SDLC Case Study Worksheet

Project Title: Hospital Management System

Team Name: Merge Minds

Team Members and Roles: Mohammad Farooq (Project Manager)

Khubaib Ahmed(Requirement Analysis)

Sarim ul Haq(Backend Developer)

Anosha Hafeez(Designing + Frontend Development)

Muhammad Umer (tester)

1. Requirements Phase

Write 5 functional and 2 non-functional requirements for your project.

Functional Requirements:

1. _Patient Registration and Profile management.
2. Appointment scheduling and Calender
3. Biling and invoice generation
4. Medical History and records
5. Reporting and Audit Logs

Non-Functional Requirements:

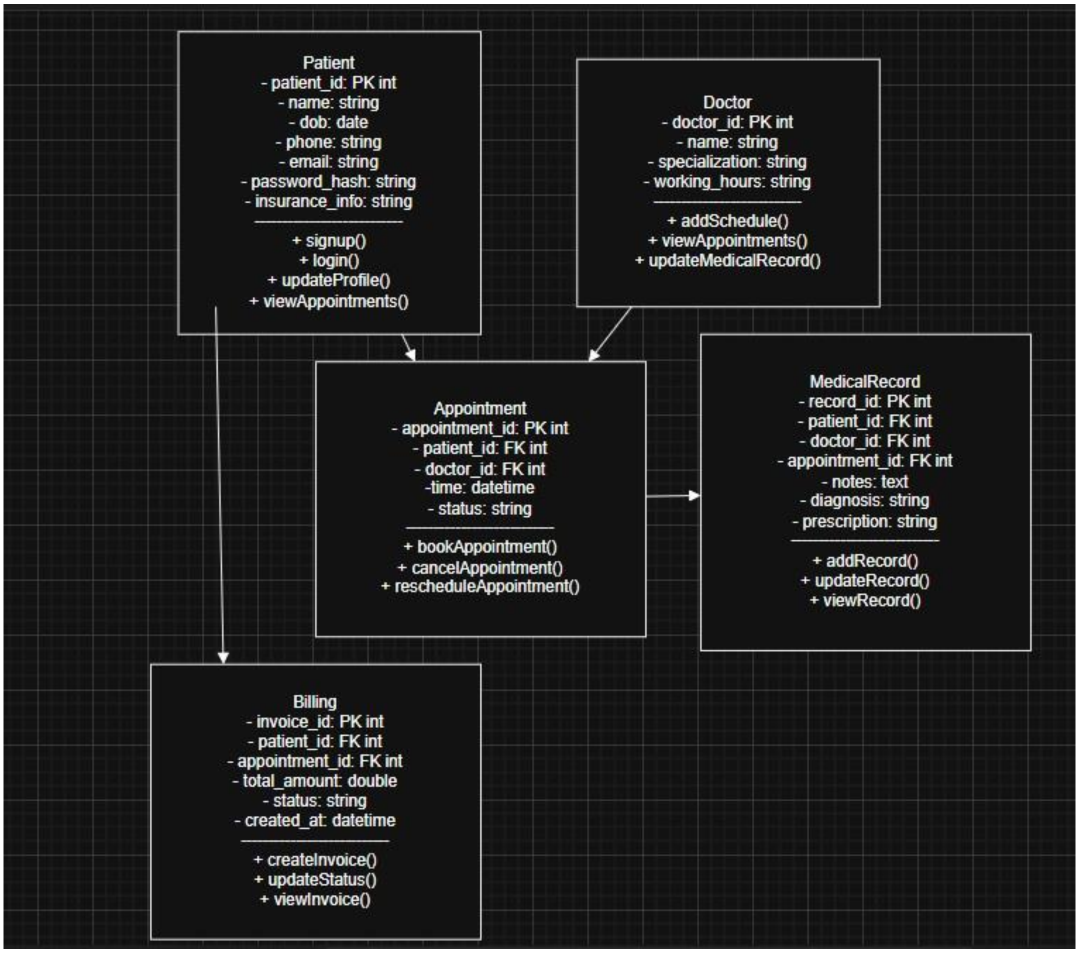
1. Security and Privacy -> Patient Data must be encrypted
2. Performance and scalibility -> The System must be able to handle the multiple users.

2. Design Phase

Draw a simple WBS (3 levels) and one UML diagram (use case/class diagram).

