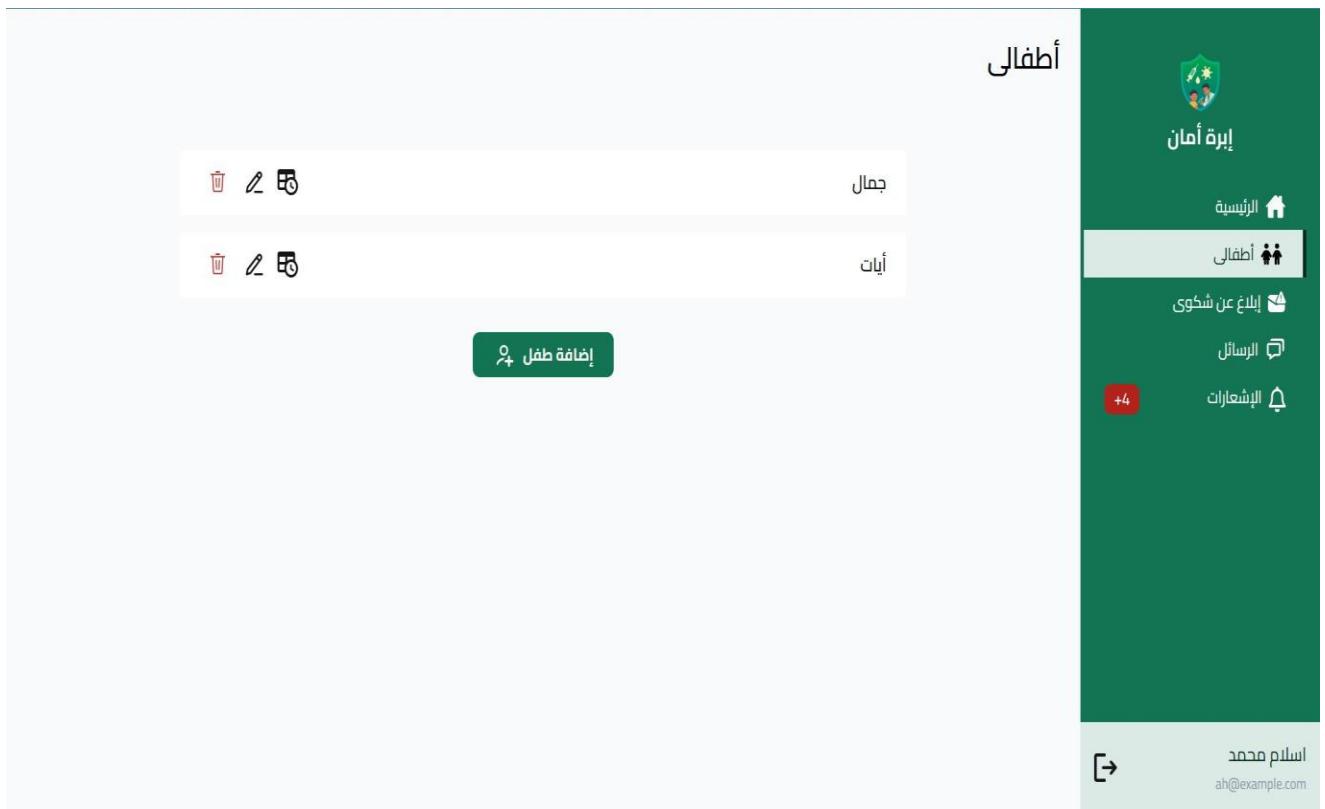


Figure 5.9.8 parent children page

Parents can view and manage their registered children's information.

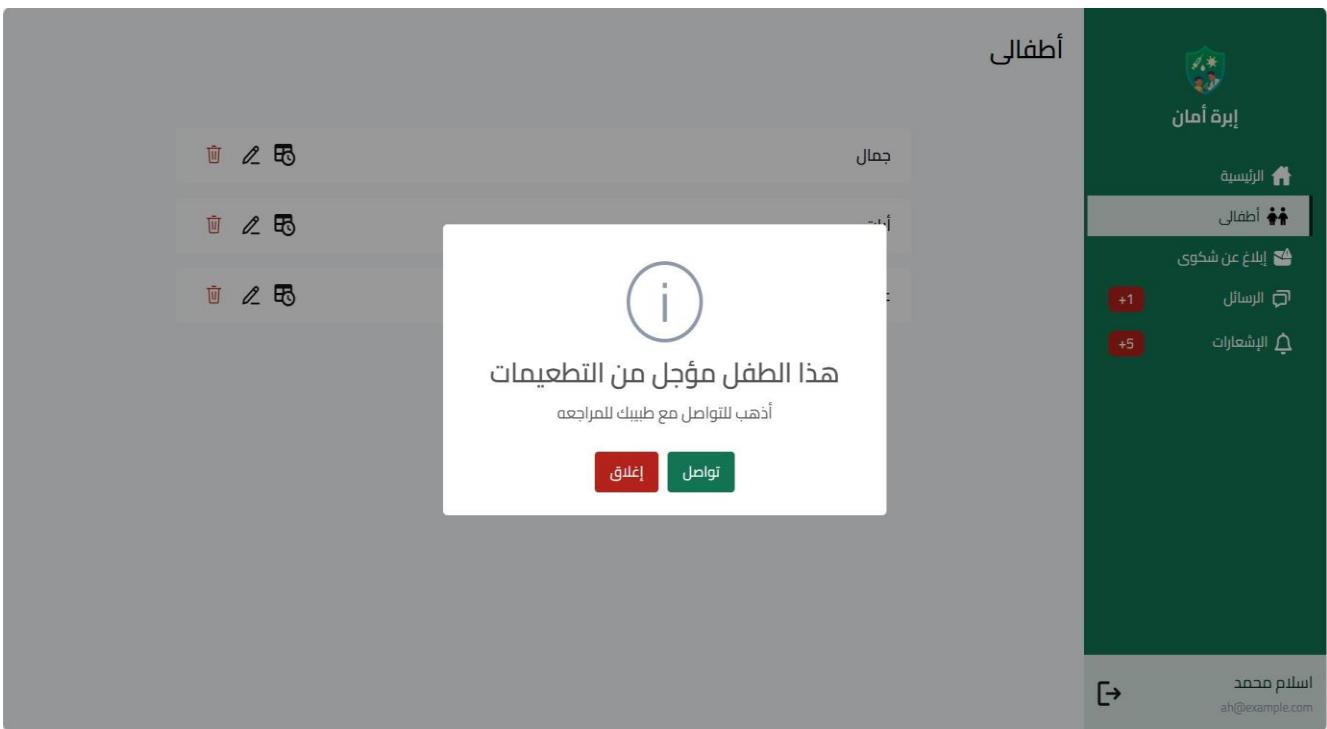
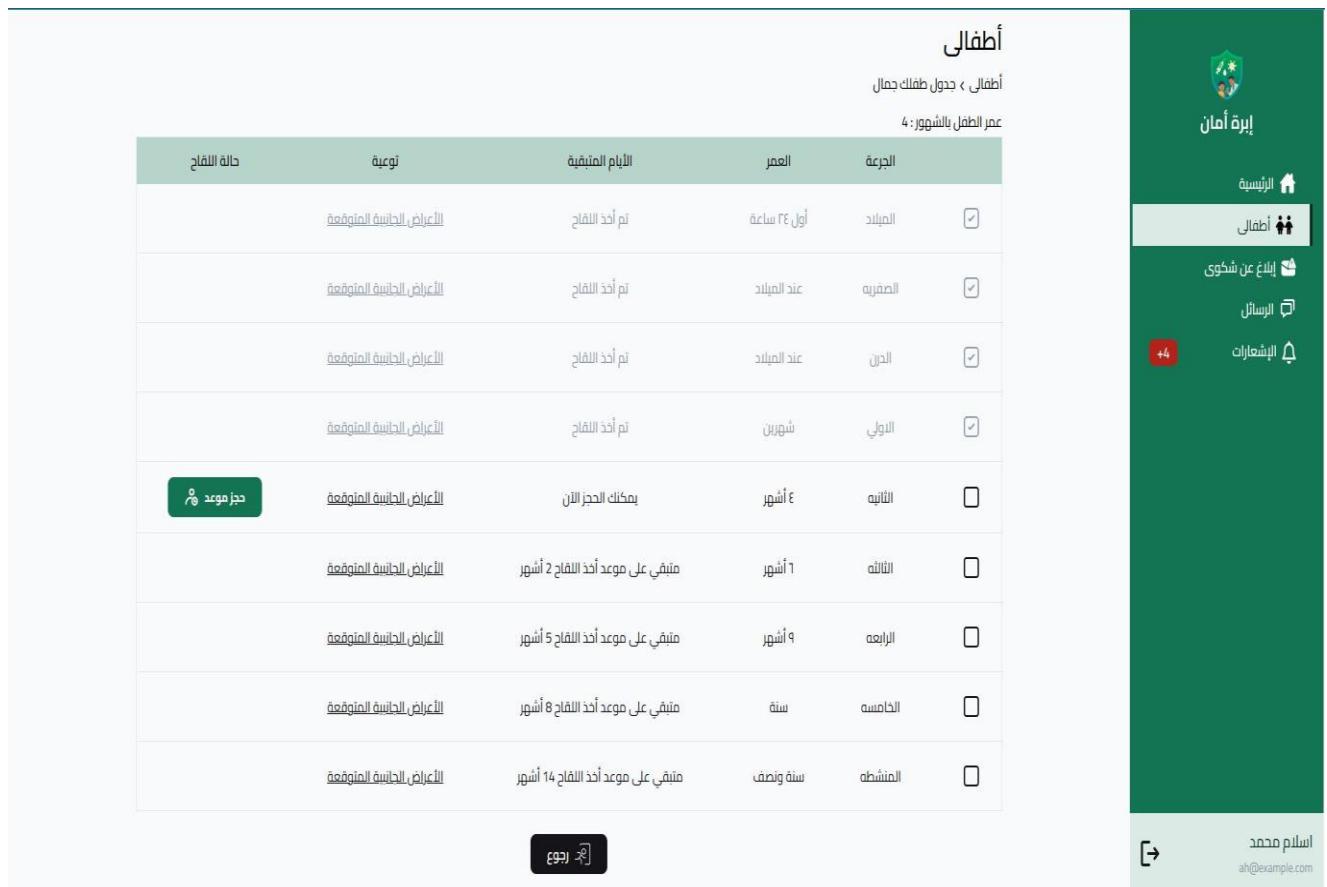


Figure 5.9.9 suspended child



The screenshot shows a vaccination schedule for a child. At the top right, there is a green header bar with icons for a shield, a user profile, and navigation options like 'الرئيسية' (Home), 'أطفال' (Children), 'إبلاغ عن شفوي', 'الرسائل' (Messages), and 'الشعارات' (Logos). Below the header is a title 'أطفال' (Children) with a sub-link 'أطفال > جدول طفلك جمال'. A note indicates the child's age: 'عمر الطفل بالشهر: ٤'. The main content is a table titled 'الجرعات' (Doses) with columns: 'حالة اللقاح' (Vaccination Status), 'نوعه' (Type), 'الأيام المتبعة' (Follow-up Days), 'العمر' (Age), and 'الجرعة' (Dose). The table lists eight rows of vaccination information, each with a checkbox column. A green button labeled 'جز موعد' (Book appointment) is located on the left side of the table. At the bottom right, there is a sign-in button with the text 'سجل' (Sign in) and an email placeholder 'ah@example.com'.

