CPSC 471

Database Management Systems

MedDiary

University of Calgary

Instructor*: Reda Alhajj*

Date*: 04/04/2023*

By Alex Yeap, Aman Hossian, Anh Nguyen

Group 11

Abstract

Miscommunication between patients and doctors can have major repercussions and, in some cases, even be fatal. Paperwork can be difficult to manage, and a doctor’s handwriting can be difficult to read, making it possible for crucial medical data to be mistakenly interpreted. Patients may also have trouble keeping their medical records safe, for instance, if they move or lose them accidentally. Moreover, patients now do not have immediate access to their medical background, test outcomes, medication and diagnosis history.

Our focus is to make a web application that would allow patients and physicians to efficiently communicate and have online access to medication in order to address these issues. Patients will be able to read and manage their medication, to-do list, and clinic visit information. Doctors will be able to add, edit, and evaluate patient medical Information. This will improve communication and decrease the possibility of miscommunication by establishing a user-friendly interface and a online form of documentation which will be more convenient for both the doctors and patients. In the end, this web application will raise the standard of healthcare and most importantly improve communication.

An extended entity relationship diagram was created as part of the project's design process to ensure a thorough understanding of the project's numerous entities and their connections. This served as the basis for a relational model diagram that helped the execution of the design using the SQLite relational database management system. It was decided to construct API endpoints so that users of the program may add, remove, edit, and delete data in the database.

Table of Contents

CPSC 471 Final Report

Introduction

Miscommunication between doctors and patients is a critical problem that can have negative effects. It may result in medical mistakes, difficulties, or even fatalities. According to research, patients and doctors don't communicate well enough to account for over 80% of medical errors. Poor communication frequently results in patients receiving ambiguous instructions or failing to comprehend their diagnosis or treatment plan. This is particularly difficult when patients are obliged to take medications or adhere to a set treatment plan. Innovative solutions that can enhance communication between patients and doctors must be created in order to address these problems.

Accessing their medical records is one of the main difficulties that individuals encounter. For instance, in Alberta, patients must complete a form as they must follow a series of procedures in order to view their medical records. For individuals who need urgent medical care, this process can be very time-consuming and frustrating as they would have to go through the headache of all the processes. When patients need to check their medical records to validate unclear instructions or a questionable diagnosis, the issue may become even worse. As a result, creative solutions that provide quick and simple patients with access to their medical records are required.

We are creating a web application that would enable doctors to log and save patient prescriptions, health IDs, medication names, schedules, notes, diagnoses, follow-ups, future exams, and test results to address these problems. Both doctors and patients would have easy access to patients' information, which would assist in limiting misunderstandings and mistakes. This technology has the potential to significantly enhance the lives of many patients and make doctors' work easier by enhancing communication and making medical records more convenient to access.

Project Design

Users

The system developed includes 2 users:

- The patient user functionality in this web app is designed to provide an easy-to-use interface for patients to access their medical information. They can review their medical summary and prescriptions that were recorded by their doctor after the examination. This makes it easy for patients to follow their medical instructions in a clearly structured format. In addition, the app can access patients' lab tests or follow-ups that the doctor has requested. Patients can keep track of all treatments they have received in their personal accounts, which makes it easy for them to stay on top of their healthcare.
- The doctor functionality of this web app is designed to enable doctors to add or edit medical information for their patients. Doctors can record a patient's medical summary, notes, prescriptions, medication schedules, diagnoses, test results, and feedback. During the examination, doctors can input all the related data into the app. At the end of the session, they can add medicines, quantities, clear instructions for medicine intake, and special notes. Doctors can also declare if they require the patient to take any supplement tests or follow-up sessions, which will be notified in the patient's account. This app makes it easy for doctors to keep track of their patient's medical history, which ultimately leads to better patient care.

The system's API capabilities and comprehensive transaction collection can both be accessed in Appendix 5.

The system's Extended Entity-Relationship Model is presented in Appendix 1. Also included are a DFD model in Appendix 4, and a HIPO model in Appendix 3. The following modifications were made to the EERD since earlier iterations:

- Removed Lab Entity due to time constraint as it was not a big factor for our web app.
- Removed MedicalHistory Entity due to time constraint as it was optional add on.
- Added Insurance Entity for Patient;s Insurance Information.
- Removed Appointment and Walk_in Relationship type as we decided to have each Visit Information instead of having appointments or Walk_in recorded.
- Removed Diagnosis and added it to Clinic Visits instead.
- Added Attributes for Clinic Visit.

Implementation

The procedure for transforming an Entity Relationship Model diagram into a Relational Schema diagram was used to create a Relational Model diagram. You can find this relational model in Appendix 2.

During this process, important or uncommon decisions were taken, including:

- Added Entity Type Clinic Visit and its attributes.
- Removed Lab, Medical History, Immunization, Prescription, PRESCRIBE Appointment, Walk_in and Diagnosis Due to time Constraint as some were replaced with a better method like adding diagnosis to clinic visits.

SQLite was selected as the relational DBMS for this project. The software for this DBMS is open-source and free. Users are able to utilize the SQL language to communicate directly with implemented databases. Furthermore, it collaborates with other frameworks or programs to develop applications that need relational database capability. Entity Framework Core was the framework used in this project's implementation to connect to SQLite and give the web application access to relational database functions.

The SQL statements for each of the transactions implemented can be found in Appendix 5.

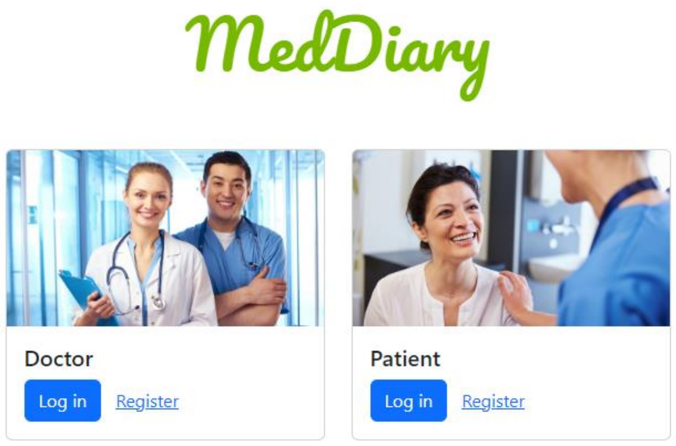
API documentation

The documentation of the API for this system was created through Swashbuckle, which is an OpenAPI implementation based on Swagger. The platform is implemented using .NET. Swagger is a specification that is independent of programming language and is used for describing REST APIs. More information about API documentation can be found in Appendix 6

User Guide

Registering User

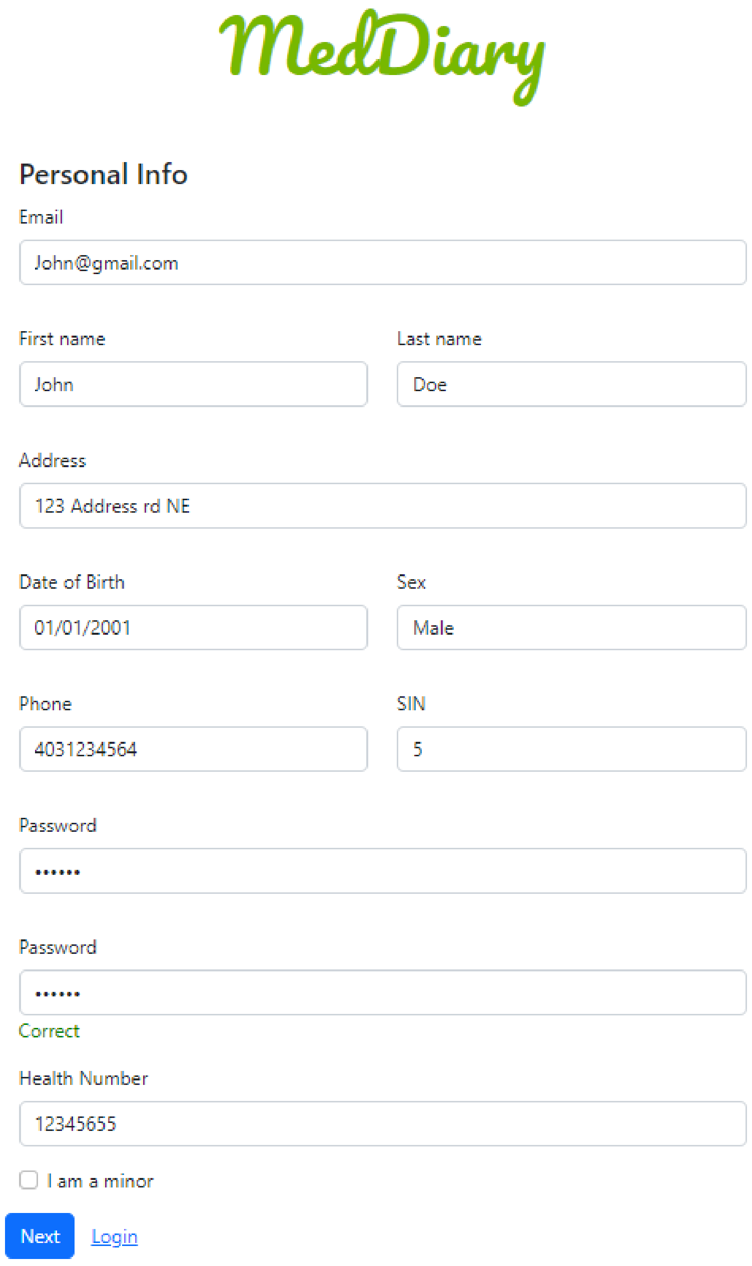
Step 1: Select Register for Doctor or Patient



Step 2: Fill in Personal Information and click Next

Note: Sin number and pracId (for doctor) or health number (for patient) needs to be unique.

Registered numbers for all ids: 1,2,3,4



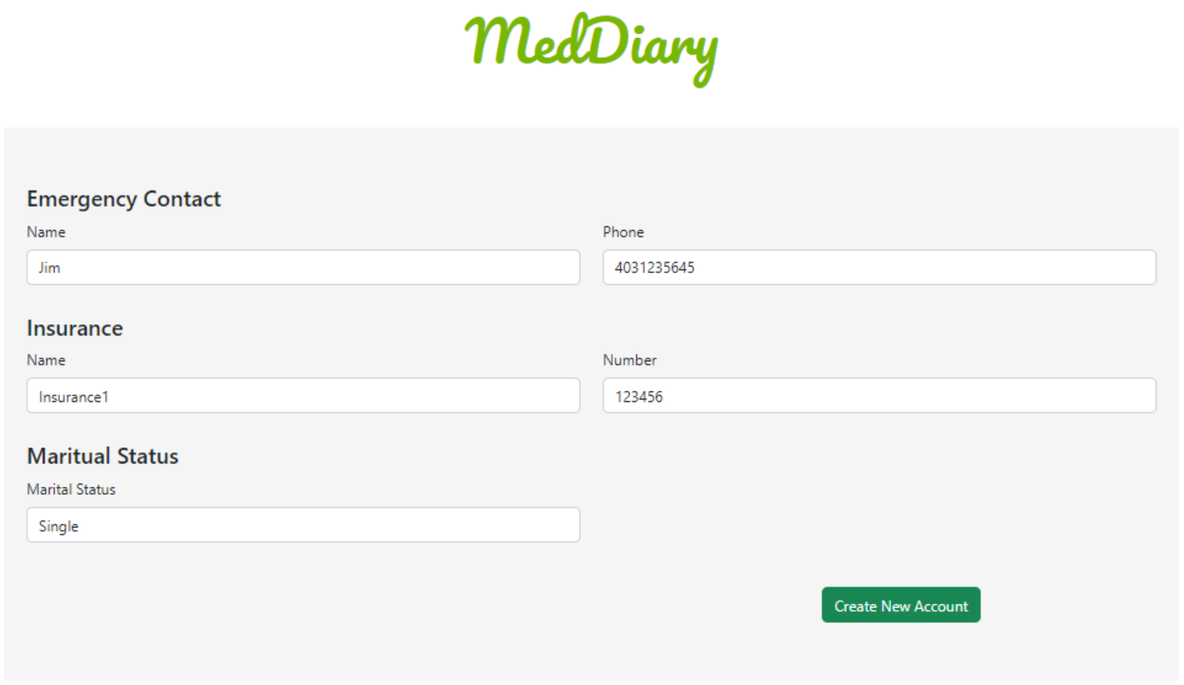
MedDiary

Personal Info

Email	John@gmail.com
First name	John
Last name	Doe
Address	123 Address rd NE
Date of Birth	01/01/2001
Sex	Male
Phone	4031234564
SIN	5
Password	••••••
Password	••••••
	Correct
Health Number	12345655