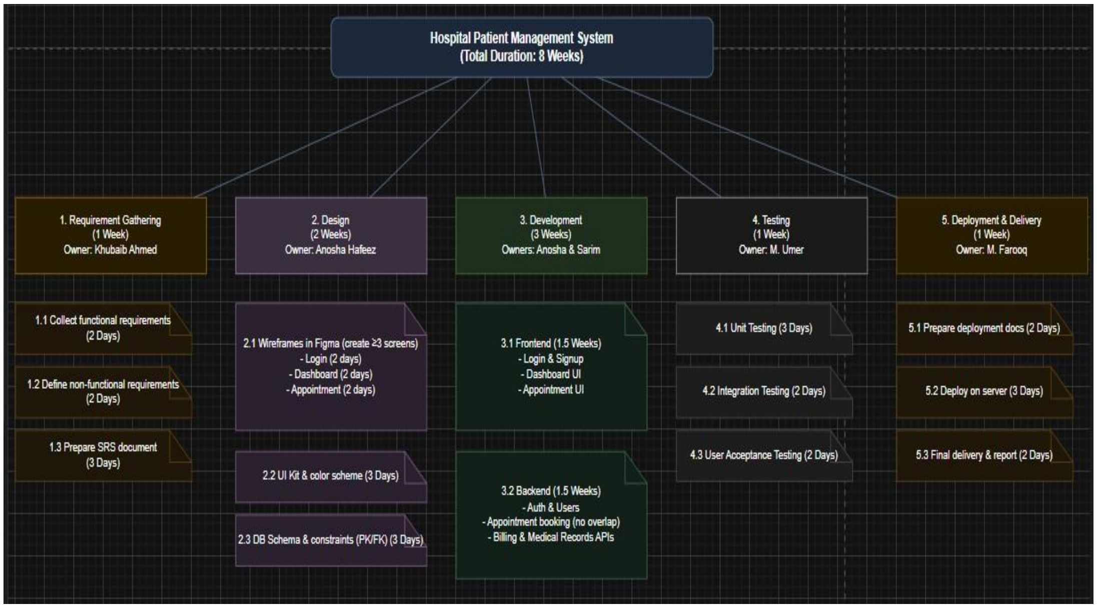
Work Breakdown Structure (WBS): (Write as list or sketch tree diagram)

UML Class Diagram:



UML Sketch (Use Case / Class Diagram/sequence Diagram): (Draw below)

WBS Tree Diagram:

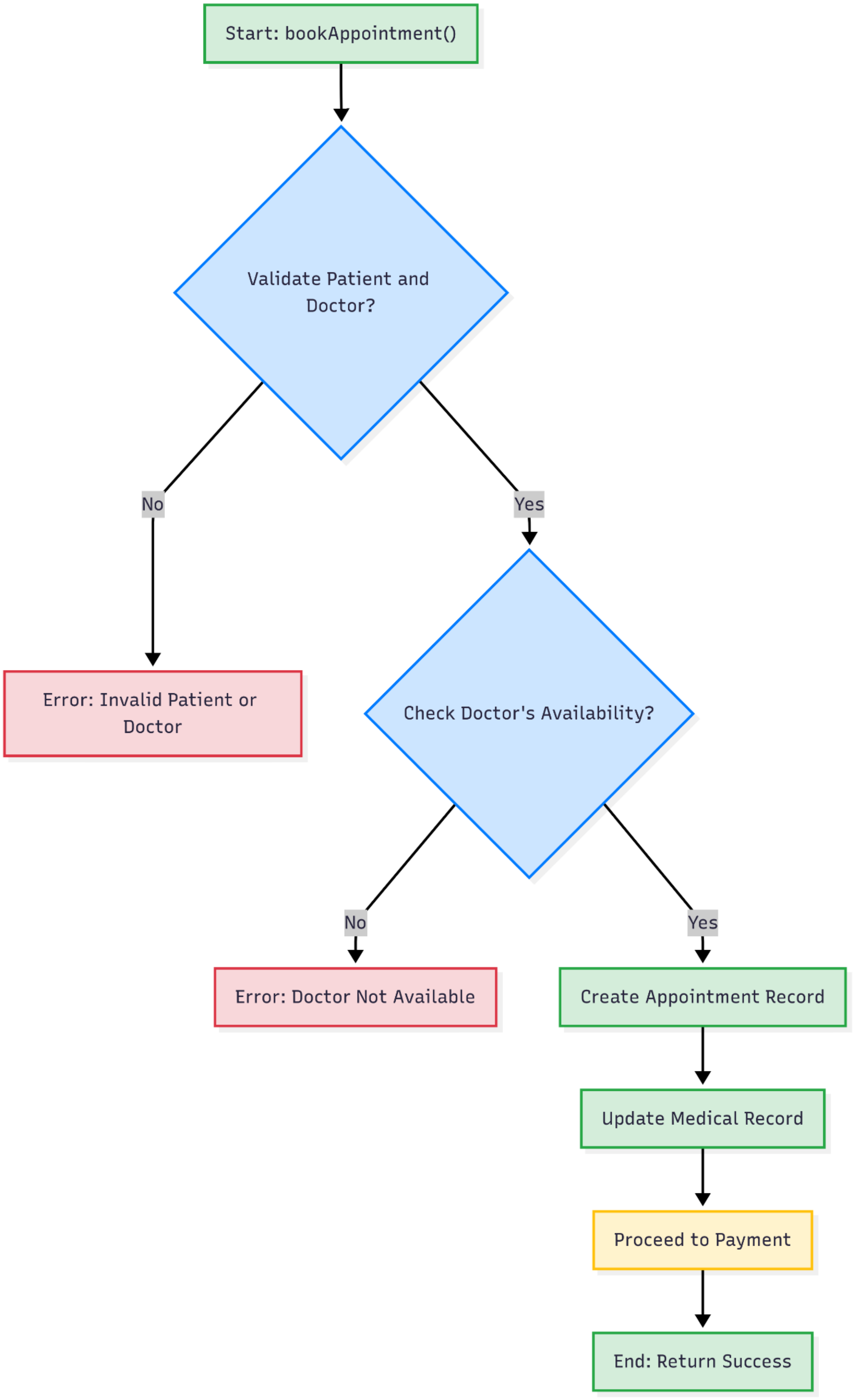

Hospital Patient Management System
(Total Duration: 8 Weeks)
1. Requirement Gathering
(1 Week)
Owner: Khubaib Ahmed
1.1 Collect functional requirements
(2 Days)
1.2 Define non-functional requirements
(2 Days)
1.3 Prepare SRS document
(3 Days)
2. Design
(2 Weeks)
Owner: Anosha Hafeez
2.1 Wireframes in Figma (create ≥3 screens)
- Login (2 days)
- Dashboard (2 days)
- Appointment (2 days)
2.2 UI Kit & color scheme (3 Days)
2.3 DB Schema & constraints (PK/FK) (3 Days)
3. Development
(3 Weeks)
Owners: Anosha & Sarim
3.1 Frontend (1.5 Weeks)
- Login & Signup
- Dashboard UI
- Appointment UI
3.2 Backend (1.5 Weeks)
- Auth & Users
- Appointment booking (no overlap)
- Billing & Medical Records APIs
4. Testing
(1 Week)
Owner: M. Umer
4.1 Unit Testing (3 Days)
4.2 Integration Testing (2 Days)
4.3 User Acceptance Testing (2 Days)
5. Deployment & Delivery
(1 Week)
Owner: M. Farooq
5.1 Prepare deployment docs (2 Days)
5.2 Deploy on server (3 Days)
5.3 Final delivery & report (2 Days)