حالة اللقاح	نوعه	الأيام المتبعة	العمر	الجرعة
تم أخذ اللقاح	للاعراض الدائمة الممنوعة	تم أخذ اللقاح	أول ٢٤ ساعة	الميلاد
تم أخذ اللقاح	للاعراض الدائمة الممنوعة	تم أخذ اللقاح	عند الميلاد	المصرفيه
تم أخذ اللقاح	للاعراض الدائمة الممنوعة	تم أخذ اللقاح	عند الميلاد	الدرين
تم أخذ اللقاح	للاعراض الدائمة الممنوعة	تم أخذ اللقاح	شتواء	النول
جز موعد ٩	للاعراض الدائمة الممنوعة	يمثلك الحجز الان	٤ أشهر	الثانية
متأخر عن موعد أول اللقاح ٢ أشهر	للاعراض الدائمة الممنوعة	متأخر عن موعد أول اللقاح ٢ أشهر	٦ أشهر	الثالثة
متأخر عن موعد أول اللقاح ٥ أشهر	للاعرض الدائمة الممنوعة	متأخر عن موعد أول اللقاح ٥ أشهر	٩ أشهر	الرابعة
متأخر عن موعد أول اللقاح ٨ أشهر	للاعرض الدائمة الممنوعة	متأخر عن موعد أول اللقاح ٨ أشهر	سنة	الخامسة
متأخر عن موعد أول اللقاح ١٤ أشهر	للاعرض الدائمة الممنوعة	متأخر عن موعد أول اللقاح ١٤ أشهر	سنة ونصف	الستة

Figure 5.9.10 child schedule

Display a detailed schedule of vaccinations for a child, including status, side effects, and remaining days. Parents view the vaccination timeline, track upcoming doses, and can reschedule or book appointments directly from the table.

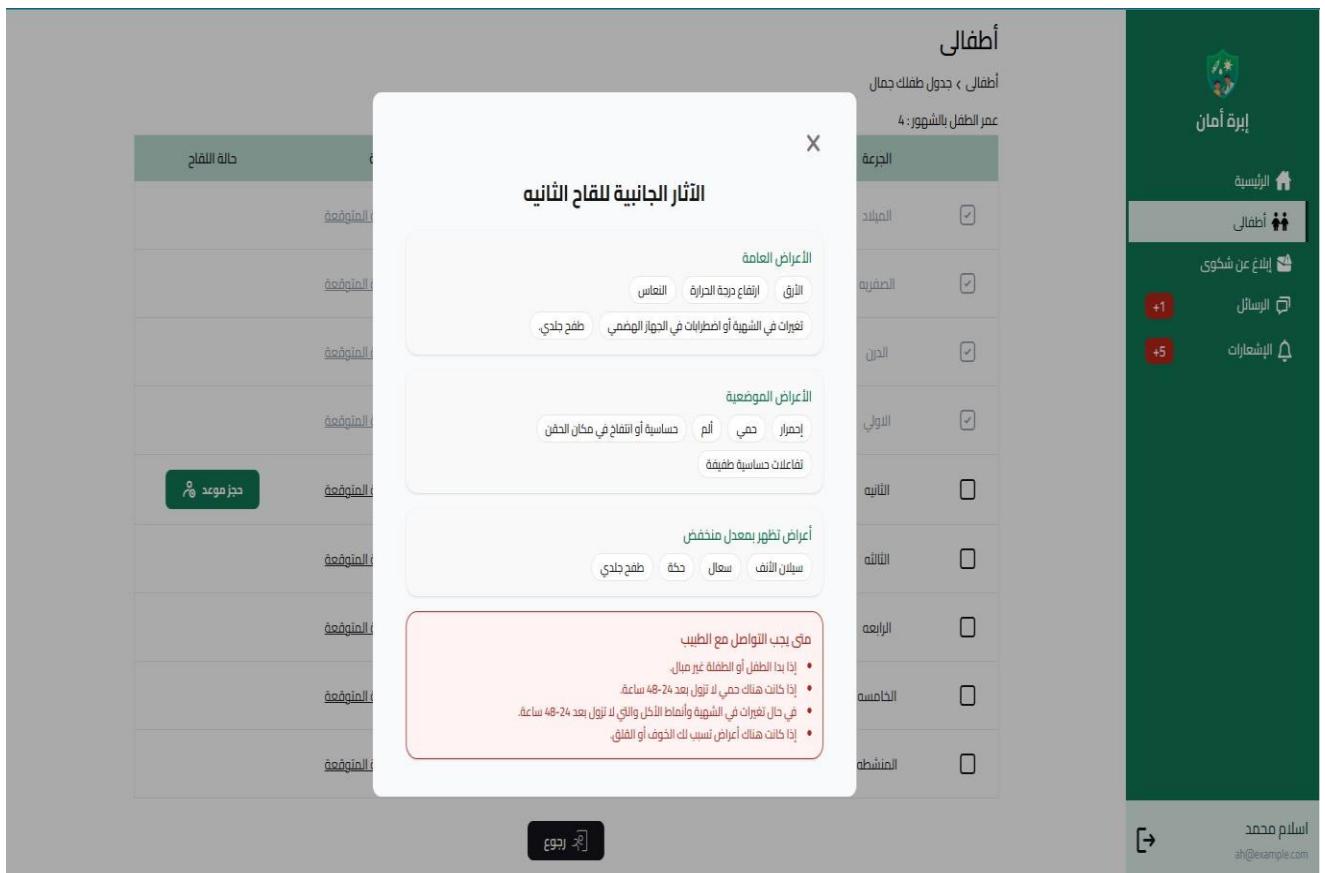


Figure 5.9.11 vaccine side effect

Inform parents about expected side effects of vaccinations and allow them to report any issues.

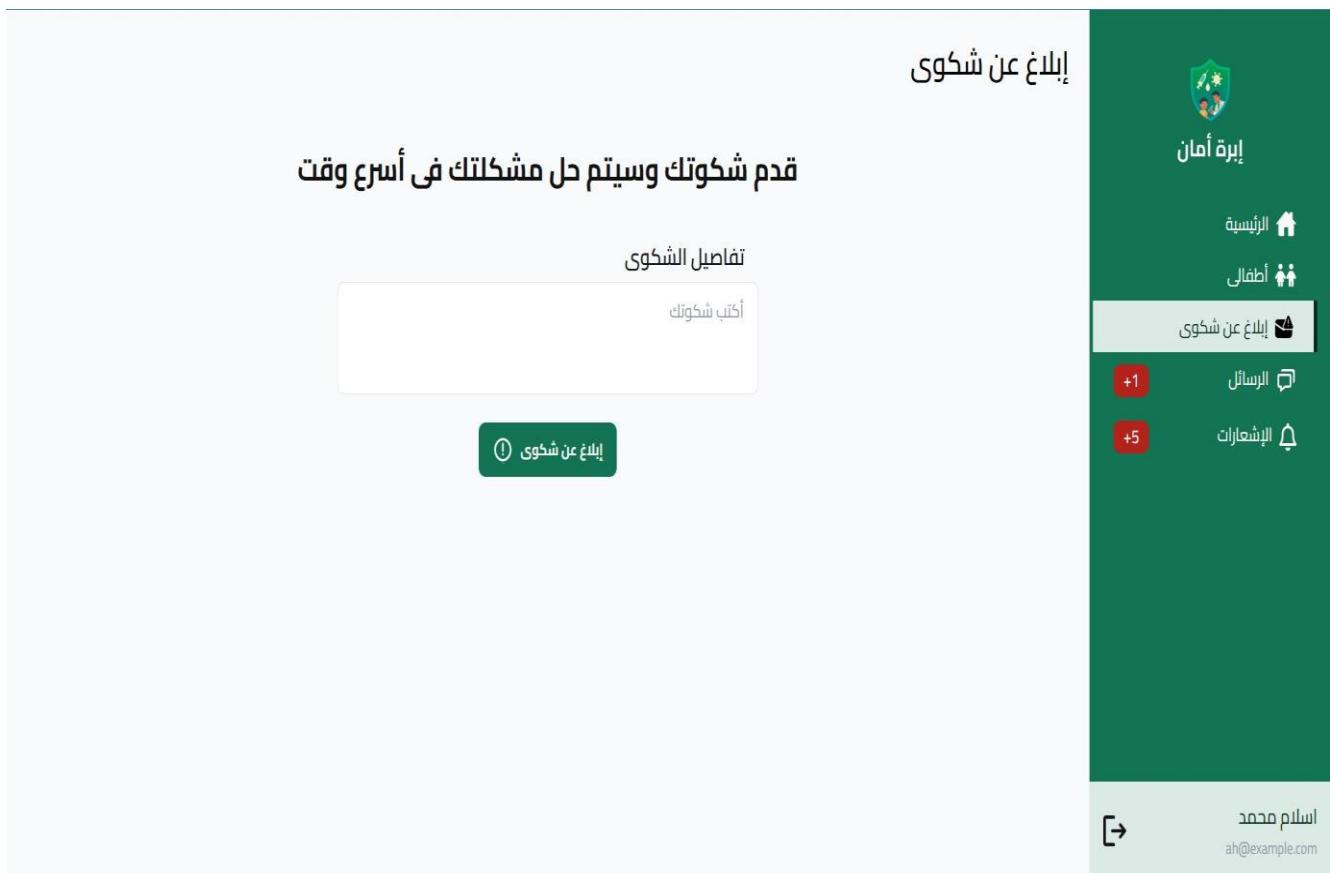


Figure 5.9.12 complaint page

Allow parents to report complaints related to vaccination services.