☐ I am a minor

Step 3: Fill in Emergency Contact, Insurance and Maritual Status information (Patient)



MedDiary

Emergency Contact

Name	Phone
Jim	4031235645

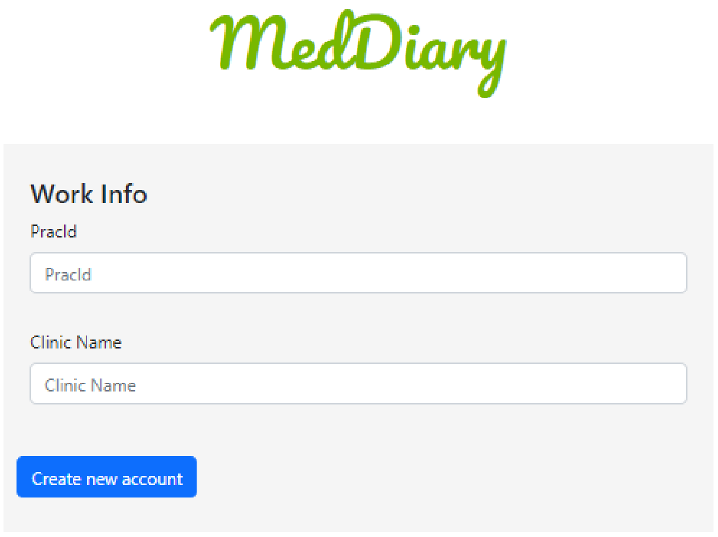
Insurance

Name	Number
Insurance1	123456

Maritual Status

Marital Status: Single

OR For Doctor



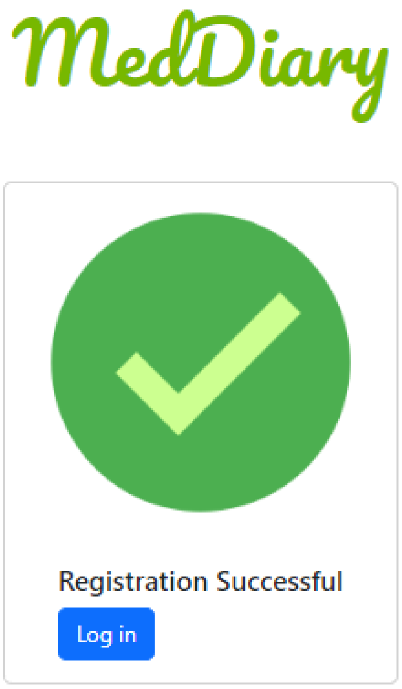
MedDiary

Work Info

PracId: PracId

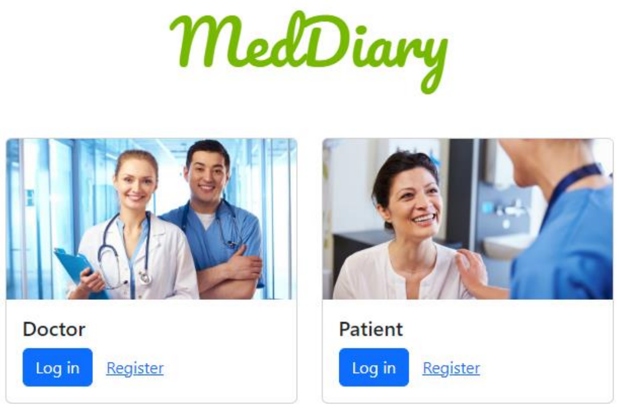
Clinic Name: Clinic Name

Step 4: Click on Create new account.



Log in (Patient)

Step 1: Select Login for patient.

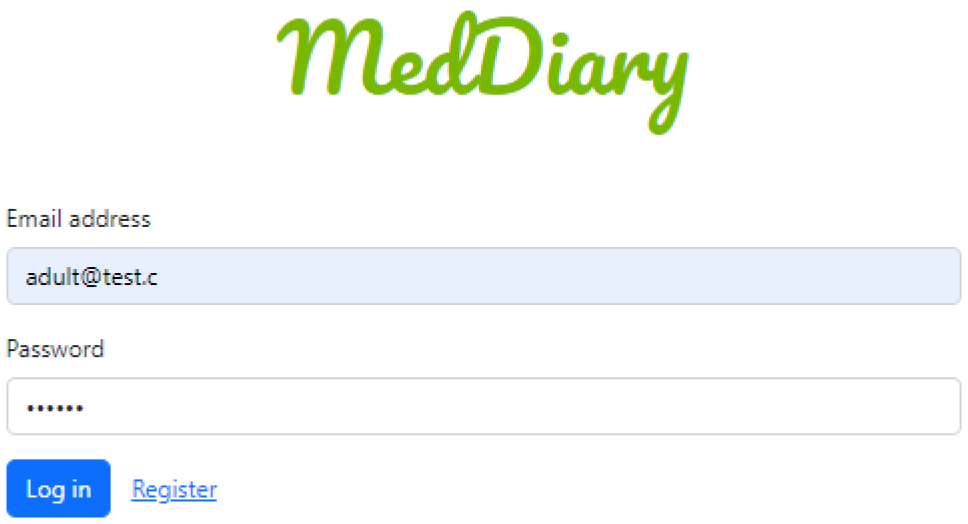


Step 2: Login to account

Test accounts for patients:

email: adult@test.c , password: P@ssw0

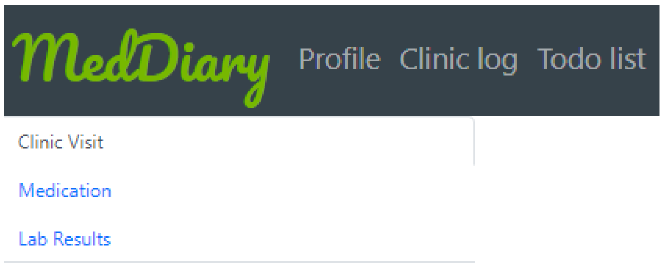
email: minor@test.c, password: P@ssw0



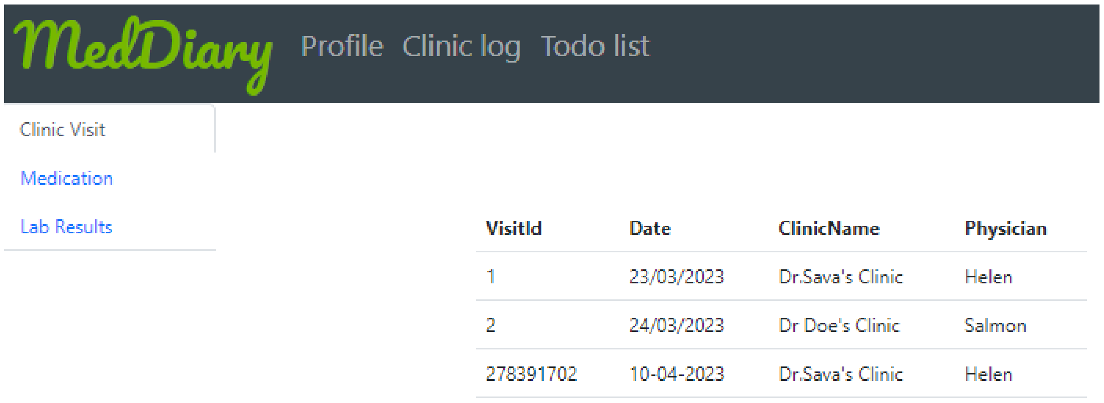
View Patient Clinic Visit Information (Patient View)

*Contains diagnosis, medication prescribed for that visit, and clinic information.

Step 1: Select Clinic Log on the Top Bar then Select Clnic Visit on the Side Bar

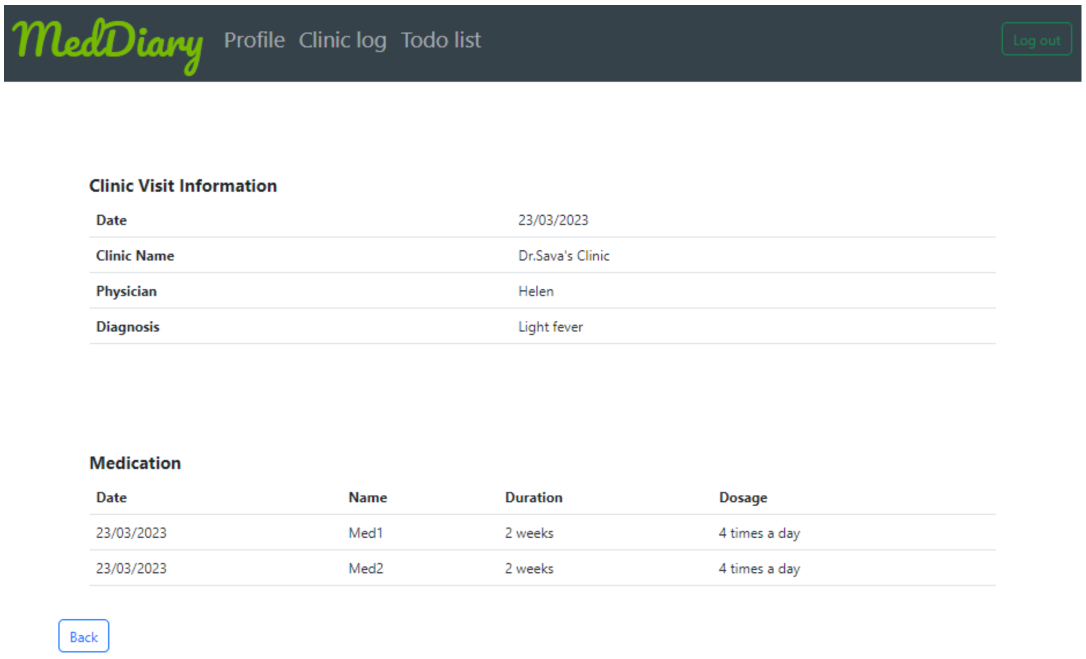


Step 2: Select Visit based on the date and location



VisitId	Date	ClinicName	Physician
1	23/03/2023	Dr.Sava's Clinic	Helen
2	24/03/2023	Dr Doe's Clinic	Salmon
278391702	10-04-2023	Dr.Sava's Clinic	Helen

Step 3: View the Clinic Visit Information, Medication and Diagnosis for that visit



Clinic Visit Information

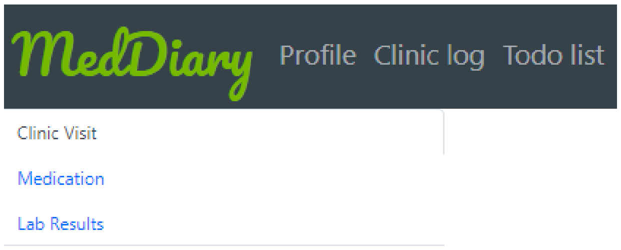
Date	23/03/2023
Clinic Name	Dr.Sava's Clinic
Physician	Helen
Diagnosis	Light fever

Medication

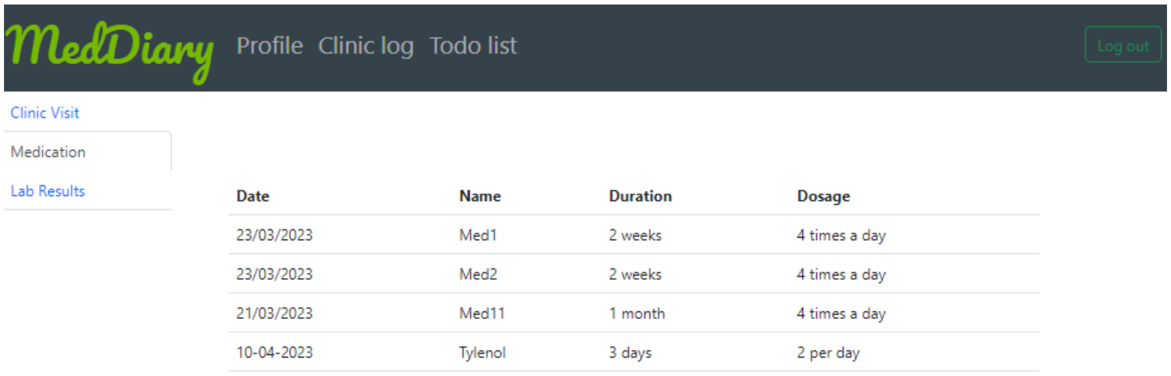
Date	Name	Duration	Dosage
23/03/2023	Med1	2 weeks	4 times a day
23/03/2023	Med2	2 weeks	4 times a day

View Patient Medication (Patient View)