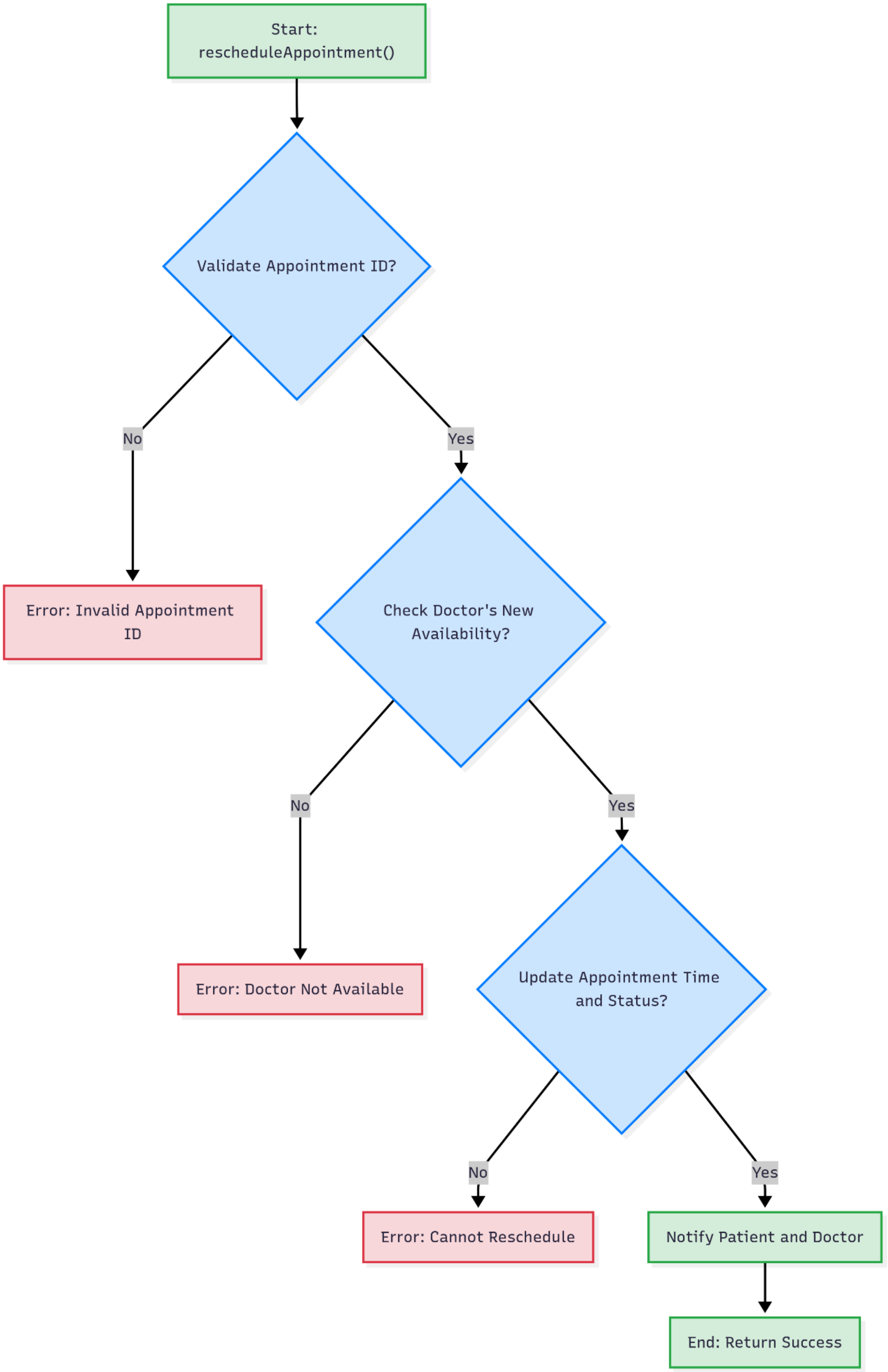
Tool Used to Draw Diagrams : Draw.io

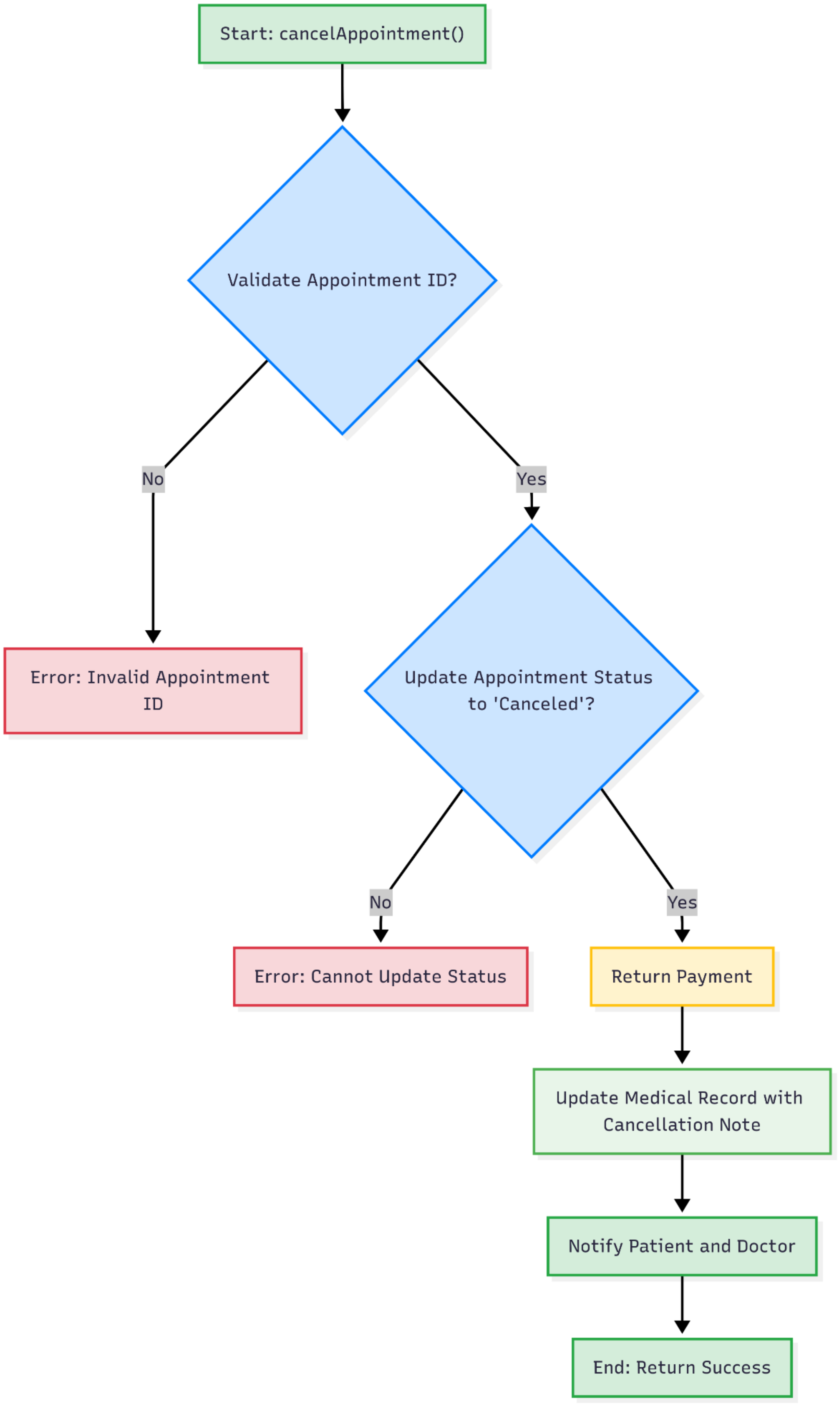
3. Backend Design

Attach you design file. And mention the tool that you used.

The tool used for these diagrams is mermaid.live


Start: bookAppointment()
Validate Patient and Doctor?
No
Yes
Error: Invalid Patient or Doctor
Check Doctor's Availability?
No
Yes
Error: Doctor Not Available
Create Appointment Record
Update Medical Record
Proceed to Payment
End: Return Success


Start:
rescheduleAppointment()
Validate Appointment ID?
No
Yes
Error: Invalid Appointment ID
Check Doctor's New Availability?
No
Yes
Error: Doctor Not Available
Update Appointment Time and Status?
No
Yes
Error: Cannot Reschedule
Notify Patient and Doctor
End: Return Success



Start: cancelAppointment()
Validate Appointment ID?
No
Yes
Error: Invalid Appointment ID
Update Appointment Status to 'Canceled'?
No
Yes
Error: Cannot Update Status
Return Payment
Update Medical Record with Cancellation Note
Notify Patient and Doctor
End: Return Success

4. Development Phase

Write pseudo-code/code for one key function in your system.

```
FUNCTION bookAppointment
 INPUT(s): patient_id, doctor_id, appointment_time
PROCESS (steps):
 1. Validate that `patient_id` and `doctor_id` are both valid and exist in the system.
2. Check the doctor's schedule to confirm that `appointment_time` is available.
 3. If all checks pass, create a new record for the appointment in the database.
4. Log the new appointment in the patient's medical record.
