Talia Muro (2050123)

Selihom Efrem Ogbe

**Clinic Management System**

Programming Patterns 420-301-VA

Niloufar Salehi Dastjerdi

Table of Contents

[Description 4](#_Toc197167565)

[Features 4](#_Toc197167566)

[Challenges 8](#_Toc197167567)

[Learning Outcomes 13](#_Toc197167568)

# Description

Our goal is to create an online medical management system that makes it easy for receptionists to manage patients, doctors, appointments, and medical records. They will be able to book appointments, view medical records electronically, add patients as well as medical professionals, and will even be able to search for and modify patients, doctors, appointments, and medical records.

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Features:

## **Patients Tab**

### Add Patient Button

A screenshot of a computer error

AI-generated content may be incorrect. A screenshot of a computer error

AI-generated content may be incorrect.A screenshot of a computer error message

AI-generated content may be incorrect. A screenshot of a computer error message

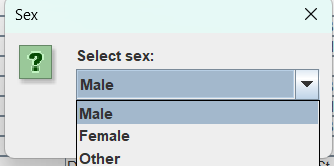
AI-generated content may be incorrect.

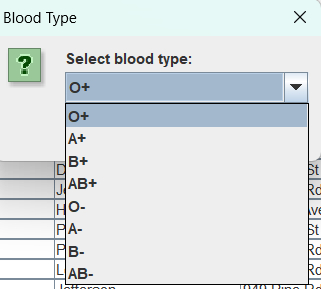
A screenshot of a computer error

AI-generated content may be incorrect. A screenshot of a calendar

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect. 



### Modify Patient Button

A screenshot of a computer error

AI-generated content may be incorrect. A screenshot of a computer error

AI-generated content may be incorrect.

A screenshot of a computer error

AI-generated content may be incorrect.

### Find Patient Button

A screenshot of a computer error

AI-generated content may be incorrect. A screenshot of a computer error

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

### View All Patients Button

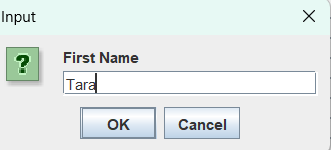
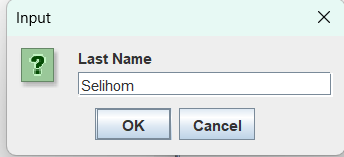
A screenshot of a computer

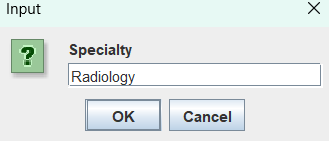
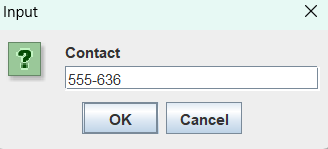
AI-generated content may be incorrect.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Doctors Tab**

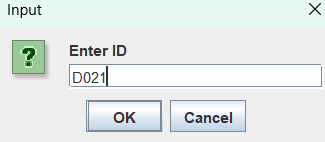
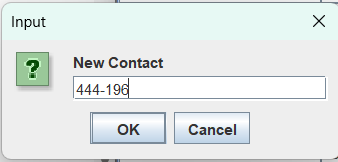
### Add Doctor Button

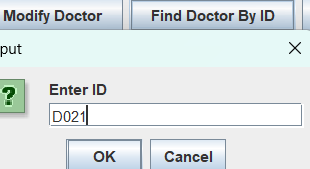


### Modify Doctor Button

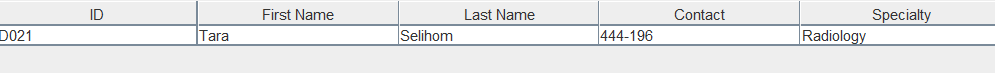
 



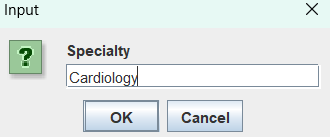
### Find Doctor By ID Button

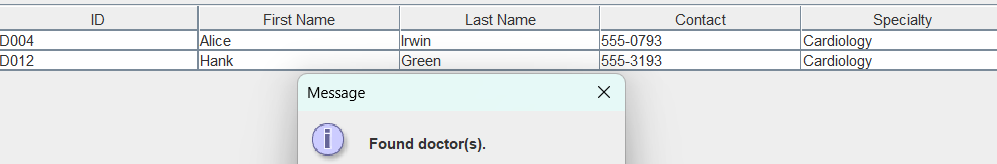
 A screenshot of a computer error

AI-generated content may be incorrect.