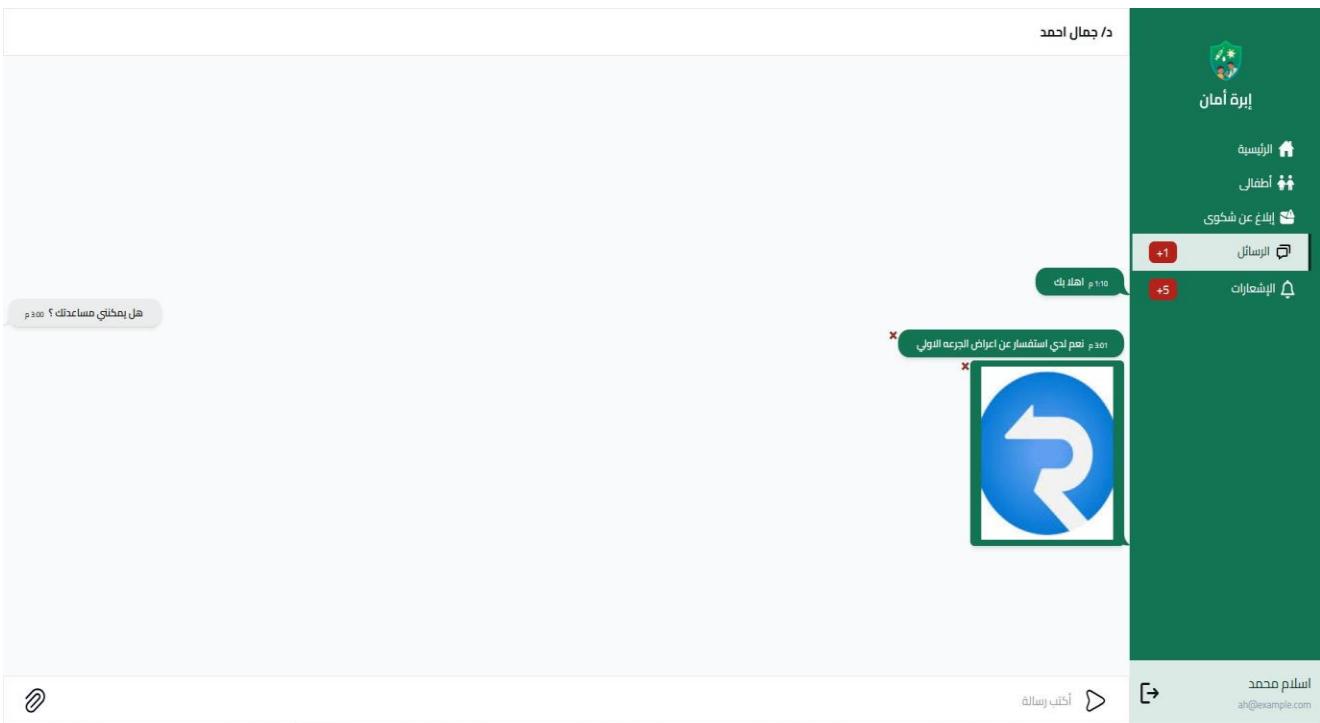


Figure 5.9.13 parent chat



Figure 5.9.14 parent notification

Keep parents informed about upcoming vaccination appointments and responses from doctors.

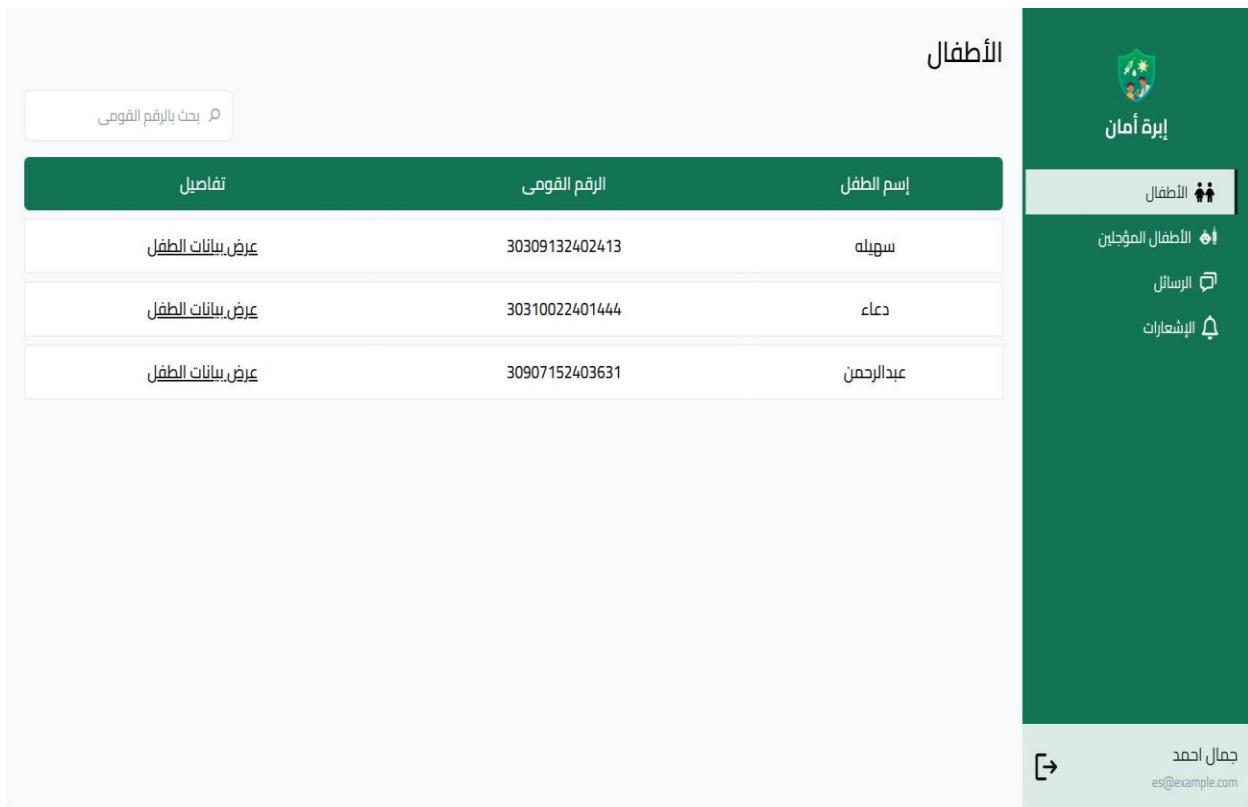


Figure 5.9.15 doctor home page

If the user was a doctor, he would be directed to the Home Page, the home page is divided into a navbar that includes links to the following pages:

- "الأطفال"
- "الرسائل"

This navbar will appear in all pages of the portal

"الأطفال" page Provide doctor with an overview of registered children and their medical histories. and can click on button “إنشاء جدول” to create vaccine schedule

تفاصيل	الرقم القومي	إسم الطفل
عرض بيانات الطفل	30309132402413	سويله سعيد خالد
عرض بيانات الطفل	30907152403631	عبدالرحمن اسلام محمد

Figure 5.9.16 doctor suspended children

This screen displays a list of children whose vaccinations were previously deferred by the doctor due to medical reasons (e.g., illness, low immunity).

It helps the doctor keep track of deferred cases and review or update their vaccination status when appropriate