Step 1: Select Clinic Log on the Top Bar then Select Medication on the Side Bar



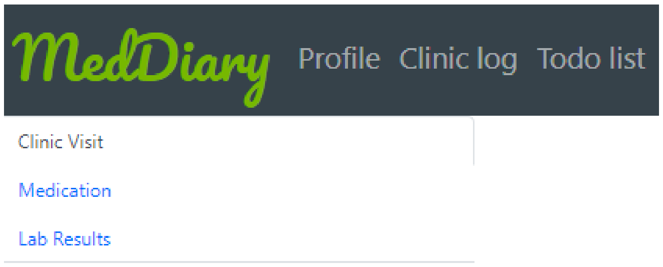
Step 2: View the Medication History



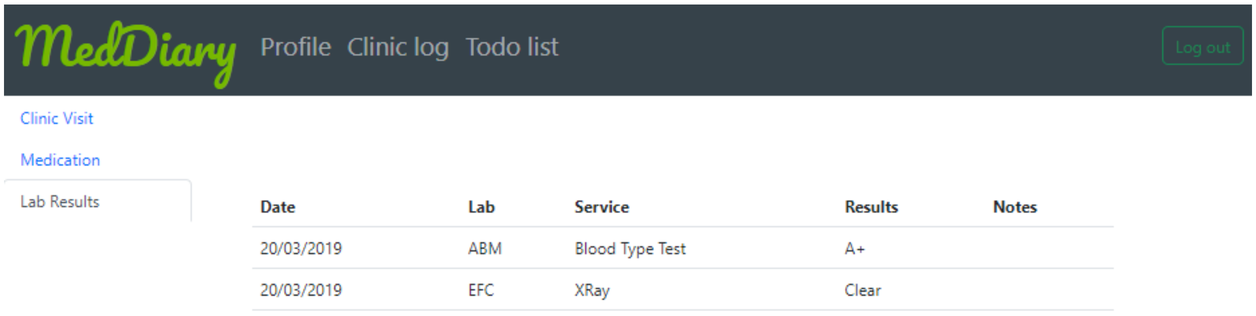
Date	Name	Duration	Dosage
23/03/2023	Med1	2 weeks	4 times a day
23/03/2023	Med2	2 weeks	4 times a day
21/03/2023	Med11	1 month	4 times a day
10-04-2023	Tylenol	3 days	2 per day

View Lab Results (Patient View)

Step 1: Select Clinic Log on the Top Bar then Select Lab Results on the Side Bar



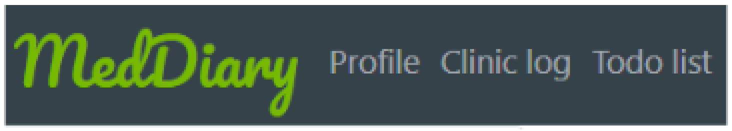
Step 2: View Lab Results



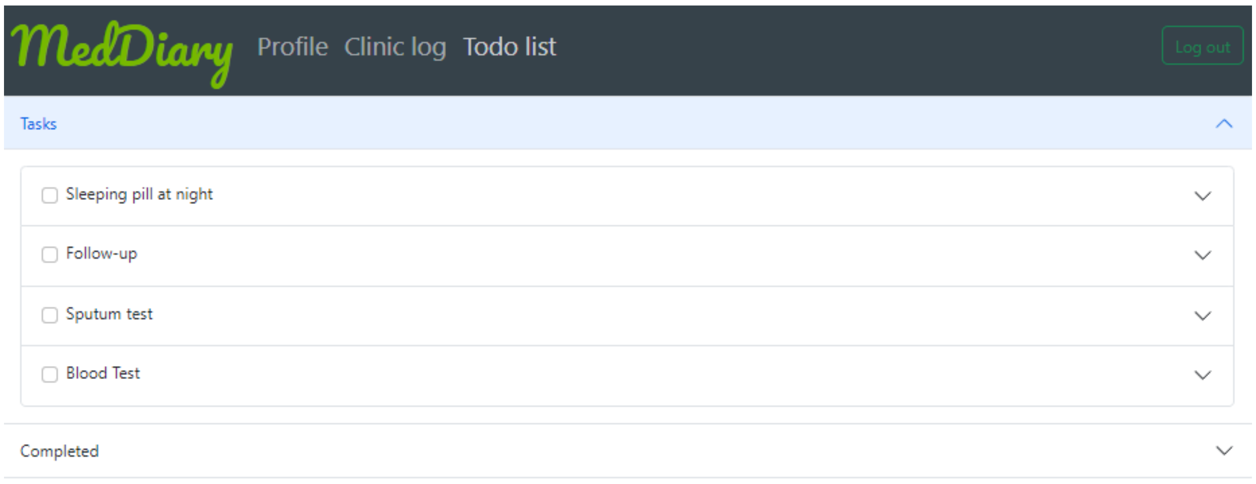
Date	Lab	Service	Results	Notes
20/03/2019	ABM	Blood Type Test	A+	
20/03/2019	EFC	XRay	Clear	

View Todo list (Patient View)

Step 1: Select Todo list on the Top Bar

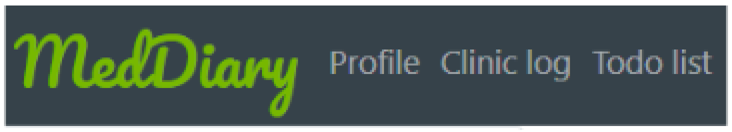


Step 2: View TodoList

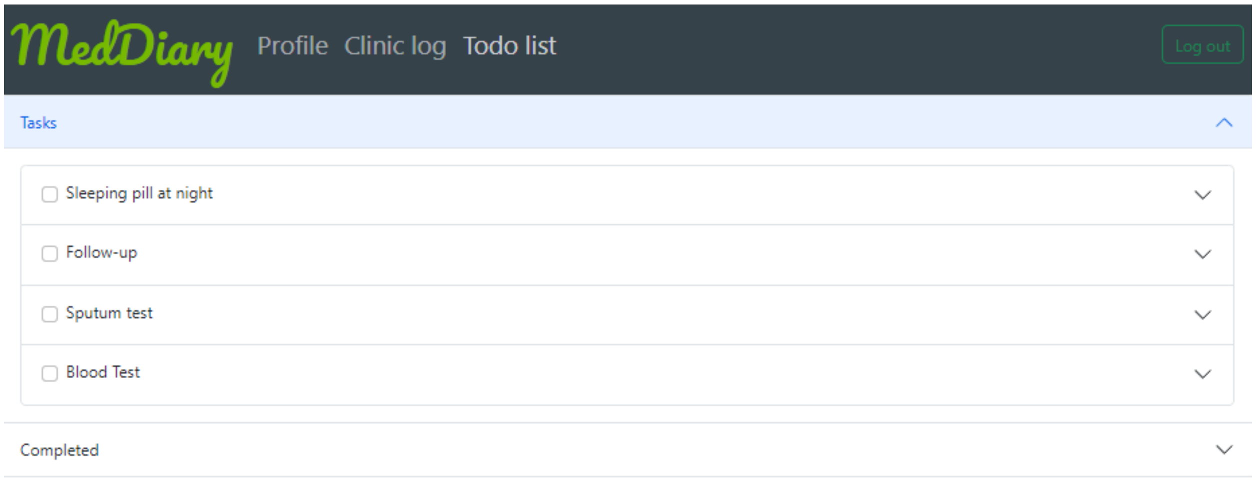


Check off Todo list (Patient View)

Step 1: Select Todo list on the Top Bar



Step 2: View TodoList

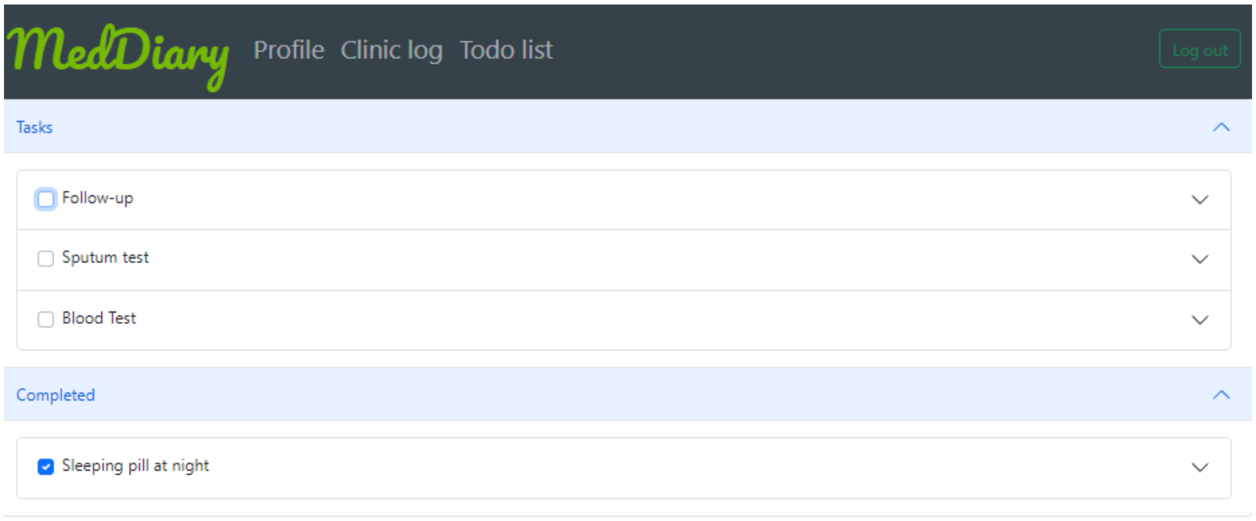


Tasks

- Sleeping pill at night
- Follow-up
- Sputum test
- Blood Test

Completed

Step 3: Check off Todo list



Tasks

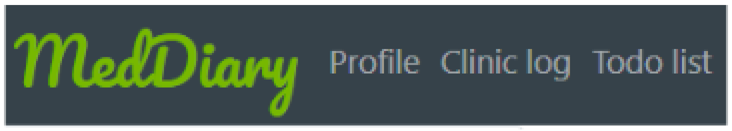
- Follow-up
- Sputum test
- Blood Test

Completed

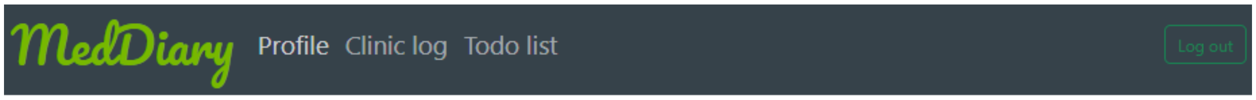
- [x] Sleeping pill at night

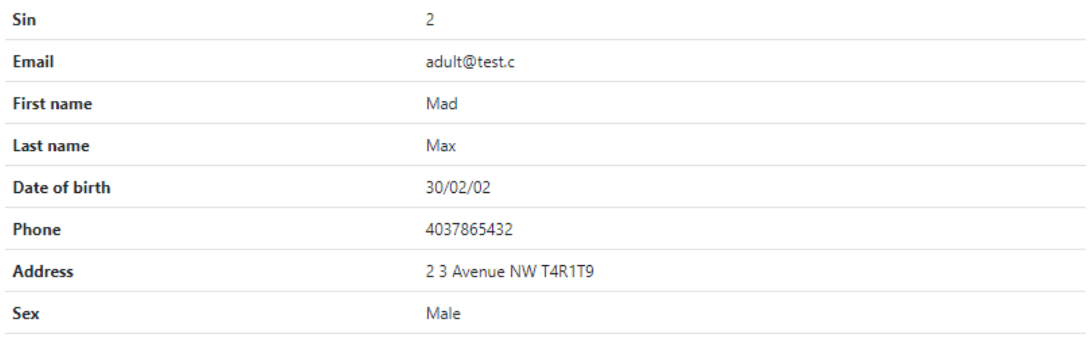
View Profile Information

Step 1: Select Profile list on the Top Bar



Step 2: View Profil

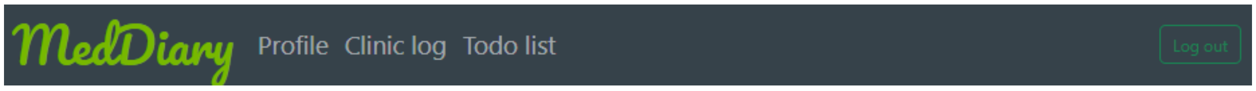




Sin	2
Email	adult@test.c
First name	Mad
Last name	Max
Date of birth	30/02/02
Phone	4037865432
Address	2 3 Avenue NW T4R1T9
Sex	Male

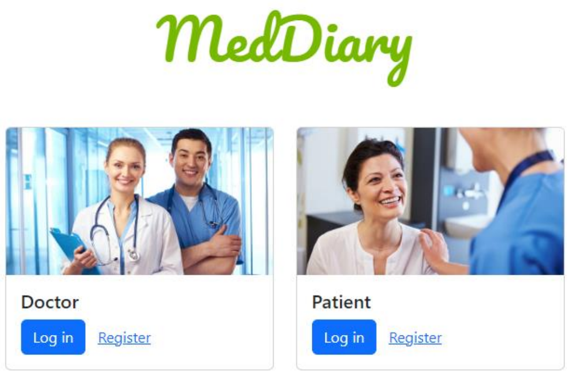
Log out of Account.