Find Doctor By Specialty Button





### View All Doctors Button

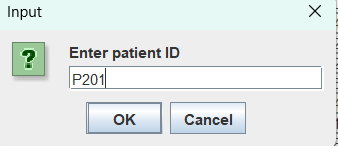
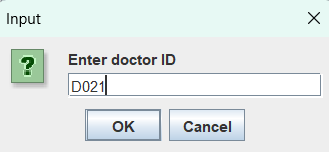
A screenshot of a computer

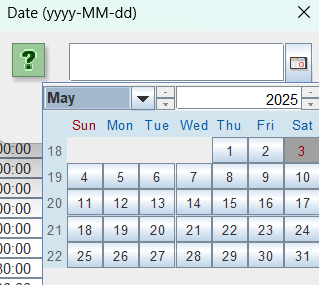
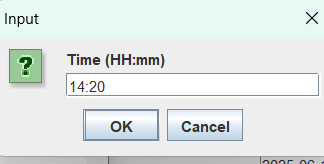
AI-generated content may be incorrect.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Appointments Tab**

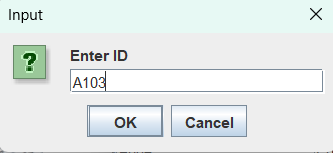
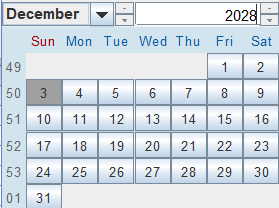
### Book Appointment Button

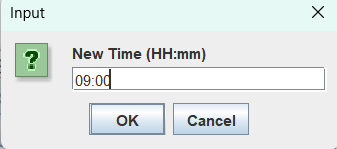
 



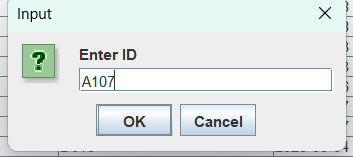
### Reschedule Appointment Button

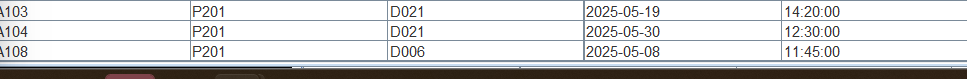




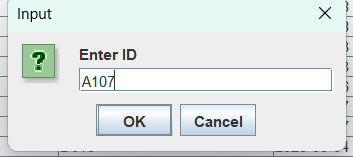
### Cancel Appointment Button

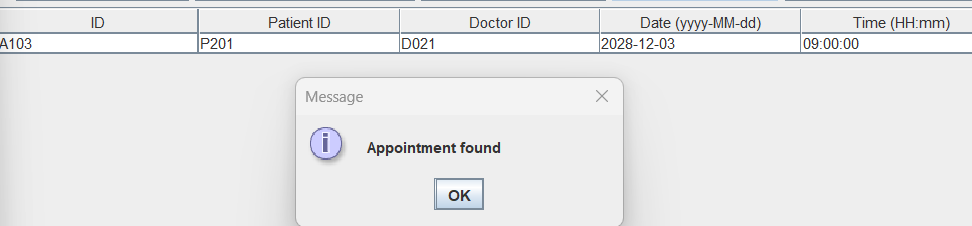
 A screenshot of a computer error

AI-generated content may be incorrect.

Appointment A105 to A107 successfully deleted.

### Find Appointment Button





### View All Appointments Button

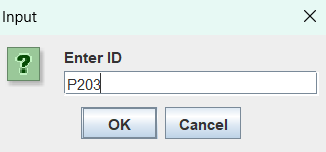
A screenshot of a computer

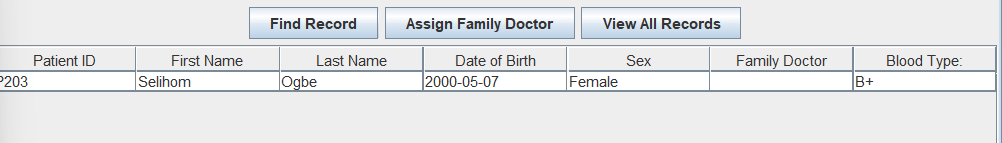
AI-generated content may be incorrect.