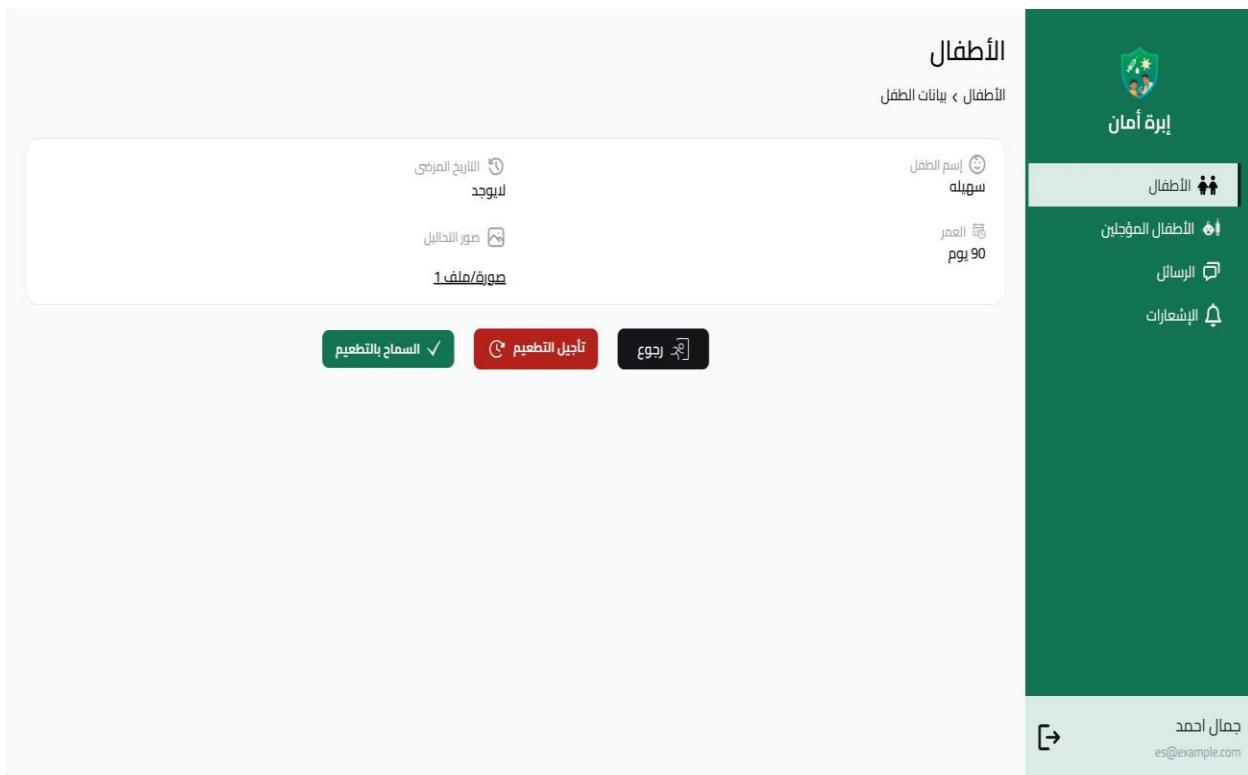


Figure 5.9.17 doctor child disease details

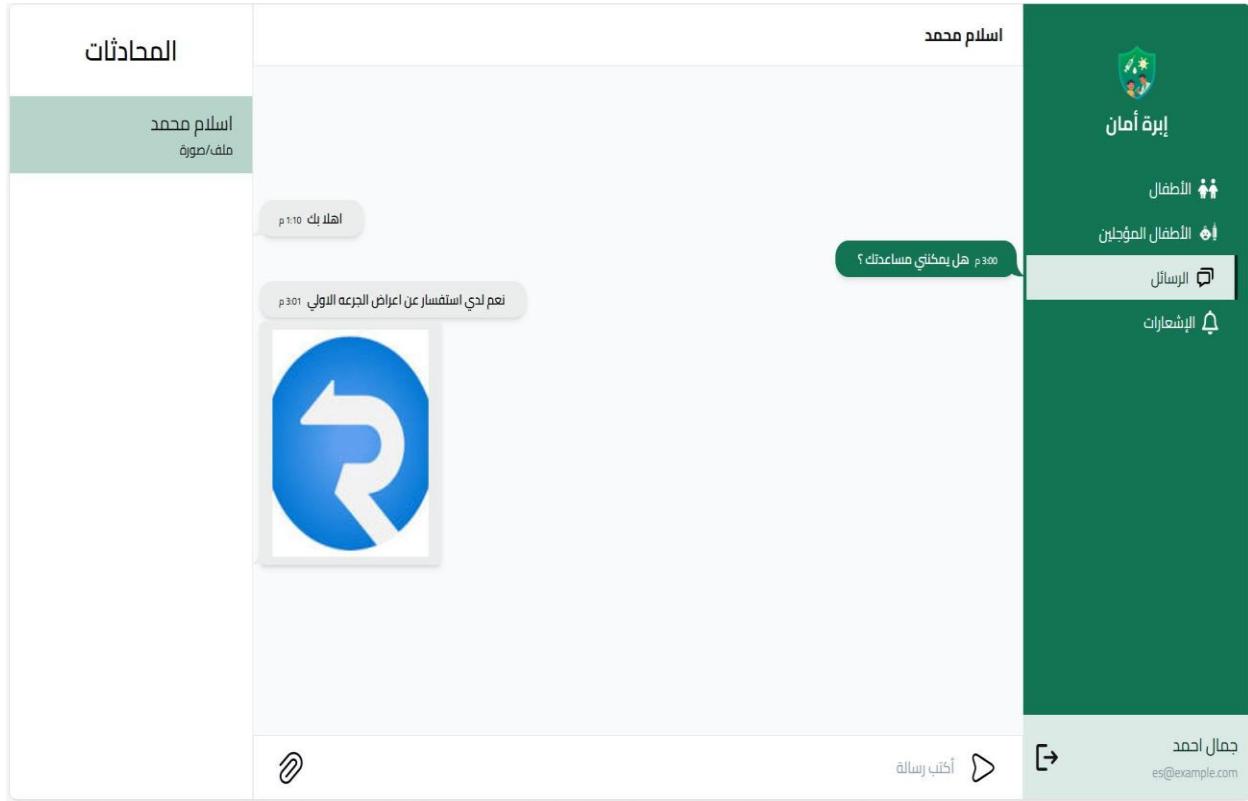


Figure 5.9.18 doctor chat

The screenshot shows the 'Organizer' home page. At the top right, there is a logo for 'إربة أمان' (Eربة أمان) with the text 'الأطفال' (Children). Below the logo, there are two buttons: 'الأطفال الحاصلة على اللقاح' (Children vaccinated) and 'الأطفال القادمون' (Coming children). A search bar at the top left contains the placeholder 'بحث بالرقم القومي' (Search by national ID). The main content area displays a table with four columns: 'الحالة' (Status), 'الجرعة' (Dose), 'الرقم القومي' (National ID), and 'إسم الطفل' (Child's name). The table contains four rows of data:

الحالة	الجرعة	الرقم القومي	إسم الطفل
تاجيل تم التطعيم	جرعة الثالثة	30309012404197	أيات اسلام محمد
تاجيل تم التطعيم	جرعة الثانية	30208052400958	جمال اسلام محمد
تاجيل تم التطعيم	جرعة الثانية	30309132402413	سونيله سعيد خالد
تاجيل تم التطعيم	جرعة الاولى	30310022401444	دعاء سعيد خالد

On the right side, there is a sidebar with a green header 'إربة أمان' and a green footer with the user information 'عبدالرحمن احمد' and 'or1@example.com'. The sidebar includes a profile picture, a 'الأطفال' button with a person icon, a 'اللقاحات' button with a syringe icon, a 'الطلبات' button with a document icon, and an 'إشعارات' button with a bell icon. There is also a red '+' button.

Figure 5.9.19 organizer home page

If the user was an Admin of healthcare, he would be directed to the Home Page, the home page is divided into a navbar that includes links to the following pages:

- "الأطفال القادمون"
- "اللقاحات"
- "الإحصائيات"

This navbar will appear in all pages of the portal

"الأطفال القادمون" page Provide admin the coming children of current day and can click on button "إعطاء اللقاح" to mark it for child as taken

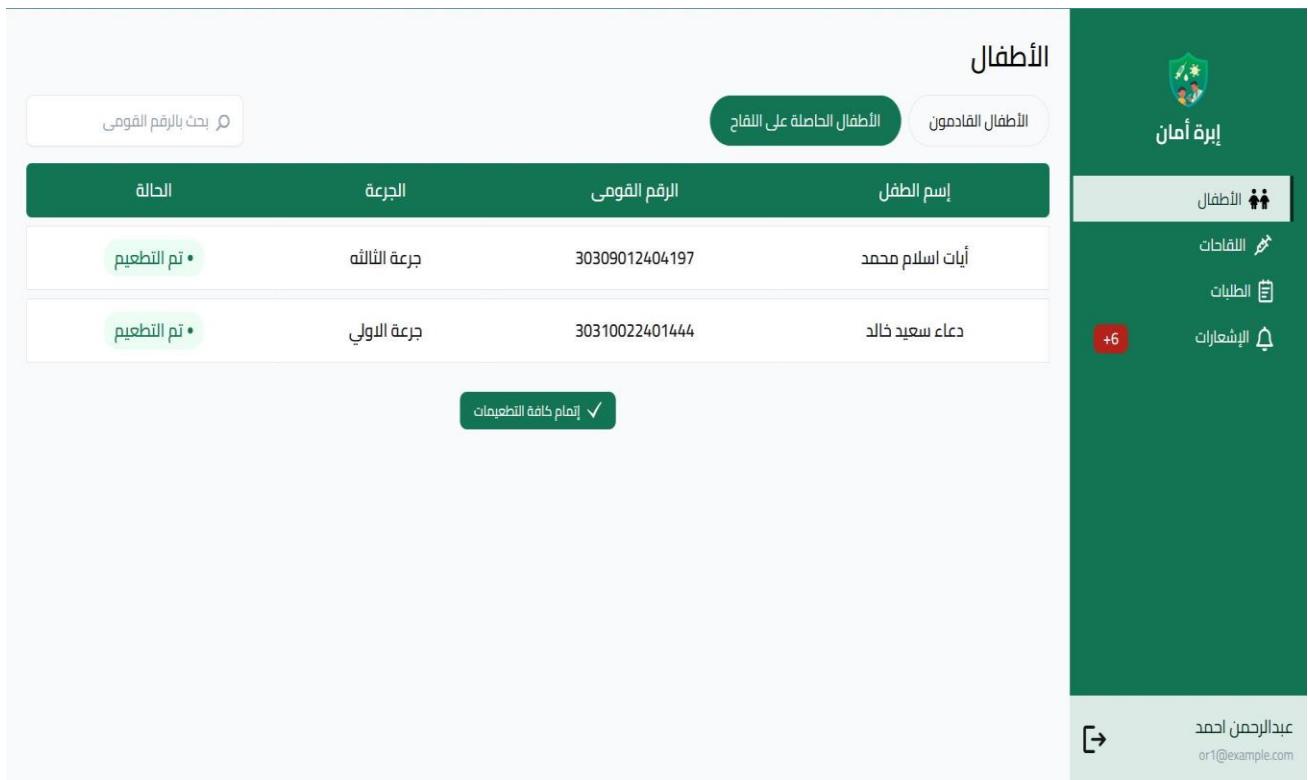


Figure 5.9.20 organizer child taked vaccines

This screen displays a list of all children who received their vaccinations today at the health unit. It allows the Organizer to keep an accurate daily record of vaccinations administered, which can be used for tracking, reporting, or follow-up purposes.



Figure 5.9.21 about website



Figure 5.9.21 Admin of Ministry of Health governorate

If the user was an Admin of Ministry of Health, he would be directed to the Home Page, the home page is divided into a navbar that includes links to the following pages:

- "الوحدات الصحية"
- "الشكاوى"

This navbar will appear in all pages of the portal

Admin of Ministry of Health can see all healthy cares belong to the ministry of health and know information about health care.

And knows the inventory amount and can send package of vaccines to any health care has shortage in vaccine amount

The screenshot displays three main sections of the dashboard:

- Left Panel (محافظة المنيا):**
 - Section title: محافظة المنيا
 - User info: المسئول (أسلام محمد), البريد الإلكتروني (doc@example.com)
 - Counters: عدد المراكز (3), عدد الوحدات الصحية (4)
 - Table: مخزون اللقاحات (Vaccine Inventory)

العدد	إسم اللقاح
1800	الثاني البكتيري
1748	الخماسي
2100	ام ام ار
2000	بي سي جي
2073	سابن
2068	سولك
2160	كدي ب رفع
- Middle Panel (مراكز محافظة المنيا):**
 - Section title: مراكز محافظة المنيا
 - Sub-section: الصفحة الرئيسية > محافظة المنيا
 - List of centers: مركز مطاي, مركز بي مزار, مركز المنيا
 - Buttons: [رجوع] (Back), [إضافة] (Add)
- Right Panel (إربة أمان):**
 - Header: إربة أمان
 - Navigation: لوحة التحكم, المحافظات, المسؤولون, الطلبات, الإشعارات (+9)
 - Footer: وزارة الصحة, da210509@gmail.com

Figure 5.9.22 Admin of Ministry of Health cites

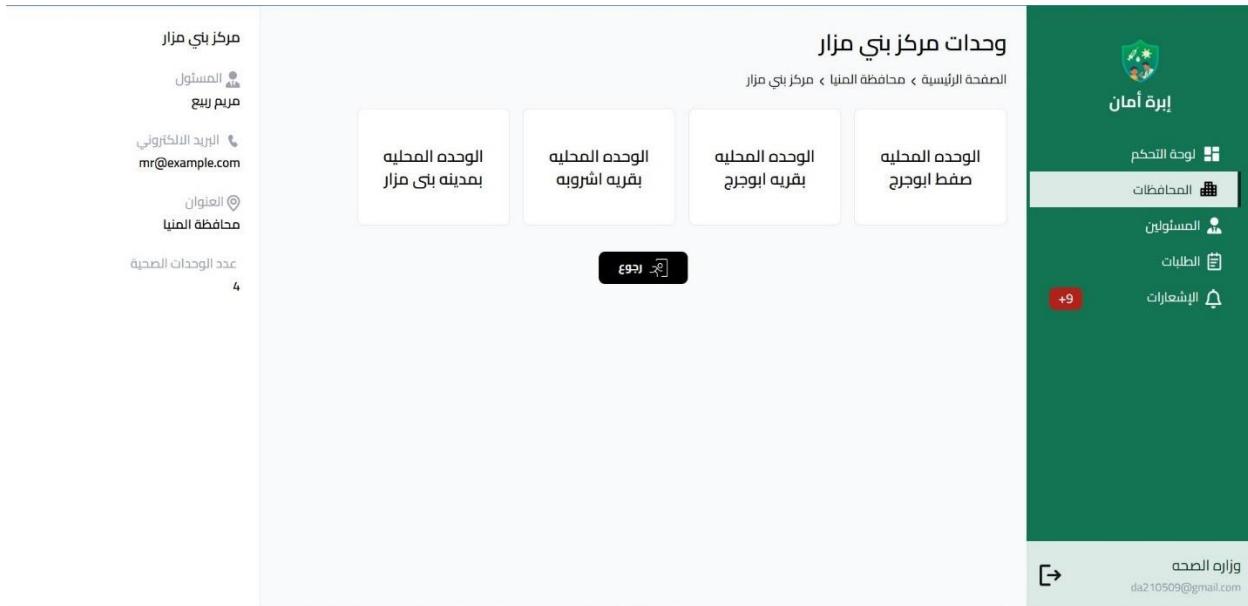


Figure 5.9.23 Admin of Ministry of Health health cares

إبراہ أمان

- لوحة التحكم**
- الوحدات**
- المسؤولون**
- الطلبات**
- اللقاءات**
- الطلبات**
- الشكاوى**
- الإشعارات**