Step 1: Select Log out on the Top Left Bar



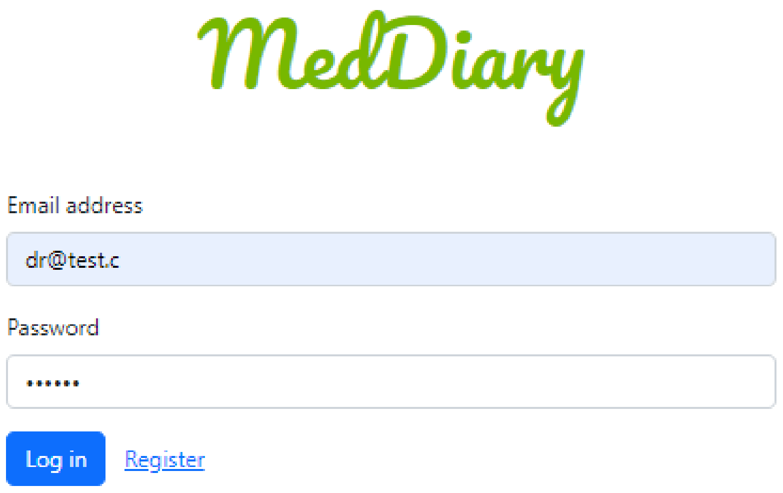
Log in (Doctor)

Step 1: Select Login for Doctor



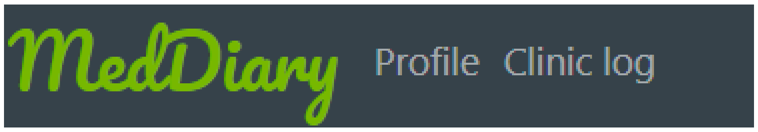
Step 2: Login to Doctor Account

Test account: email: dr@test.c , password: P@ssw0



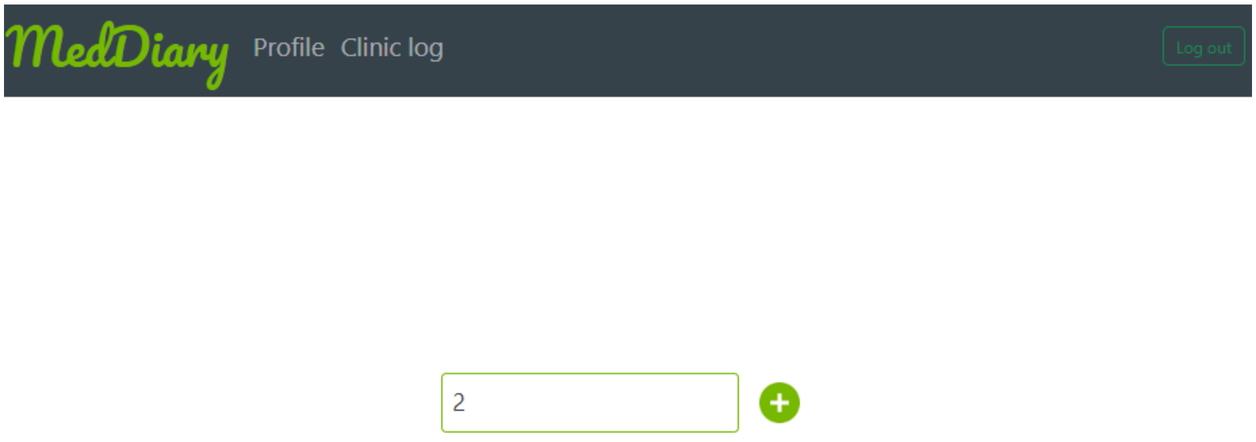
Creating Clinic Form (Doctor Only)

Step 1: Select Clnic log onTop Bar

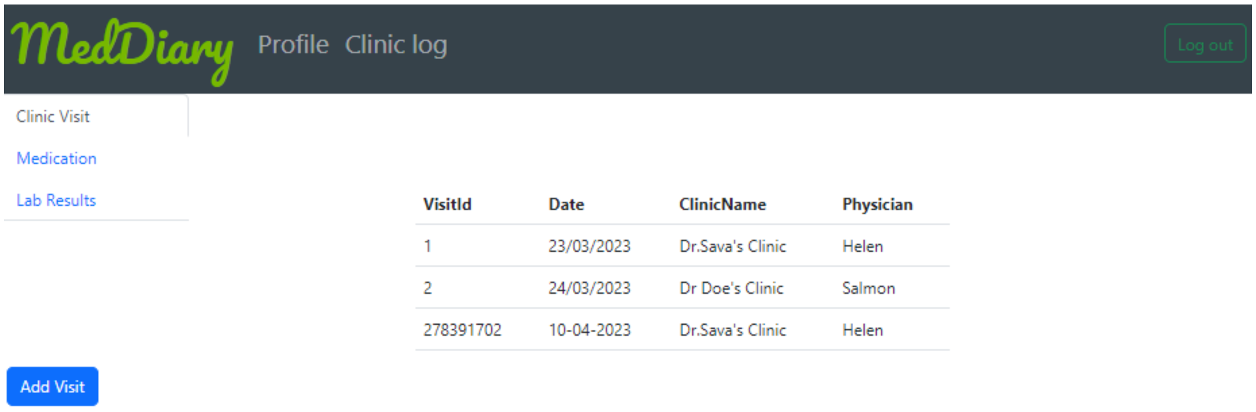


Step 2: Search Patient and then press +

Existing patient ids: 2 and 3

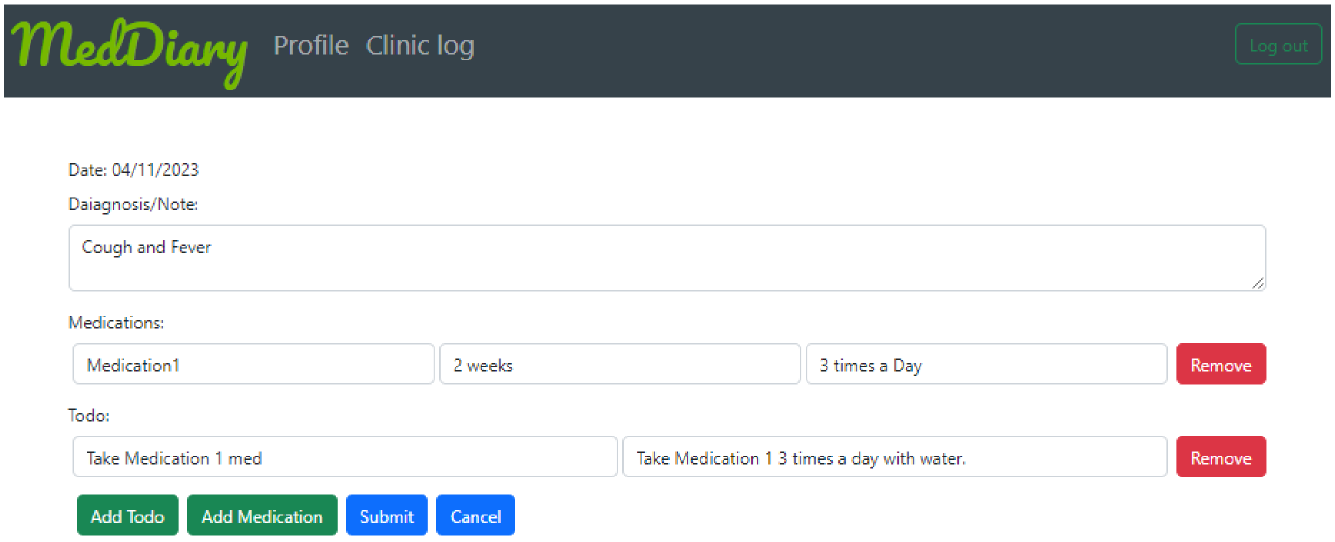


Step 3: Click on Add Visit



VisitId	Date	ClinicName	Physician
1	23/03/2023	Dr.Sava's Clinic	Helen
2	24/03/2023	Dr Doe's Clinic	Salmon
278391702	10-04-2023	Dr.Sava's Clinic	Helen

Step 4: Fill in Visit Form and click Submit



Date: 04/11/2023

Daiagnosis/Note:

Cough and Fever

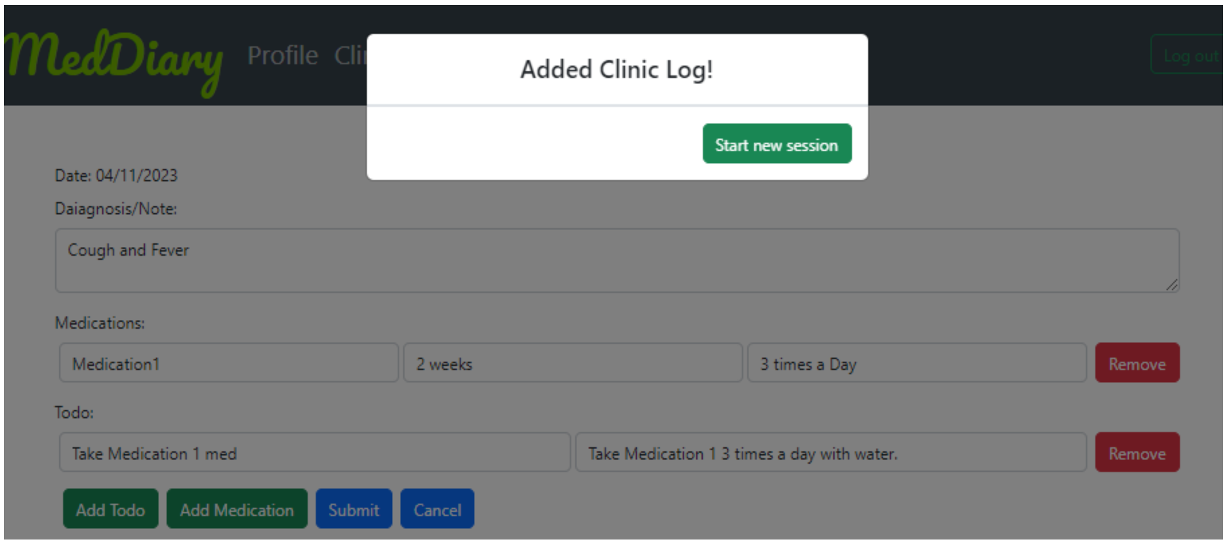
Medications:

Medication1 | 2 weeks | 3 times a Day

Todo:

Take Medication 1 med | Take Medication 1 3 times a day with water.

Step 5: Verify added Clinic Visit and Select Start New Session



Date: 04/11/2023

Daiagnosis/Note:

Cough and Fever

Medications:

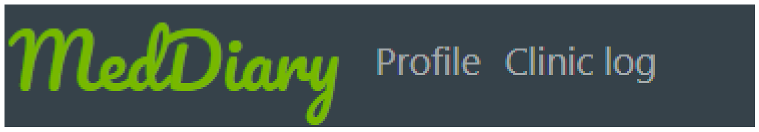
Medication1 | 2 weeks | 3 times a Day

Todo:

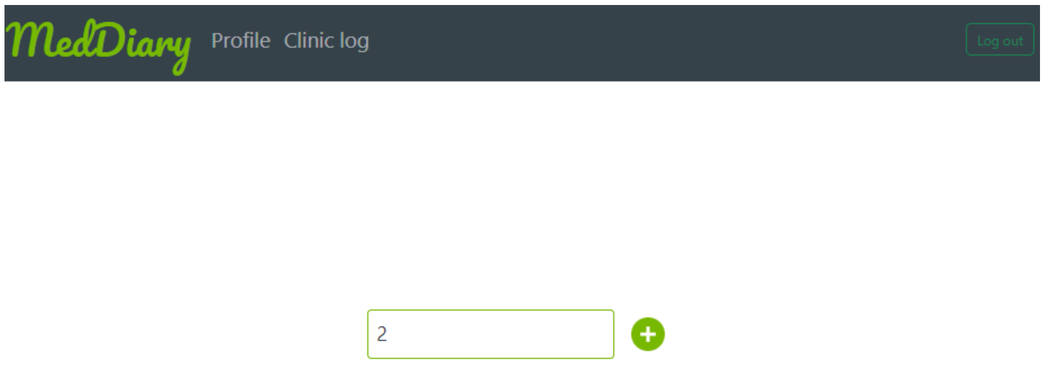
Take Medication 1 med | Take Medication 1 3 times a day with water.