## **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Records Tab**

### Find Record Button



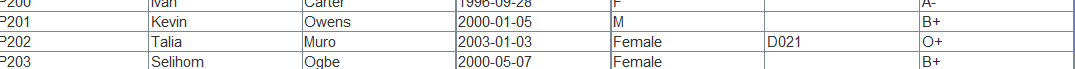
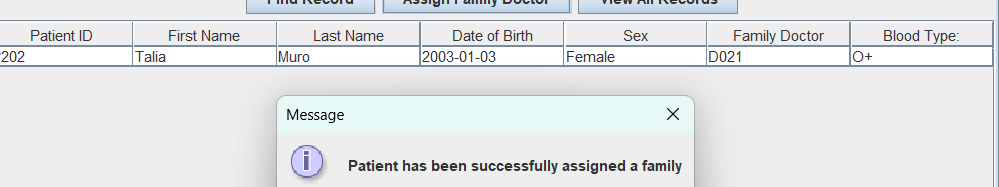


### Assign Family Doctor Button

A screenshot of a computer error

AI-generated content may be incorrect. A screenshot of a computer error

AI-generated content may be incorrect.



### View All Records Button

A screenshot of a computer

AI-generated content may be incorrect.

## **Language Selector**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Challenges

## java.lang.NullPointerException

We had a lot of trouble with this exception because our GUI’s buttons were not initialized properly. The reason we were getting this error is because we were using a GUI designer, but we also created a separate GUI without the designer so the IDE was trying to initalize the GUI made in the designer, but it had not class to controll its controls.

## java.lang.ExceptionInInitializerError

We kept getting this error when attempting to implement our internationalization. It had something to do with the IDE automatically creating a MessageBundle folder so the ResourceBundle.getBundle(BASE) was never reaching our MessageBundle.properties files. Finally, we were able to manually remove this automatically generated directory, and it worked.

## Error –preview not enable for java 22

We kept getting this error when attempting to view at what stages we had gotten with our project however we were unable to preview the gui in order to interact with it. We had to change the target language level to that which we were using (java 23) in the java compiler and rebuild the project.

## java.lang.ClassCastException: java.util.Date cannot be cast to java.sql.Date

We kept running into issues with java.util.Date and java.sql.Date not being compatible. The JDateChooser returned a util.Date that was incompatible with the database. To fix it, we had to convert all dates into sql.Dates, we cast it into a new object, and this fixed the errors for both database and the program.

## java.util.UnknownFormatConversionException: Conversion = 'F'

This was an inattention mistake from our part that we kept ignoring for the longest time, we fimply had to turn it to a lowercase ‘f’ for it work.

## [SQLITE\_ERROR] SQL error or missing database (table Patients has 9 columns but 10 values were supplied)

During our implementation, especially remote work we would have issues such as this since we did not communicate properly, it was also due to our constructors that did not involve the creation of the id to prevent duplicates. We dropped the previous table and recreated the correct schema as well as adding another constructor for it to be compatible with the database in order to fetch the values.

## [SQLITE\_CONSTRAINT\_PRIMARYKEY] A PRIMARY KEY constraint failed

This was also an inattention mistake on our part the database reads the last input values’ id it increments the new inserted value’s id and injects in the database, however since we started our reading from index 3 (to ignore the P(and redundant 0)) once we reach the number 100 the next id will be “1” which would mean it is a duplicate. To fix this we simply had to read the value from index 1 instead to only ignore P.

## Reverting changes made in git to go back to a stable version

One challenge we faced during development was accidentally making changes that broke the application. To fix this, we often had to use git reset --hard to return to a previous working commit. While this helped undo mistakes quickly, it also meant we sometimes lost newer changes, which taught us to commit more strategically and use branches to experiment safely. We did not thing about this in moments of stress but toward the end we made new branches in order to experiment the different fixes as well as to not lose data during a specific merge that forced us to force push the changes which means the other teammate would have lost their progress but thanks to the other branch it was an easy enough fix. In future projects we will definitely keep this in mind.

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Learning Outcomes

We have learned a lot about Swing and all of the GUI controls it has and how to connect them to our code. We also learned how to keep our database persistent. Through this project, we learned how to build and maintain a Java Swing application using JDBC for database interactions. We applied the MVC (Model-View-Controller) pattern to keep our code organized and consistent across different components. Working as a team, we also became comfortable using Git for version control effectively and manage changes without conflicts.