+6

وحدات مركز العنيا

الصفحة الرئيسية > الوحدة المختصة بفرعه ابوجرج

معلومات الوحدة المختصة بفرعه ابوجرج

	العنوان
	محافظة العنيا - مركز نبي مزار - ابوجرج
	العنوان
	الحادي عشر - الاربعاء
	العنوان
	عبدالرحمن احمد
	العنوان
	جمال احمد

العدد	اسم اللقاء
450	الثلاثي البكتيري
750	الخامسي
300	ام ام ار
450	ابي سي دي
375	سايون
250	سولك
250	كريدي ب رضع

إضافة
تعديل

[→] مریم معن
mr@example.com

Figure 5.9.24 health care unites details

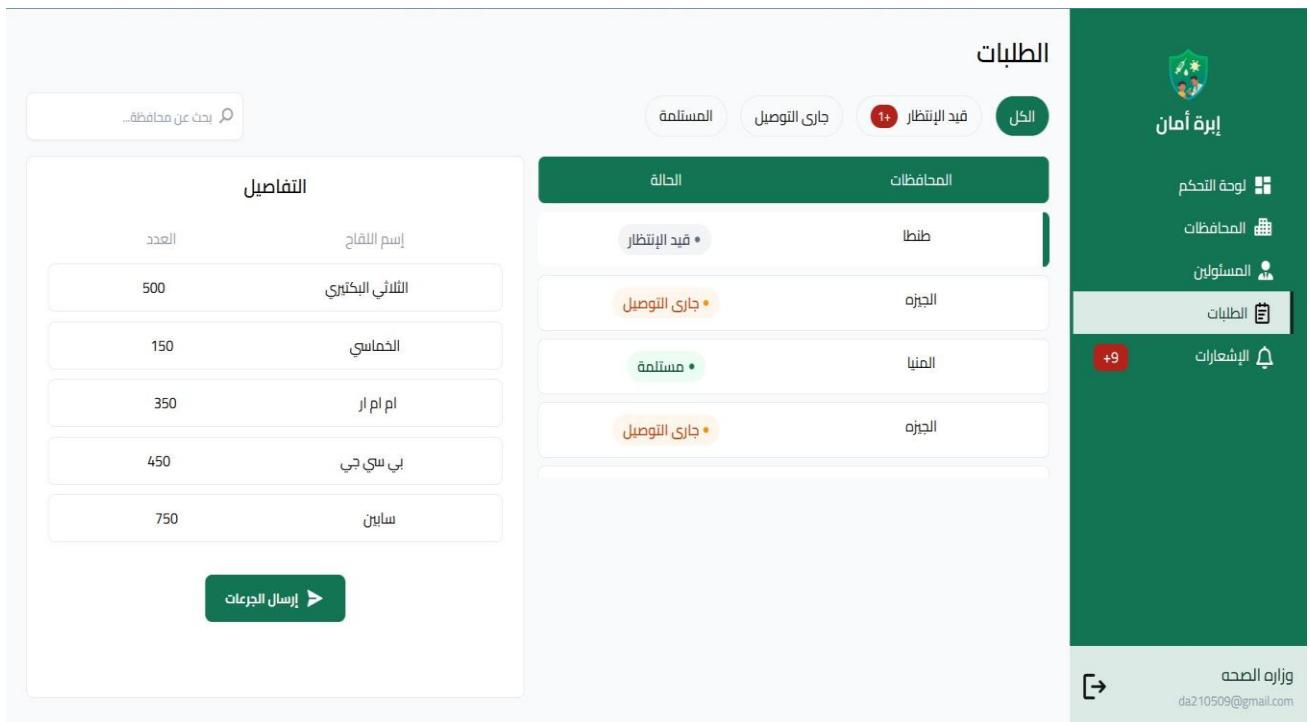


Figure 5.9.25 Admin of Ministry of Health governorate orders

مسؤول الوحدات

إضافة مسؤول وحدة... ٥

الإجراءات	البريد الإلكتروني	الوحدة الصنفية	الإسم
	or1@example.com	الوحدة المحلية بقرية ابوذرخ	عبدالرحمن احمد
	or2@example.com	الوحدة المحلية صفط ابوذرخ	محمود علاء
	mo@gmail.com	الوحدة المحلية بقرية اشجوبه	محمد ناصر
	mn@gmail.com	الوحدة المحلية بمدينة بني مزار	منه ربيع

لوحة التحكم

إدراة أمان

لوحة التحكم

الوحدات

المشروعات

الطلبات

الطلبات

الشكاوى

الإشعارات

+6

مريم ربيع
mir@example.com

Figure 5.9.26 organizer list

الأطباء

إضافة طبيب

بحث عن طبيب

الإجراءات	البريد الإلكتروني	الوحدة الصديقة	الاسم
	es@example.com	الوحدة المحلية بقرية ابوصرخ	جمال احمد
	ah@gmail.com	الوحدة المحلية بقرية اشروعه	احمد اسماعيل
	dr2@example.com	الوحدة المحلية صفت ابوصرخ	محمد خالد
	hn@gmail.com	الوحدة المحلية بعدهنه بني مزار	هناء رسن

+6

إبرة أمان

لوحة التحكم

الوحدات

المصليون

الطلبات

اللقاءات

الشكوى

الإشعارات

مريم رسن
mr@example.com

Figure 5.9.27 doctor list

اللقاحات

إسم اللقاح	عدد الجرعات
الثلاثي البكتيري	150
المهاسي	250
أم ام ار	500
اب سي جي	482
سابين	350
سولك	500
كدي ب رضع	500

[إبلاغ عن نقص](#)

+6
مردم رفع
mr@example.com

Figure 5.9.28 inventory page

vaccines screen that provide the admin with the knowledge of inventory quantity for each vaccine and enable to request more vaccines if any amount about to end by clicking “ارسال انذار” button.

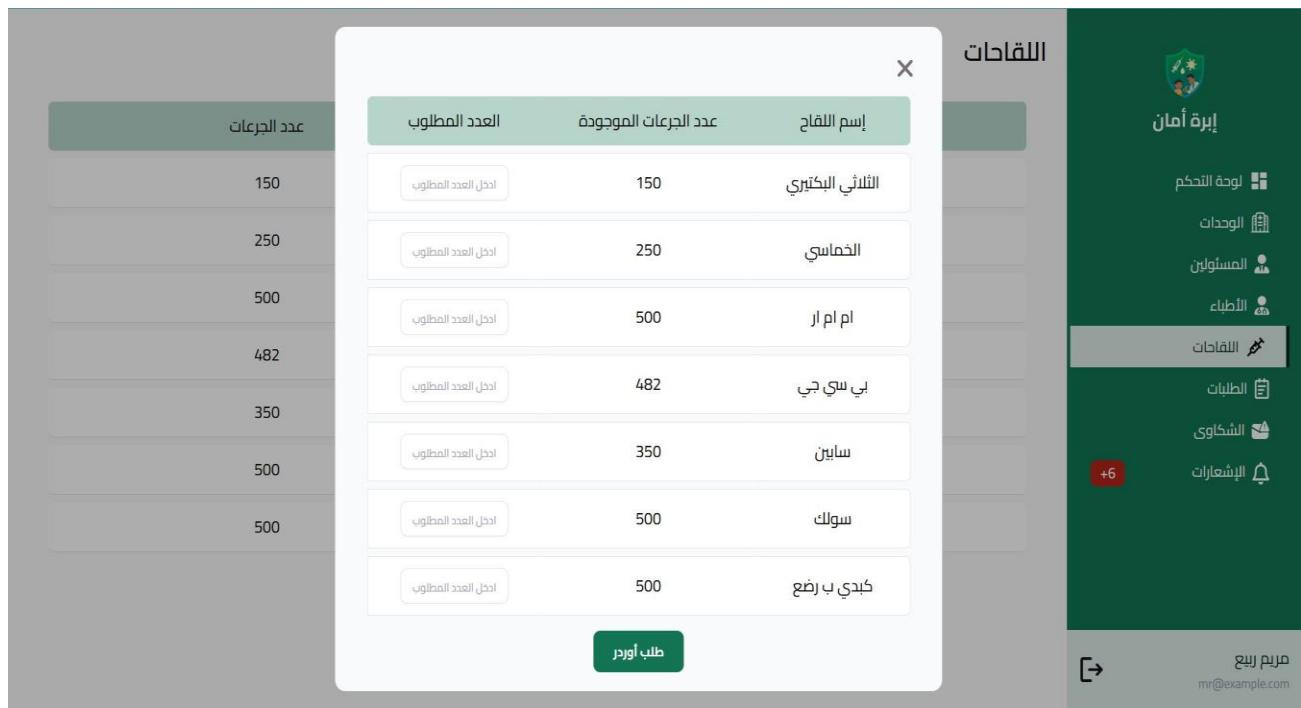


Figure 5.9.29 inventory shortage request

This screen allows the user to view a list of available vaccines and specify the quantity needed for each one.

It is used when placing an order request for vaccines, ensuring that the right types and amounts are requested based on current needs.

The screenshot shows the 'Coming Orders' section of the 'Ibera Amman' mobile application. At the top, there's a search bar with placeholder text 'ابحث عن وحدة...' and a dropdown menu labeled 'كل الطلبات' (All Orders). Below this, there are two main sections: 'التفاصيل' (Details) on the left and 'الطلبات' (Orders) on the right.

التفاصيل (Details):

الوحدة	إسم المخزن
500	الثلاثي الثاني
750	الخماسي
300	اب سي دي
500	سابون
500	سولوك
18	كبدى ب رفع

الطلبات (Orders):

الحالة	الوحدة المصدرة
مستلمة *	الوحدة المصدرة بمفرده اشترى وحدة
قيد الإنتظار *	الوحدة المصدرة بمفرده اشترى وحدة
مستلمة *	الوحدة المصدرة صفت ابوجرج
جارى التوصيل *	الوحدة المصدرة بمفرده ابوجرج

On the right side of the screen, there's a sidebar titled 'إربة أمان' (Ibera Amman) with various navigation options: لوحة التحكم (Control Panel), الوحدات (Units), المسؤولون (Managers), الأطقم (Teams), الناقلات (Transporters), الطلبات (Orders), الشكاوى (Complaints), and الإشعارات (Notifications). A red notification badge with the number '+6' is visible on the 'الطلبات' icon. At the bottom right, there's a user profile section for 'مريم ربيع' (Meriem Rabea) with the email 'mr@example.com'.