View Patient Information (Doctor View)

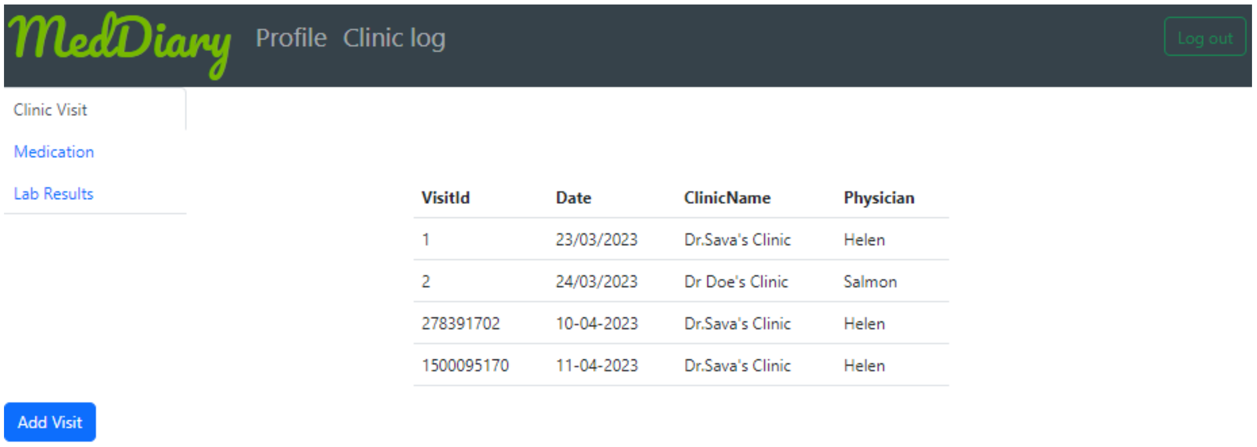
Step 1: Select Clnic log onTop Bar



Step 2: Search Patient and then press +

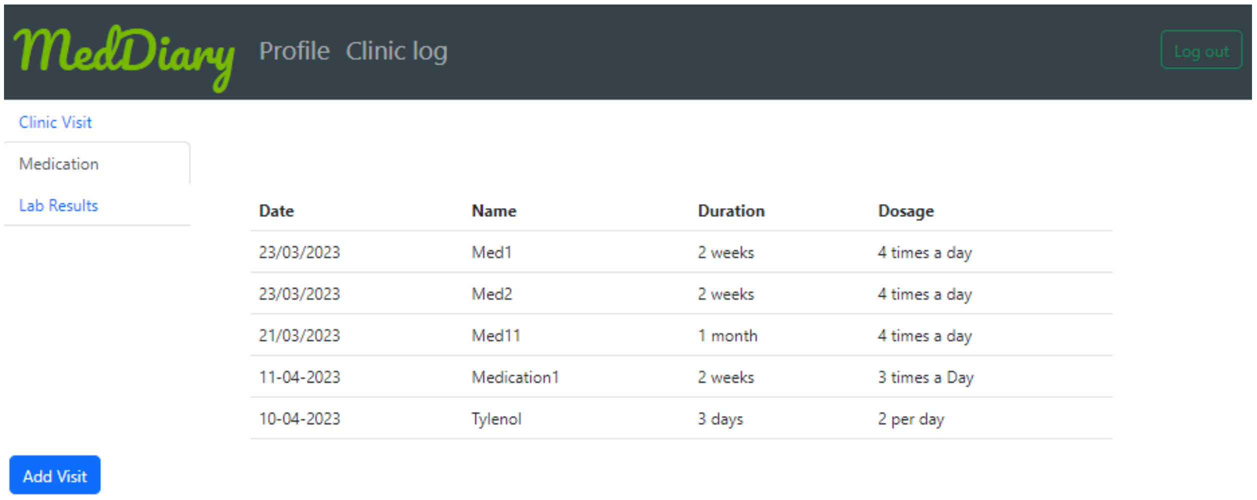


Step 3: Select Clinic Visit Information or Medication History or Lab Results



VisitId	Date	ClinicName	Physician
1	23/03/2023	Dr.Sava's Clinic	Helen
2	24/03/2023	Dr Doe's Clinic	Salmon
278391702	10-04-2023	Dr.Sava's Clinic	Helen
1500095170	11-04-2023	Dr.Sava's Clinic	Helen

Step 4: View Selected Bar

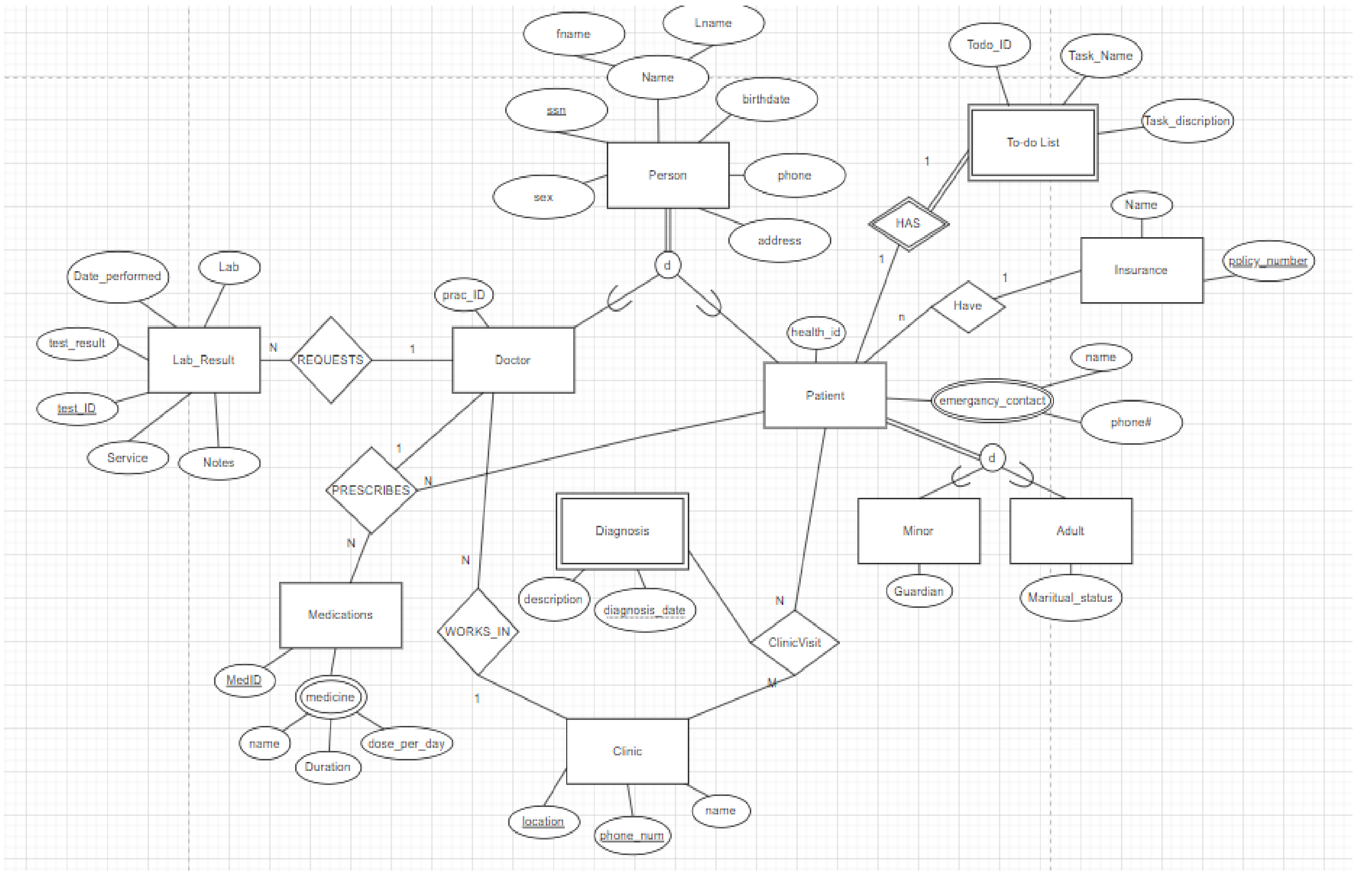


Date	Name	Duration	Dosage
23/03/2023	Med1	2 weeks	4 times a day
23/03/2023	Med2	2 weeks	4 times a day
21/03/2023	Med11	1 month	4 times a day
11-04-2023	Medication1	2 weeks	3 times a Day
10-04-2023	Tylenol	3 days	2 per day