Figure 5.9.30 coming orders details

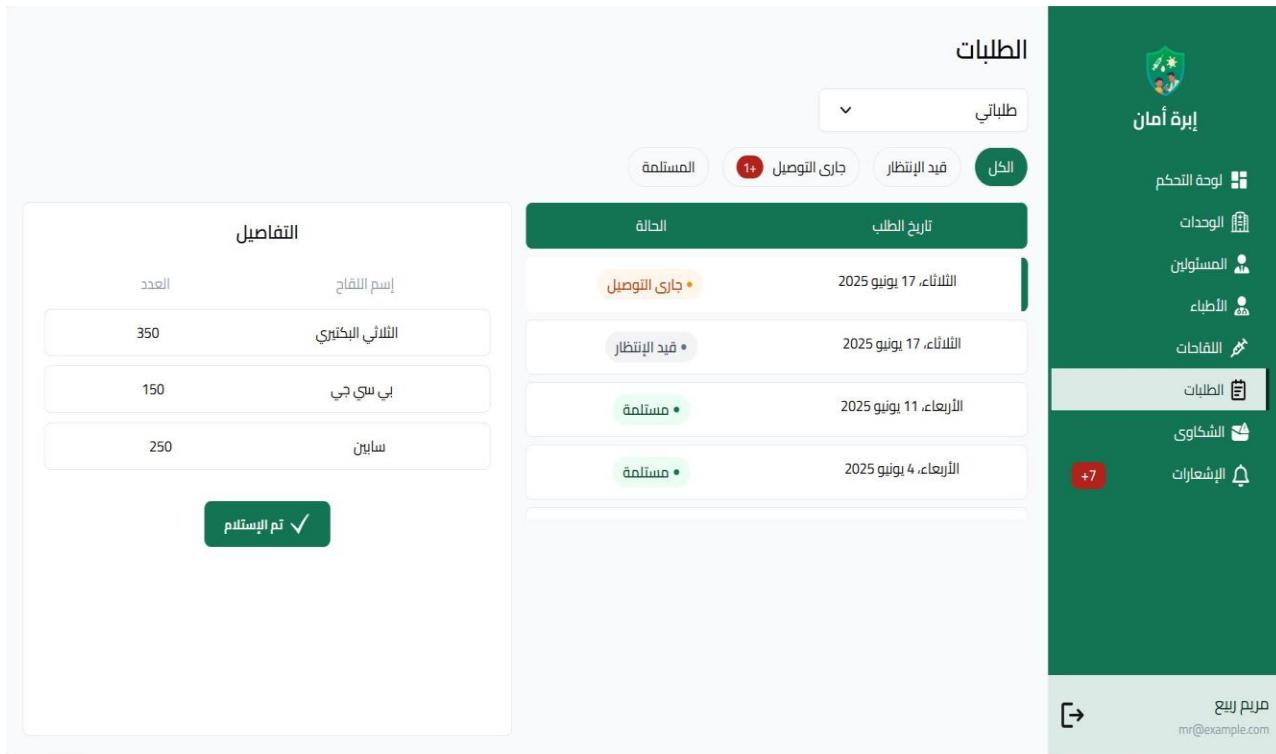


Figure 5.9.31 my orders details

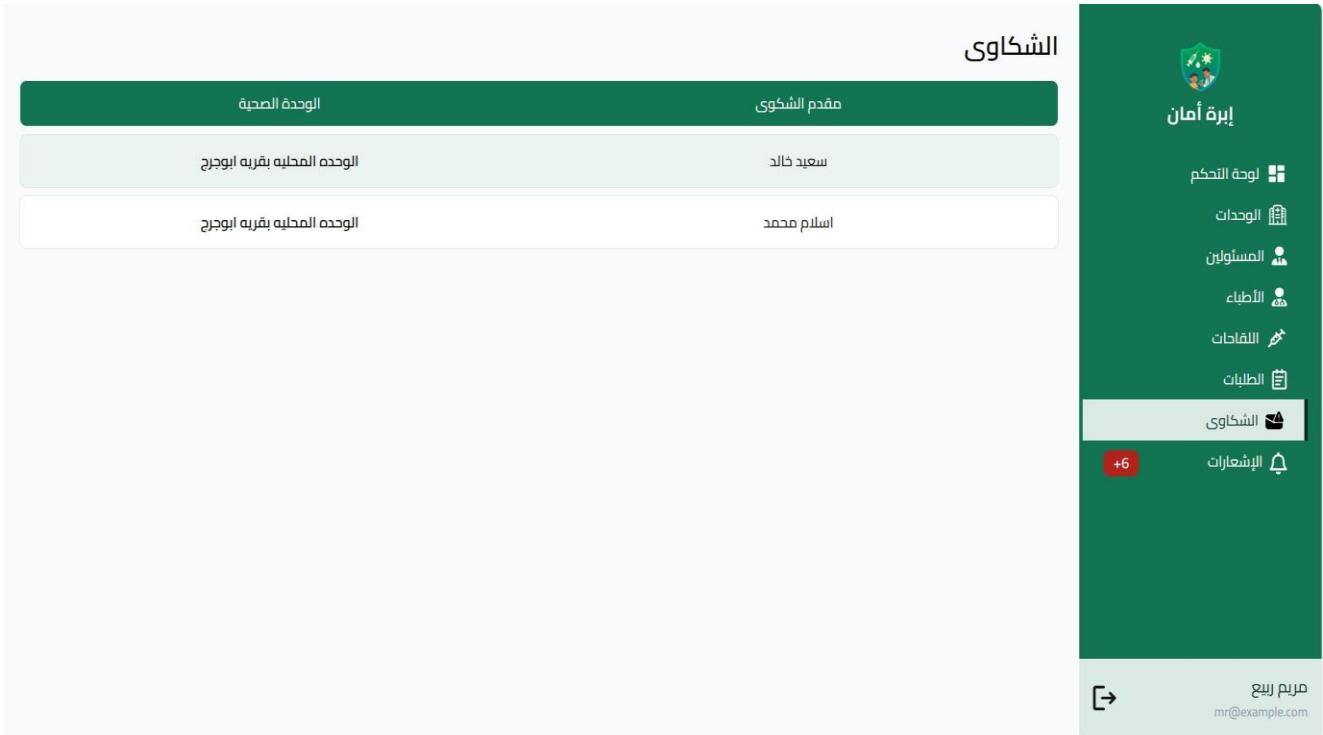


Figure 5.9.32 requested complaint page

Admin can see the requested complaint from parents about any something wrong or any disrespectful issues that happened to him in health care or any failing in contacting with doctor

تفاصيل الشكوى

البريد الإلكتروني الخاص بمسئولي الوحدة	العنوان	اسم مقدم الشكوى
mr@example.com	محافظة المنيا - مركز ناي مزار	اسلام محمد
عدم الالتزام بمواعيد عمل الوحدة الصديقة		

▶ تبليغ مقدم الشكوى بحل المشكلة

إبرة أمان

وحدة التدكم

الوحدات

المسؤولون

الطلبات

اللقاءات

الطلبات

الشكوى

+6 إشعارات

مريم ربيع

mr@example.com

Figure 5.9.33 requested complaint details

Implementation

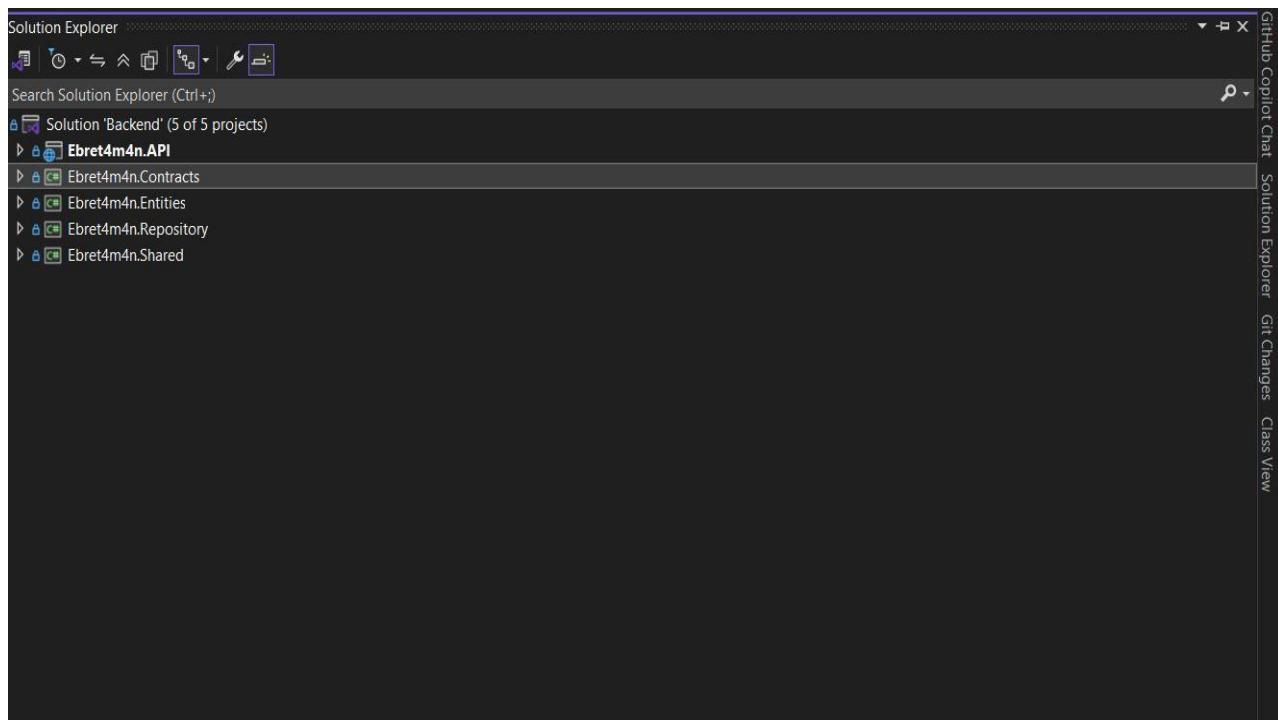


Figure 6 project structure

```

[HttpPost(template: "register")]
public async Task<IActionResult> Register([FromBody] RegisterDto model)
{
    if (!ModelState.IsValid)
        return UnprocessableEntity(GeneralResponse<object>.FailureResponse(ModelState));

    var user = model.Adapt<ApplicationUser>();

    var result = await userManager.CreateAsync(user, model.Password);

    if (!result.Succeeded)
        return BadRequest(GeneralResponse<List<string>>.
FailureResponse(["لم يتم إنشاء مستخدم新账号，因为输入的密码不正确."]));
    " يجب أن تكون كلمة المرور على الأقل من 8 أرقام وحروف انجلزية مغيرة وكبيرة.";

    await userManager.AddToRoleAsync(user, model.Role);

    var userDto = user.Adapt<UserDataDto>();

    var response = GeneralResponse<UserDataDto>.SuccessResponse(userDto);

    return Ok(response);
}

```

Figure 6.1 registration code

```
[HttpPost(template: "Login")]
public async Task<ActionResult> Login([FromBody] LoginDto model)
{
    if (!ModelState.IsValid)
        return UnprocessableEntity(
            generalResponse<string>.FailureResponse(message: "الرجاء إدخال البرور وكلمه المرور"));

    _user = await userManager.FindByEmailAsync(model.Email);

    if (_user is null)
        return BadRequest(generalResponse<string>.FailureResponse(message: "لم يتم العثور على المستخدم"));

    bool checkPassword = await userManager.CheckPasswordAsync(_user, model.Password);

    if (await userManager.IsLockedOutAsync(_user))
        return BadRequest(generalResponse<string>.FailureResponse(message: "تم حظر هذا الحساب من الدخول الرجاء المحاولة لاحقاً"));

    if (!checkPassword)
    {
        await userManager.AccessFailedAsync(_user);
        return BadRequest(generalResponse<string>.FailureResponse(message: "الرجاء إدخال البرور وكلمه المرور"));
    }

    await userManager.ResetAccessFailedCountAsync(_user);

    var token = await GenerateToken(populateExp: true);

    var response = GeneralResponse<TokenDto>.SuccessResponse(token);

    return Ok(response);
}
```

Figure 6.2 login code

```

[HttpPost(template: "forget-password")]
public async Task<IActionResult> ForgetPassword([FromBody] ForgotPasswordDto model)
{
    if (!ModelState.IsValid)
        return UnprocessableEntity(
            GeneralResponse<object>.FailureResponse(ModelState));
    _user = await userManager.FindByEmailAsync(model.Email);

    if (_user is null)
        return Ok(GeneralResponse<string>.SuccessResponse(data: "سوف يتم ارسال رساله علي الجيميل اذا كنت مسجل به"));

    var token = await userManager.GeneratePasswordResetTokenAsync(_user);

    var callbackUrl = $"http://localhost:4200/auth/change-password?userId={_user.Id}&token={Uri.EscapeDataString(token)}";

    string email = model.Email;
    string subject = "اعاده تعيين كلمه المرور";
    string body = await GenerateEmailMessage(callbackUrl,
        emailTitle: "اعاده تعيين كلمه المرور",
        emailBody: "تم طلب اعاده تعيين كلمه مرور جديد");

    await emailSender.SendEmailAsync(email, subject, body);

    var response = GeneralResponse<string>.SuccessResponse(data: "سيتم ارسال رساله اعاده تعيين كلمه سر لهذا البريد الالكتروني");
    return Ok(response);
}

```

Figure 6.3 forgetPassword code

```

[HttpPost(template: "reset-password")]
public async Task<IActionResult> ResetPassword([FromBody] ResetPasswordDto model)
{
    if(!ModelState.IsValid)
        return UnprocessableEntity(
            GeneralResponse<object>.FailureResponse(ModelState));

    _user = await userManager.FindByIdAsync(model.UserId);

    if (_user is null)
        return NotFound(GeneralResponse<string>.FailureResponse(message: $"لا يوجد مستخدم بهذا الرقم {_user.Id}"));

    string decodedToken = model.Token;

    string newPassword = model.NewPassword;

    var result = await userManager.ResetPasswordAsync(_user, decodedToken, newPassword);

    if(!result.Succeeded)
        throw new InValidTokenBadRequest();

    var response = GeneralResponse<string>.SuccessResponse(data: "تم اعاده تعيين كلمه المرور بنجاح");
    return Ok(response);
}

```

Figure 6.4 resetPassword code

```

[HttpGet(template: "admin")]
[Authorize(Roles = "admin")]
public async Task<ActionResult<AdminDto>> GetAdminStats()
{
    try
    {
        var children = await _unitOfWork.ChildRepo
            .FindAll(trackChanges: false, includes: "Vaccines", "User", "User.HealthCareCenter")
            .ToListAsync();

        var vaccines = await _unitOfWork.VaccineRepo
            .FindAll(trackChanges: false, includes: "Child.User", "Child.User.HealthCareCenter")
            .ToListAsync();

        var complaints = await _unitOfWork.ComplaintRepo
            .FindAll(trackChanges: false, includes: "User", "User.HealthCareCenter")
            .CountAsync();

        var healthUnits = await _unitOfWork.HealthCareCenterRepo
            .FindAll(trackChanges: false)
            .CountAsync();

        var orders = await _unitOfWork.OrderItemRepo
            .FindAll(trackChanges: false, includes: "Order", "Order.GovernorateAdminStaff")
            .ToListAsync();

        var (maleCount, femaleCount, malePercentage, femalePercentage, fullyVaccinated, vaccinesTaken) = await GetBaseStats(children, vaccines);

        var topGovs = orders
            .Where(o => o.Order?.GovernorateAdminStaff != null)
            .GroupBy(o => o.Order.GovernorateAdminStaff!.Governorate)
            .Select(g => new GovernorateVaccineRequestDto
            {
                Governorate = g.Key,
                TotalVaccinesRequested = g.Sum(x => (int)x.Amount)
            })
            .OrderByDescending(x => x.TotalVaccinesRequested)
            .Take(count: 10)
            .ToList();
    }
}

```

```

foreach (var gov in govAdmins)
{
    var healthUnitCount = await _unitOfWork.HealthCareCenterRepo
        .FindByCondition(h => h.Governorate == gov.Governorate, trackChanges: false)
        .CountAsync();

    var vaccineCount = await _unitOfWork.VaccineRepo
        .FindByCondition(v => v.IsTaken && v.Child.User.Governorate == gov.Governorate, trackChanges: false, includes: "Child.User")
        .CountAsync();

    govReports.Add(item: new GovernorateReportDto
    {
        Governorate = gov.Governorate,
        CityCount = gov.CityAdminStaffs.Select(c => c.City).Distinct().Count(),
        HealthUnitCount = healthUnitCount,
        VaccinesTaken = vaccineCount
    });
}

var vaccineRequests = orders
    .GroupBy(o => o.Antigen)
    .Select(g => new VaccineRequestDto
    {
        VaccineName = g.Key,
        RequestedAmount = g.Sum(x => (int)x.Amount)
    })
    .OrderByDescending(x => x.RequestedAmount)
    .ToList();

```

return Ok(values: new AdminDto
{
 HealthCareUnits = healthUnits,
 RegisteredChildren = children.Count,
 FullyVaccinatedChildren = fullyVaccinated,
 TotalVaccinesTaken = vaccinesTaken,
 TotalComplaints = complaints,
 MaleChildren = maleCount,
 FemaleChildren = femaleCount,
 MalePercentage = malePercentage,
 FemalePercentage = femalePercentage,
 TopGovernoratesByVaccines = topGovs,
 GovernoratesReport = govReports,
 VaccineRequests = vaccineRequests
});

Figure 6.5 statistics code

```

[HttpGet(template: "child-base-vaccines")]
public IActionResult GetChildVaccines()
{
    try
    {
        var vaccines = Utility.ReadBaseVaccineFromJson();
        var response = GeneralResponse<List<BaseVaccine>>.SuccessResponse(vaccines);
        return Ok(response);
    }
    catch(Exception ex)
    {
        return BadRequest(GeneralResponse<string>.FailureResponse(ex.Message));
    }
}

```

Figure 6.6 add child code

```

[HttpPost(template: "child-add")]
public async Task<IActionResult> AddChild([FromForm]AddChildDto dto)
{
    if (!ModelState.IsValid)
        return UnprocessableEntity(
            GeneralResponse<object>.FailureResponse(ModelState));

    if (dto.BirthDate > DateTime.UtcNow)
        return BadRequest(GeneralResponse<string>.FailureResponse(message: "لا يمكن اختيار هذا التاريخ اكبر من التاريخ الحالي"));

    var parentId = User.FindFirst(type: "id")!.Value;

    var checkUniqNameIdentifier =
        await unitOfWork.ChildRepo.FindAsync(c => c.Id == dto.Id, trackChanges: false);

    if (checkUniqNameIdentifier != null)
        return BadRequest(GeneralResponse<string>.FailureResponse(message: "من فضلك ادخل رقم الطفل القومى المصحح"));

    var child = (dto, parentId).Adapt<Child>();

    if(dto.ReportFiles == null && dto.PatientHistory == null)
    {
        var vaccines = Utility.ChildVaccines(dto.TakedVaccines, child.Id);
        await unitOfWork.VaccineRepo.AddRangeAsync(vaccines);
    }

    if (dto.ReportFiles != null)
    {
        var childReports = Utility.SaveReportFiles(dto.ReportFiles, dto.Id);
        await unitOfWork.HealthyReportRepo.AddRangeAsync(childReports);
    }

    await unitOfWork.ChildRepo.AddAsync(child);

    var result = await unitOfWork.SaveAsync();

    if (result == 0)
        return BadRequest(GeneralResponse<string>.FailureResponse(message: "لم يتم حفظ الطفل الرجاء المحاولة مرة اخرى"));

    var response = GeneralResponse<string>.SuccessResponse(data: "تم اضافة الطفل بنجاح");
    return Ok(response);
}

```

Figure 6.7 add child code

```

[HttpGet(template: "{id:guid}/healthCareCenter")]
[Authorize(Roles = "cityAdmin,governorateAdmin,admin")]
public async Task<ActionResult> HealthCareDetails(Guid id)
{
    var healthCareCenter = await unitOfWork.Work.HealthCareCenterRepo
        .FindAsync(hc => hc.HealthCareCenterId == id, trackChanges: false, ["Inventories"]);

    if (healthCareCenter is null)
        return NotFound(GenericResponse<string>.FailureResponse(message: "لم يتم العثور على المنشأة"));

    var healthCareStaff = unitOfWork.Work.StaffRepo.FindByCondition(staff => staff.HCCenterId == healthCareCenter.HealthCareCenterId, trackChanges: false, ["User"])
        .ToList();

    var organizerName =
        healthCareStaff.Where(hc => hc.StaffRole == StaffRole.Organizer)
        .Select(org => $"{org.User.FirstName} {org.User.LastName}")
        .FirstOrDefault();

    var doctorName = healthCareStaff.Where(hc => hc.StaffRole == StaffRole.Doctor)
        .Select(doc => $"{doc.User.FirstName} {doc.User.LastName}")
        .FirstOrDefault();

    var healthCareDetails = healthCareCenter.Adapt<HealthCareDetailsDto>();
    healthCareDetails.OrganizerName = organizerName;
    healthCareDetails.DoctorName = doctorName;
    healthCareDetails.Inventories = healthCareCenter.Inventories.Adapt<List<InventoryDto>>();

    var response = GeneralResponse<HealthCareDetailsDto>.SuccessResponse(healthCareDetails);
    return Ok(response);
}