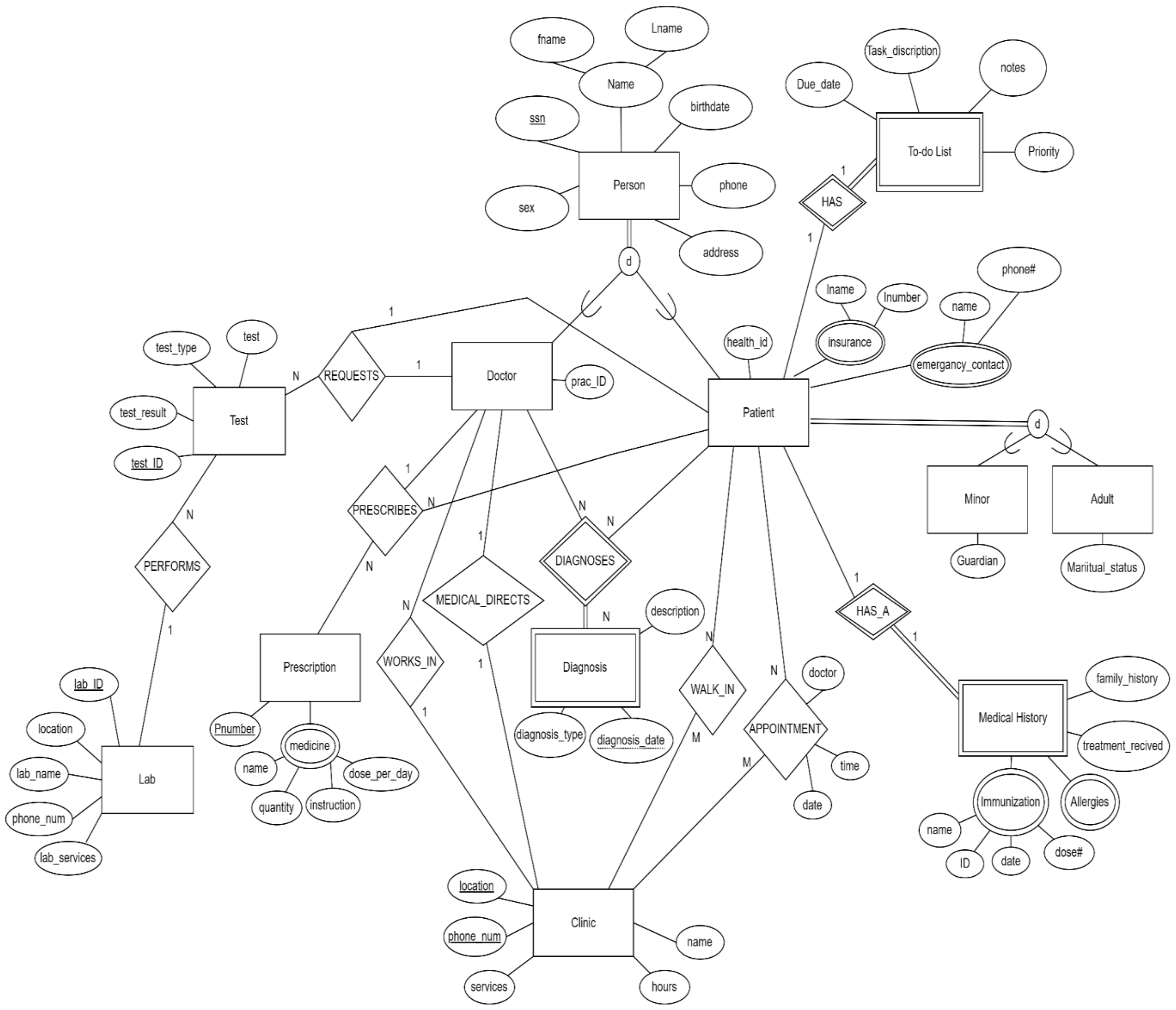
Appendices

Appendix 1: Extended Entity Relationship Model

Updated

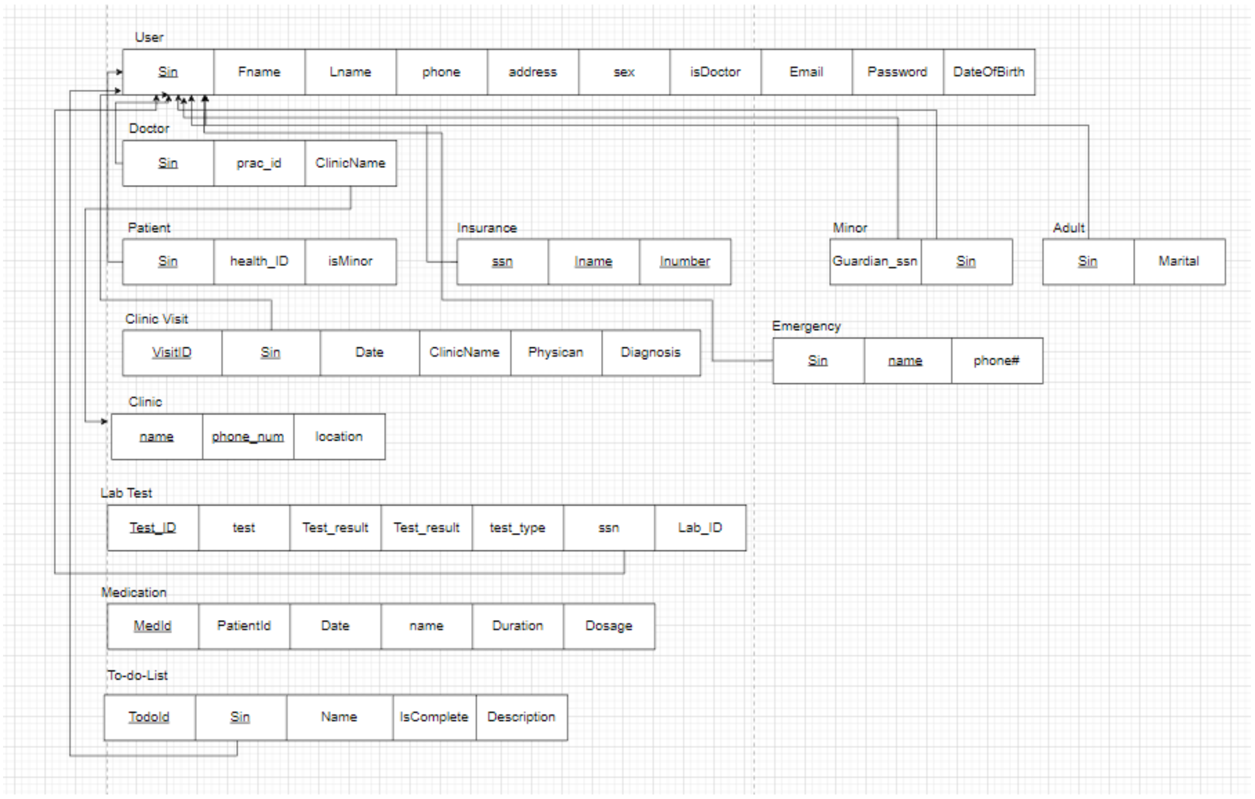


Old

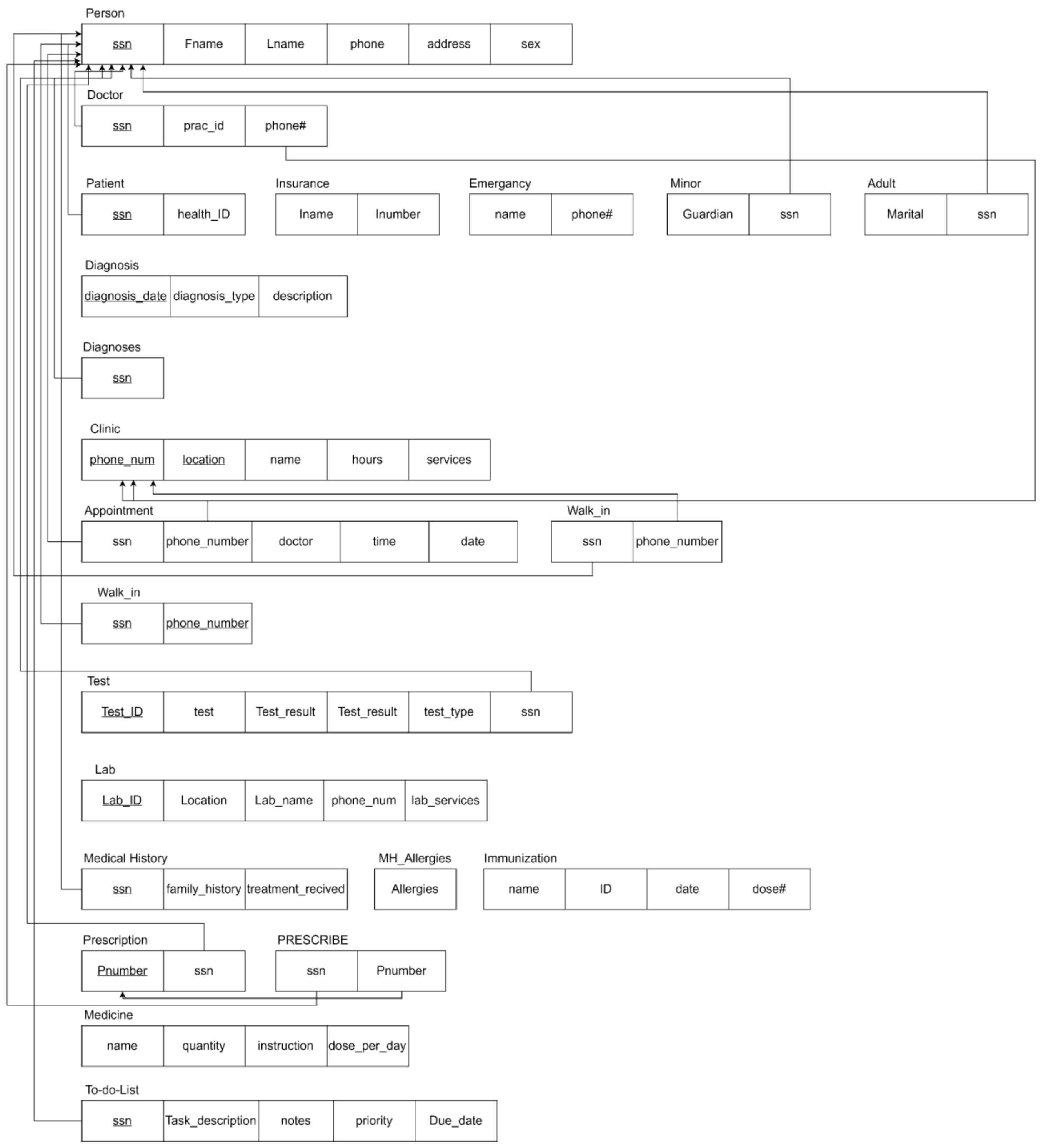

fname
Lname
Name
ssn
birthdate
Person
phone
sex
address
d
Task_discription
notes
Due_date
To-do List
Priority
HAS
1
1
lname
lnumber
insurance
phone#
name
emergancy_contact
health_id
Patient
d
Minor
Adult
Guardian
Mariitual_status
test_type
test
test_result
Test
test_ID
REQUESTS
N
1
1
Doctor
prac_ID
PRESCRIBES
1
N
N
DIAGNOSES
N
N
1
MEDICAL_DIRECTS
1
description
Diagnosis
diagnosis_type
diagnosis_date
PERFORMS
N
1
lab_ID
location
lab_name
phone_num
lab_services
Lab
Prescription
Pnumber
medicine
name
dose_per_day
quantity
instruction
WORKS_IN
N
1
WALK_IN
N
M
APPOINTMENT
N
M
doctor
time
date
HAS_A
1
1
Medical History
family_history
treatment_recived
Immunization
Allergies
name
ID
date
dose#
Clinic
location
phone_num
services
name
hours

Appendix 2: Relational Model

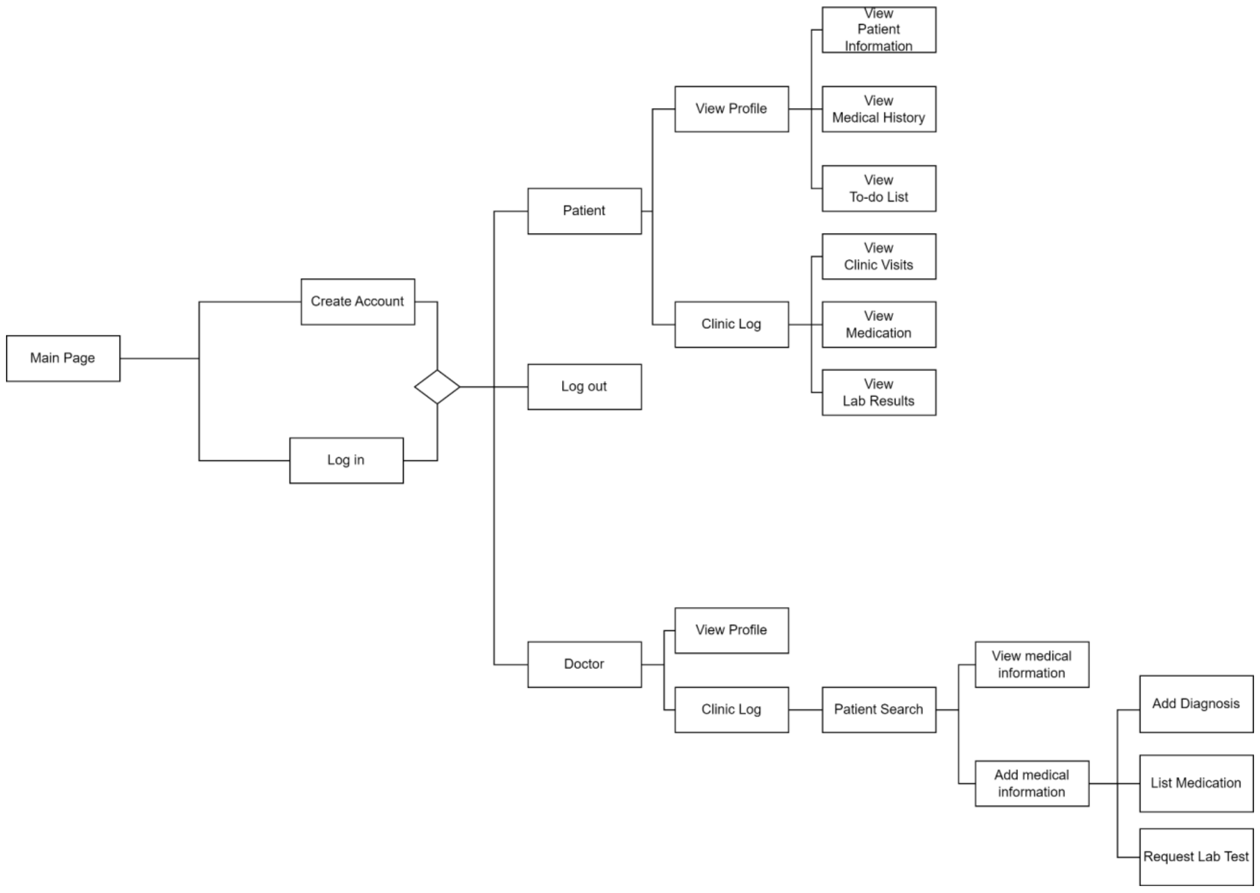
Updated


User
Sin | Fname | Lname | phone | address | sex | isDoctor | Email | Password | DateOfBirth
Doctor
Sin | prac_id | ClinicName
Patient
Sin | health_ID | isMinor
Insurance
ssn | Iname | Inumber
Minor
Guardian_ssn | Sin
Adult
Sin | Marital
Clinic Visit
VisitID | Sin | Date | ClinicName | Physican | Diagnosis
Emergency
Sin | name | phone#
Clinic
name | phone_num | location
Lab Test
Test_ID | test | Test_result | Test_result | test_type | ssn | Lab_ID
Medication
MedId | PatientId | Date | name | Duration | Dosage
To-do-List
TodoId | Sin | Name | IsComplete | Description