```

Figure 6.8 HealthCare details code

```

[HttpGet(template: "children-disease")]
public IActionResult GetChildrenWithDisease()
{
    var doctorHcCenterId = User.FindFirst(type: "healthCareId")!.Value;

    var children = unitOfWork.ChildRepo.FindByCondition(
        c => (c.PatientHistory != null || (c.HealthReportFiles != null && c.HealthReportFiles.Any()) &&
        (c.IsNormal != false && (c.Vaccines == null || c.Vaccines.Count == 0) ) &&
        c.User.HealthCareCenterId.ToString() == doctorHcCenterId,
        trackChanges: false, ["User", "Vaccines", "HealthReportFiles"]).ToList();

    var childrenDto = children.Adapt<List<ChildDto>>();
    var response = GeneralResponse<List<ChildDto>>.SuccessResponse(childrenDto);

    return Ok(response);
}

```

Figure 6.9 HealthCare details code

```

[HttpPost(template: "{childId}/suspend")]
public async Task<IActionResult> SuspendChildVaccine(string childId)
{
    var child =
        await unitOfWork.ChildRepo.FindAsync(child => child.Id == childId, trackChanges: false, ["User"]);

    if (child is null)
        return NotFound(GeneralResponse<string>.FailureResponse(message: "الطفل غير موجود"));

    if (child.IsNormal == false)
        return BadRequest(GeneralResponse<string>.FailureResponse(message: "تم افافه الطفل الى قائمته المؤجلين من قبل"));

    child.IsNormal = false;

    unitOfWork.ChildRepo.Update(child);
    int result = await unitOfWork.SaveChangesAsync();

    if (result == 0)
        return BadRequest(GeneralResponse<string>.FailureResponse(message: "لم يتم تأجيل اللقاحات لهذا الطفل حاول مرة اخرى"));

    await emailSender
        .SendEmailAsync(child.User.Email!,
        subject: "بناء على التحاليل المقدمة تم تأجيل التطعيم لطفلك " + child.Name, htmlMessage: $"<p>تأجيل التطعيمات</p>");

    var response = GeneralResponse<string>.SuccessResponse(data: "تم تأجيل اللقاحات لهذا الطفل");
    return Ok(response);
}

```

Figure 6.10 supspend child code

```

[HttpPost(template: "{childId}/process-payment")]
public async Task<IActionResult> ProcessPayment(string childId)
{
    var parentId = User.FindFirst(type: "id")!.Value;
    var child = await unitOfWork.ChildRepo.FindAsync(child => child.Id == childId, trackChanges: false, ["User"]);

    if (child is null)
        return BadRequest(GeneralResponse<string>.FailureResponse(message: "لا يوجد طفل مسجل بهذا الرقم"));

    var session = await Utility.CreateSessionPayment(parentId, child.User.Email!, childId, child.Name);

    var response = GeneralResponse<string>.SuccessResponse(session.Url);

    return Ok(response);
}

```

Figure 6.11 payment code

```

[HttpPost(template: "stripe")]
public async Task<IActionResult> StripeWebhook()
{
    var json = await new StreamReader(HttpContext.Request.Body).ReadToEndAsync();
    try
    {
        var stripeEvent =
            EventUtility.ConstructEvent(json,
                Request.Headers[key: "Stripe-Signature"],
                _stripConfig.WebhookSecret);

        if (stripeEvent.Type == "checkout.session.completed")
        {
            var session = stripeEvent.Data.Object as Session;

            if (session != null &&
                session.Mode == "payment" &&
                session.PaymentStatus == "paid")
            {
                var childId = session.Metadata[key: "childId"];
                var childName = session.Metadata[key: "childName"];
                var parentEmail = session.Metadata[key: "parentEmail"];

                var transaction = (session, session.ClientReferenceId, childId).Adapt<Transaction>();

                transaction.PaymentIntentId = session.PaymentIntentId;
                transaction.Paid = true;
                transaction.PaidAt = DateTime.UtcNow;

                await unitOfWork.TransactionRepo.AddAsync(transaction);

                var result = await unitOfWork.SaveAsync();

                if (result == 0)
                    return StatusCode(StatusCodes.Status500InternalServerError);

                await emailSender.SendEmailAsync(parentEmail, subject: "عملية دفع ناجحة لطفل", htmlMessage: $"عملية الدفع : {childName}</p>");
            }
        }
        catch (StripeException ex)
        {
            return BadRequest(GeneralResponse<string>.FailureResponse(message: $"stripe webhook error {ex.Message}"));
        }
    }
    return Ok();
}

```

Figure 6.12 webhook code

```

public static class ScheduleJob
{
    public static void NotificationMessagJob(this WebApplication app)
    {
        RecurringJob.AddOrUpdate<ReservationReminderService>(recurringJobId: "reservation-reminders",
            service => service.CheckReservationsAndSendNotificationsAsync(),
            Cron.Daily());

        RecurringJob.AddOrUpdate<VaccineReminderService>(recurringJobId: "vaccine-reminders",
            service => service.CheckVaccineRemindersAndSendNotificationsAsync(),
            Cron.Weekly(DayOfWeek.Tuesday));
    }
}

```

Figure 6.13 background jobs code

```

public async Task CheckVaccineRemindersAndSendNotificationsAsync()
{
    await unitOfWork.BeginTransactionAsync();
    try
    {
        var children = unitOfWork.ChildRepo.FindAll(trackChanges: false, ["User", "Vaccines"])
            .ToList() ?? [];

        if(children.Count == 0)
            return;

        foreach(var child in children)
        {
            if(child.Vaccines == null || child.Vaccines.Count == 0)
                continue;

            var vaccine = child.Vaccines.FirstOrDefault(vaccine => vaccine.ChildAge == child.AgeInMonth);

            if (vaccine == null || vaccine.IsTaken == true)
                continue;

            var message = $"{child.Name} تم فتح حجز اللقاحات لرجاء الحجز والتوجه للوحدة المصحبة التابع لها {vaccine.Name} لتفادي التأخير";
            var notification = Utility.CreateNotification(title: "تنبيه لاقتراب موعد التطعيم", message, child.UserId);

            await hubContext.Clients.User(child.UserId).SendAsync(method: "NotificationMessage", notification.Adapt<NotificationDto>());

            await unitOfWork.NotificationRepo.AddAsync(notification);
        }

        await unitOfWork.SaveAsync();
        await unitOfWork.CommitTransactionAsync();
    }
    catch (Exception)
    {
        await unitOfWork.RollbackTransactionAsync();
        throw;
    }
}

```

Figure 6.14 notificatoin vaccine reminder background jobs code

```

public async Task SendMessage(ChatMessageDto message)
{
    var chat = message.Adapt<Chat>();
    chat.SentAt = DateTime.UtcNow;

    if (message.File != null)
        chat.File = Utility.SaveBase64File(message.File);

    await unitOfWork.ChatRepo.AddAsync(chat);

    string preview;
    if (chat.Message != null)
        preview = chat.Message.Length > 50 ? $"{chat.Message.Substring(0, 45)}..." : chat.Message;
    else
        preview = "تم ارسال ملف جديد";

    var notification = Utility.CreateNotification("رسالة جديدة",
                                                preview,
                                                chat.ReceiverId);

    await unitOfWork.NotificationRepo.AddAsync(notification);

    await unitOfWork.SaveAsync();

    var chatDto = chat.Adapt<ChatMessageDetailsDto>();

    // Notify the NotificationHub about the new message

    var notificationDto = notification.Adapt<NotificationDto>();

    await hubContext.Clients.User(chat.ReceiverId)
                    .SendAsync("NotificationMessage", notificationDto);

    await Clients.Users([chatSenderId!, chat.ReceiverId!]).SendAsync("ReceiveMessage", chatDto);
}

```

Figure 6.15 send realtime message code

```

public async Task MarkMessagesAsRead(Guid senderId)
{
    var receiverId = Context.UserId;

    // Get the IDs first
    var readMessageIds = await unitOfWork.ChatRepo
        .FindByCondition(m => m.SenderId == senderId.ToString()
                           && m.ReceiverId == receiverId
                           && !m.IsRead, trackChanges: true)
        .Select(m => m.Id)
        .ToListAsync();

    if (readMessageIds.Count == 0)
        return;

    // Now update those by ID
    await unitOfWork.ChatRepo
        .FindByCondition(m => readMessageIds.Contains(m.Id), trackChanges: true)
        .ExecuteUpdateAsync(setter => setter
            .SetProperty(m => m.IsRead, valueExpression: true));

    await Clients.User(senderId.ToString()).SendAsync(method: "MessagesMarkedAsRead" , new
    {
        ReceiverId = receiverId,
        MessageIds = readMessageIds
    });
}

```

Figure 6.16 mark message as read code

```
public async Task DeleteMessage(Guid messageId)
{
    var chatMessage =
        await unitOfWork.ChatRepo.FindAsync(message => message.Id == messageId, trackChanges: true);

    if (chatMessage is null)
    {
        await Clients.Caller.SendAsync(method: "DeleteMessageError", arg1: "هذه الرساله غير موجوده");
        return;
    }

    if(chatMessage.SenderId != Context.UserId)
    {
        await Clients.Caller.SendAsync(method: "DeleteMessageError", arg1: "يمكنك حذف الرساله الخاصه بك");
        return;
    }

    unitOfWork.ChatRepo.Remove(chatMessage);

    await unitOfWork.SaveAsync();

    await Clients.Users([chatMessage.SenderId, chatMessage.ReceiverId])
        .SendAsync(method: "MessageDeleted", messageId);
}
```

Figure 6.17 delete message code

Conclusion

Conclusion

The Child Vaccination Management System successfully addresses the critical challenges in managing child vaccinations by offering a centralized, user-friendly platform. It enables parents to track their children's vaccination schedules, book appointments, and stay informed about vaccination-related information. Additionally, it assists healthcare providers in managing vaccine inventory and schedules while maintaining transparency and efficiency. By integrating modern technologies like ASP.NET Core, Angular, and Google Maps API, the system ensures accessibility, scalability, and reliability, creating a positive impact on public health management.

Difficulties Faced

While working on this project, we encountered the following challenges:

- Selecting a project subject/topic and finding a client.
- Analysing the current workflow.
- Finding information about such systems and the databases required.
- Working on the relationships between the entities.
- Managing time effectively.
- Designing a suitable interface.

Future Work

1. **Adult Vaccination Support:** Expand the system to include vaccination scheduling and management for adults, covering diseases like hepatitis and COVID-19.
2. **Mobile Application Development:** Create a mobile app version to enhance accessibility for parents and healthcare providers.
3. **AI Integration:** Incorporate AI for personalized vaccination recommendations and automated side effect analysis.
4. **Globalization and Localization:** Adapt the system to support multiple languages and regional vaccination guidelines.
5. **Advanced Reporting:** Develop more sophisticated dashboards for real-time statistics and reports for healthcare providers and government agencies.

References

[World Health Organization - Vaccination](#)

[Centres for Disease Control and Prevention \(CDC\) - Vaccine Schedules](#)