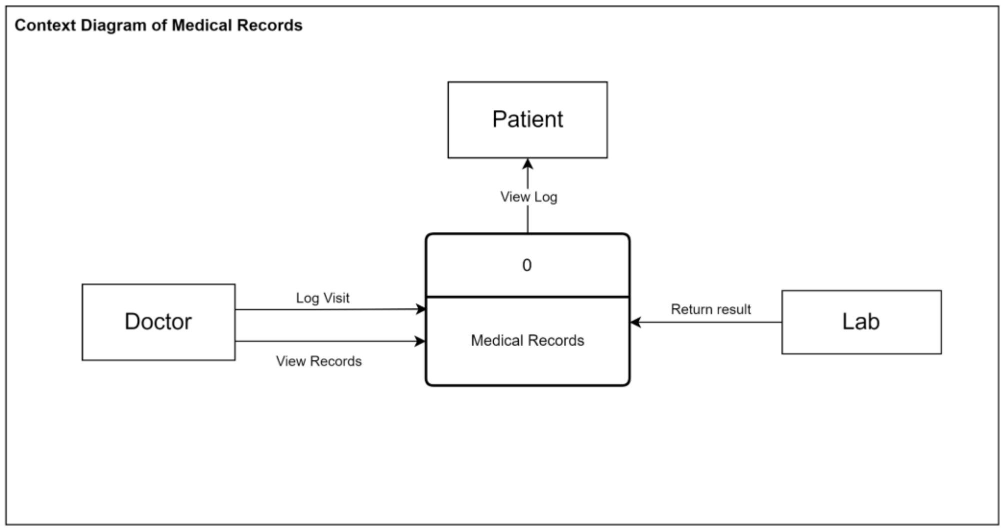
Old

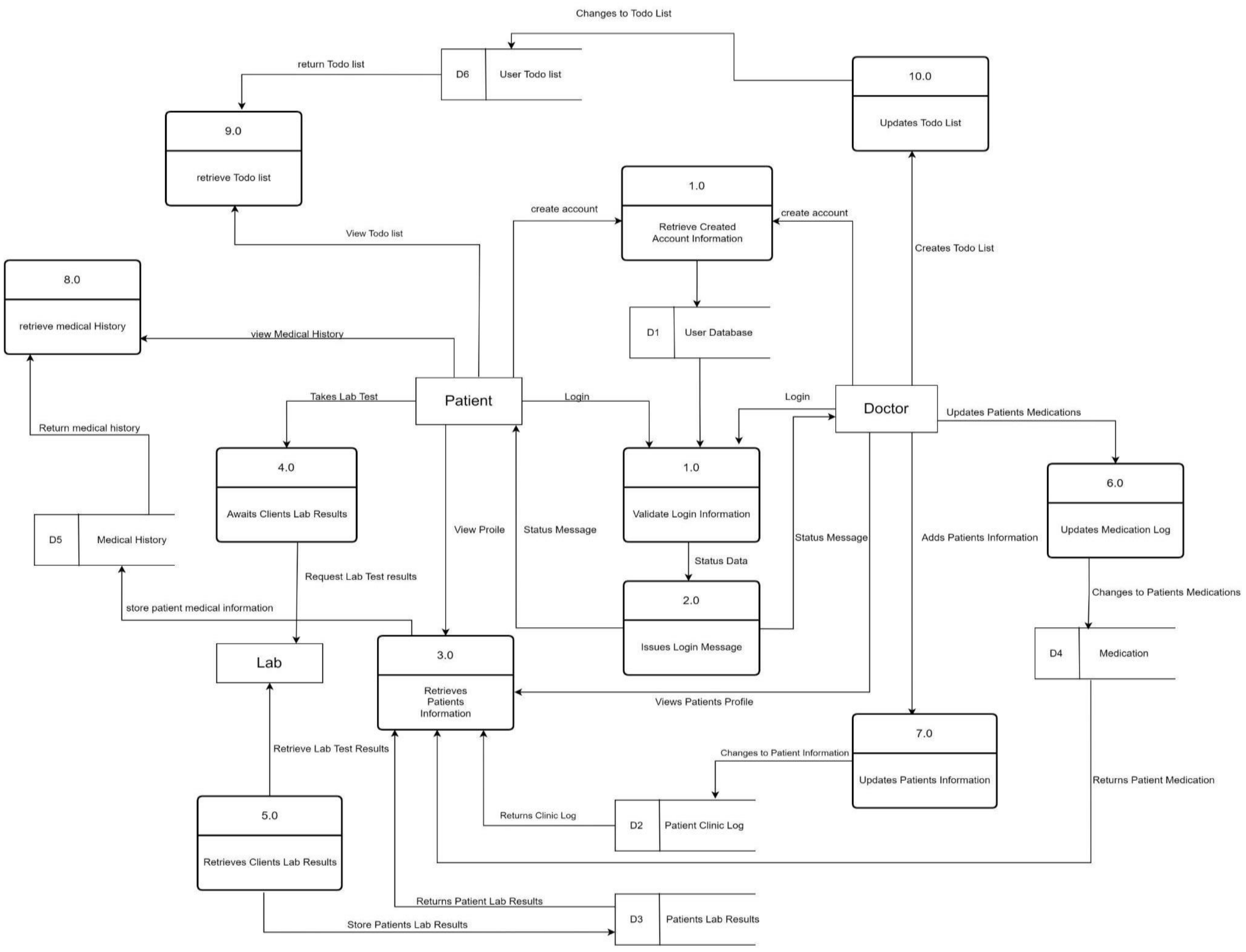

Person
ssn
Fname
Lname
phone
address
sex
Doctor
ssn
prac_id
phone#
Patient
ssn
health_ID
Insurance
Iname
Inumber
Emergancy
name
phone#
Minor
Guardian
ssn
Adult
Marital
ssn
Diagnosis
diagnosis_date
diagnosis_type
description
Diagnoses
ssn
Clinic
phone_num
location
name
hours
services
Appointment
ssn
phone_number
doctor
time
date
Walk_in
ssn
phone_number
Walk_in
ssn
phone_number
Test
Test_ID
test
Test_result
Test_result
test_type
ssn
Lab
Lab_ID
Location
Lab_name
phone_num
lab_services
Medical History
ssn
family_history
treatment_recived
MH_Allergies
Allergies
Immunization
name
ID
date
dose#
Prescription
Pnumber
ssn
PRESCRIBE
ssn
Pnumber
Medicine
name
quantity
instruction
dose_per_day
To-do-List
ssn
Task_description
notes
priority
Due_date

Appendix 3: HIPO Model



Appendix 4: DFD Model




Changes to Todo List
D6
User Todo list
return Todo list
10.0
Updates Todo List
9.0
retrieve Todo list
1.0
Retrieve Created Account Information
create account
create account
View Todo list
Creates Todo List
8.0
retrieve medical History
D1
User Database
view Medical History
Patient
Takes Lab Test
Login
Login
Doctor
Updates Patients Medications
Return medical history
4.0
Awaits Clients Lab Results
1.0
Validate Login Information
6.0
Updates Medication Log
D5
Medical History
View Proile
Status Message
Status Message
Adds Patients Information
Status Data
Request Lab Test results
Changes to Patients Medications
store patient medical information
2.0
Issues Login Message
D4
Medication
Lab
3.0
Retrieves Patients Information
Views Patients Profile
7.0
Updates Patients Information
Retrieve Lab Test Results
Changes to Patient Information
Returns Patient Medication
Returns Clinic Log
D2
Patient Clinic Log
5.0
Retrieves Clients Lab Results
Returns Patient Lab Results
D3
Patients Lab Results
Store Patients Lab Results

Appendix 5: Transactions, and their Associated SQL Queries and API Functionality

*Sin of 2, 6, 7 was used as an example for all SQL queries

Function: Create Patient Account

Statement:

```
INSERT INTO
"Users"("Sin","Address","Email","Fname","IsDoctor","Lname","Password","Phone","Sex")
VALUES (87, NULL,",",0,",",NULL,NULL);
INSERT INTO "EmergencyContacts"("Sin","Name","Phone") VALUES (6,",");
INSERT INTO "Insurances"("Sin","Iname","Inumber") VALUES (6,",0);
INSERT INTO "Adults"("Sin","MaritalStatus") VALUES (6, NULL);
```

Function: Delete Patient Account

Statement:

```
DELETE FROM "Users" WHERE SIN == 7;
DELETE FROM "EmergencyContacts" WHERE SIN == 7;
DELETE FROM "Insurances" WHERE SIN == 7;
DELETE FROM "Adults" WHERE SIN == 7;
```

Function: Create Doctor Account

Statement:

```
INSERT INTO
"Users"("Sin","Address","Email","Fname","IsDoctor","Lname","Password","Phone","Sex")
VALUES (87,NULL,",",1,",",NULL,NULL);
```

INSERT INTO "Doctor"("Sin","PracId","Clinic") VALUES (6,12,"Dr.Sava’s Clinic");

Function: Delete Doctor Account

Statement:

DELETE

FROM "Users"

WHERE SIN == 6;

DELETE

FROM "Doctors"

WHERE SIN == 6;

Function: Check Login Information

Statement:

SELECT Email, Password

FROM "Users"

WHERE Email == "dr@test.c" and Password == "P@ssw0"

Function: Insert Medication

Statement:

INSERT INTO "Medications"("MedId","Date","Dosage","Duration","Name","PatientId")

VALUES (2,",",",",0);

Function: View Medication

Statement:

```
SELECT *
FROM "Medications"
 WHERE PatientId == 2
```

Function: Delete Medication

Statement:

```
DELETE FROM "Medications"
WHERE MedId = “?’
```

Function: Insert Clinic Visit

Statement:

```
INSERT INTO "ClinicVisits"("VisitId","Sin","Date","ClinicName","Physician","Diagnosis")
VALUES (2,'',NULL,NULL,NULL);
```

Function: View Clinic Visit

Statement:

```
SELECT *
FROM "ClinicVisits"
WHERE Sin == 2
```

Function: Delete Clinic Visit

Statement:

```
DELETE FROM "Clinic Visits"
WHERE VisitId = “?’
```

Function: Insert Lab Results

Statement:

```
INSERT INTO "Labtest"("TestId","PatientSin","Date","Lab","Service","Results","Notes")
VALUES (5,2,NULL,NULL,NULL,NULL,NULL);
```

Function: View Lab Results

Statement:

```
SELECT *
FROM "Labtest"
WHERE PatientSin == 2
```

Function: Insert TodoLists

Statement:

```
INSERT INTO "TodoLists"("TodoId","Sin","Name","IsComplete","Description") VALUES
(476920626,2,NULL,0,NULL);
```

Function: View TodoLists

Statement:

```
SELECT *
FROM "Todo Lists"
WHERE Sin == 2
```

Function: Insert EmergancyContacts

Statement:

```
INSERT INTO "EmergencyContacts"("Sin","Name","Phone") VALUES (2,",");
```

Function: View EmergancyContacts

Statement:

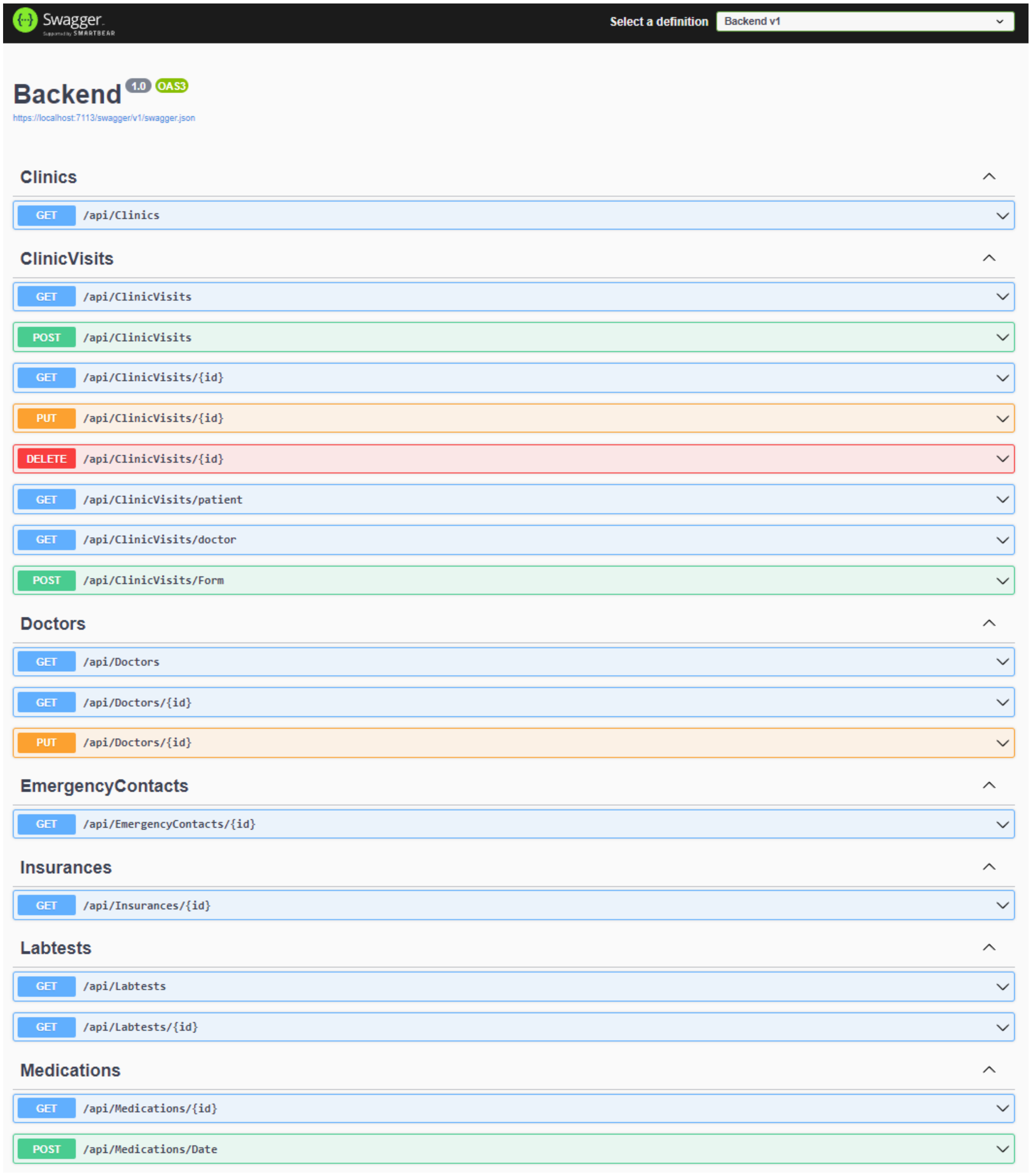
```
SELECT *
FROM "EmergancyContacts"
WHERE Sin == 2
```

Function: Insert Clinic Visit Form

Statement:

```
INSERT INTO "ClinicVisits"("VisitId","Sin","Date","ClinicName","Physician","Diagnosis")
VALUES (2,",NULL,NULL,NULL);
INSERT INTO "Medications"("MedId","Date","Dosage","Duration","Name","PatientId")
VALUES (2,",",",",0);
INSERT INTO "TodoLists"("TodoId","Sin","Name","IsComplete","Description") VALUES
(476920626,2,NULL,0,NULL);
```

Appendix 6: API Documentation



Backend 1.0 OAS3

https://localhost:7113/swagger/v1/swagger.json

Clinics

GET /api/Clinics

ClinicVisits

GET /api/ClinicVisits

POST /api/ClinicVisits

GET /api/ClinicVisits/{id}

PUT /api/ClinicVisits/{id}

DELETE /api/ClinicVisits/{id}

GET /api/ClinicVisits/patient

GET /api/ClinicVisits/doctor

POST /api/ClinicVisits/Form

Doctors

GET /api/Doctors

GET /api/Doctors/{id}

PUT /api/Doctors/{id}

EmergencyContacts

GET /api/EmergencyContacts/{id}

Insurances

GET /api/Insurances/{id}

Labtests

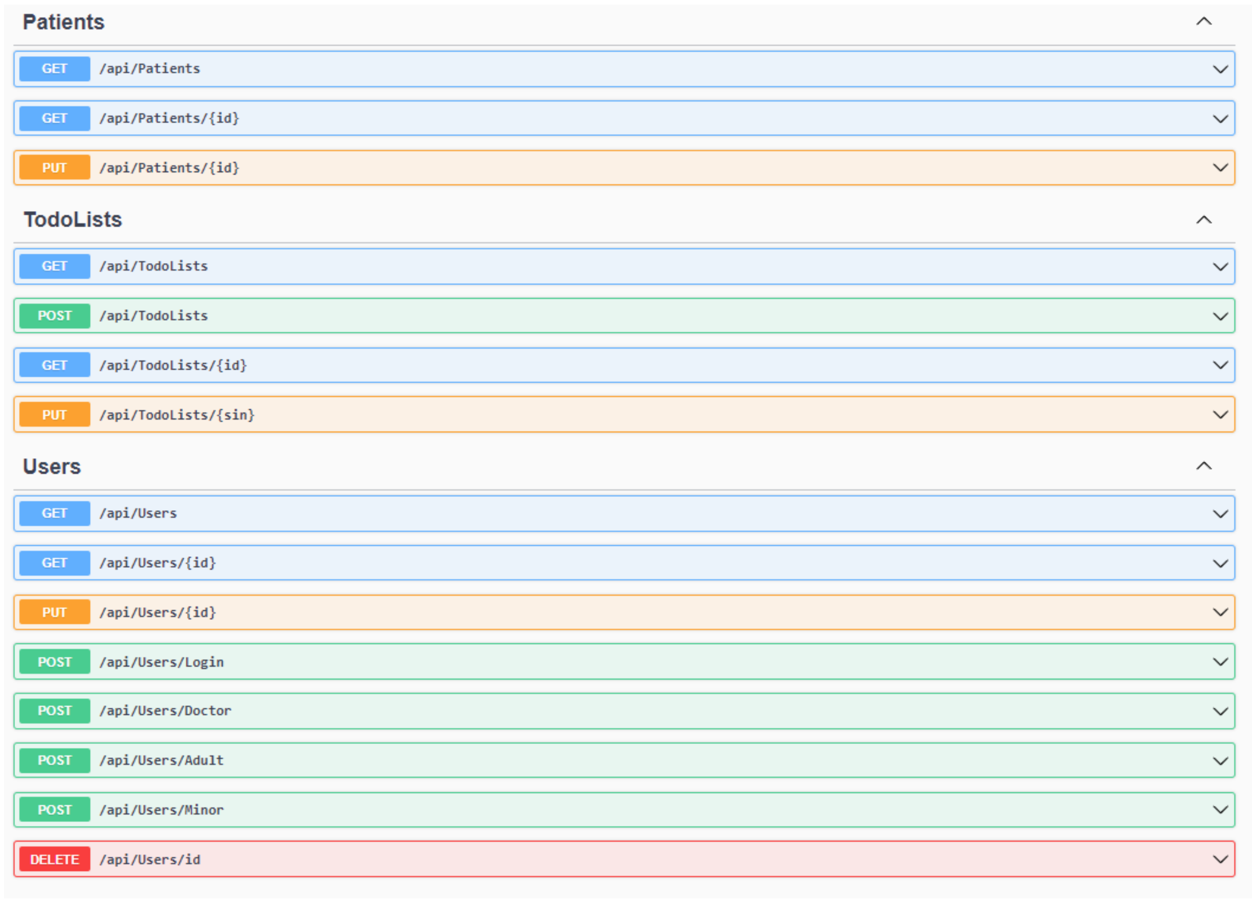
GET /api/Labtests

GET /api/Labtests/{id}

Medications

GET /api/Medications/{id}

POST /api/Medications/Date



Patients

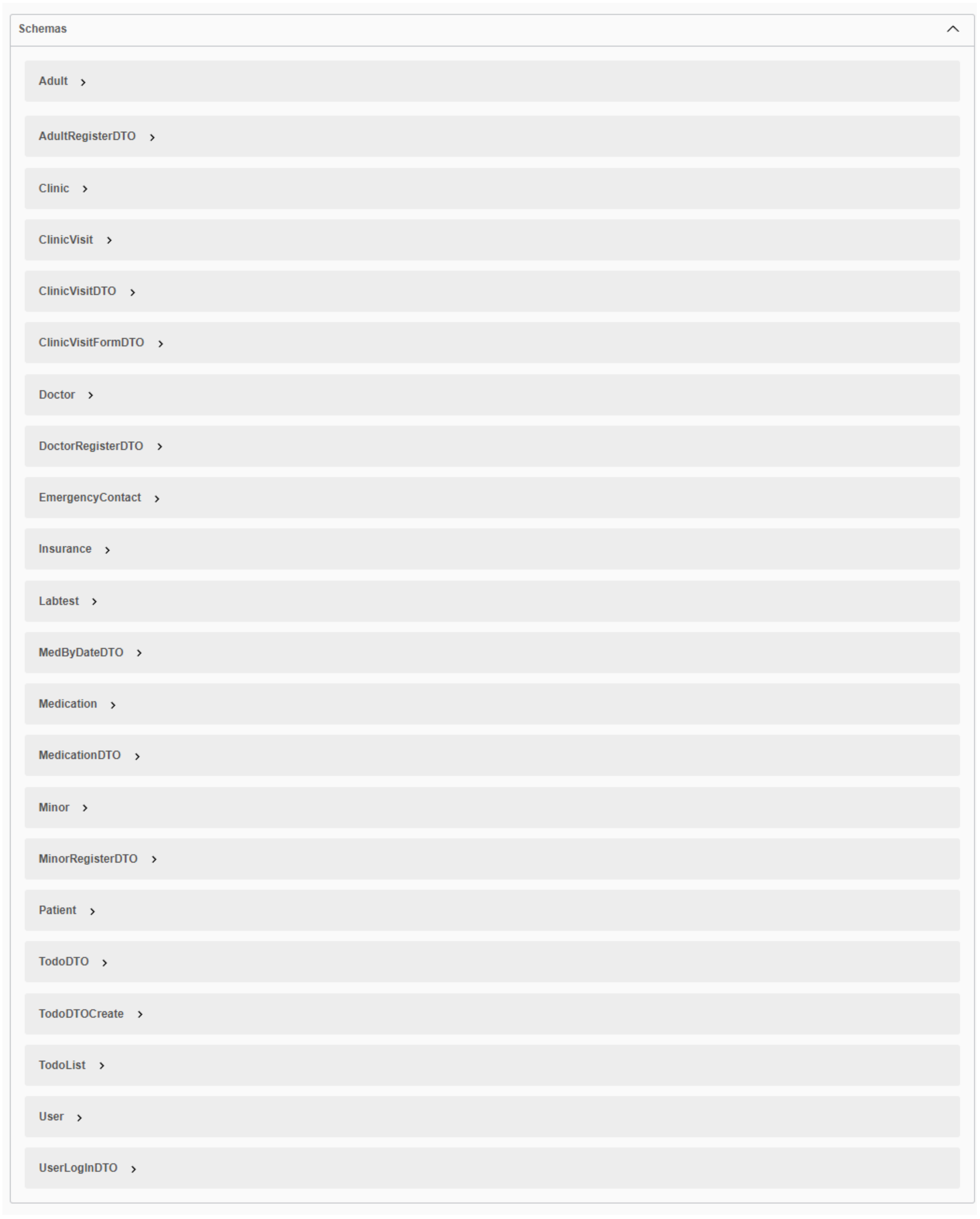
```
GET     /api/Patients
GET     /api/Patients/{id}
PUT     /api/Patients/{id}
```

TodoLists

```
GET     /api/TodoLists
POST    /api/TodoLists
GET     /api/TodoLists/{id}
PUT     /api/TodoLists/{sin}
```

Users

```
GET     /api/Users
GET     /api/Users/{id}
PUT     /api/Users/{id}
POST    /api/Users/Login
POST    /api/Users/Doctor
POST    /api/Users/Adult
POST    /api/Users/Minor
DELETE  /api/Users/id
```



Schemas

Adult

AdultRegisterDTO

Clinic

ClinicVisit

ClinicVisitDTO

ClinicVisitFormDTO

Doctor

DoctorRegisterDTO

EmergencyContact

Insurance

Labtest

MedByDateDTO

Medication

MedicationDTO

Minor

MinorRegisterDTO

Patient

TodoDTO

TodoDTOCreate

TodoList

User

UserLogInDTO

References

1. Woodson, J. (2017). *When things go wrong: Medical error a leading cause of death*. Pulsara. Retrieved January 29, 2023, from https://www.pulsara.com/blog/medical-error-a-leading-cause-of-death
2. Government of Alberta. (n.d.). Common Questions from Albertans. Alberta Netcare. Retrieved January 29, 2023, from https://www.albertanetcare.ca/commonquestions.htm
3. McCabe, R. (2018). *Miscommunication in doctor–patient communication*. Retrieved January 29, 2023, from https://onlinelibrary.wiley.com/doi/epdf/10.1111/tops.12337
4. Morgan, S. (1970, January 1). *Miscommunication between patients and General Practitioners: Implications for Clinical Practice*. CSIRO PUBLISHING. Retrieved January 29, 2023, from https://www.publish.csiro.au/HC/HC13123