5. Process the payment for the service.
6. Send a confirmation message to both the patient and the doctor.
OUTPUT: A confirmation message of success or a specific error message.
END FUNCTION


FUNCTION cancelAppointment
  INPUT(s): appointment_id
  PROCESS (steps):
    1. Validate that the `appointment_id` is valid and corresponds to an existing appointment.
    2. Change the status of the appointment record to "Canceled."
    3. If a payment was made, initiate and process a refund.
    4. Add a note to the patient's medical record indicating that the appointment was canceled.
    5. Send a notification to the patient and the doctor confirming the cancellation.
  OUTPUT: A success message, or an error if the appointment ID is invalid.
```

```
END FUNCTION


FUNCTION rescheduleAppointment

  INPUT(s): appointment_id, new_time

  PROCESS (steps):

    1. Validate the `appointment_id` to ensure it is a valid, existing appointment.

    2. Check the doctor's schedule to verify that the `new_time` is available.

    3. If available, update the existing appointment record with the new time and change its
status         to "Rescheduled."

    4. Send an updated confirmation to the patient and doctor with the new details.

  OUTPUT: A success message with the rescheduled time, or an error message if the new time is
unavailable or the ID is invalid.

END FUNCTION
```

5. Testing Phase

Write 3 test cases.

Test Case ID	Description
TC-01	Validate booking of an appointment with valid patient and doctor IDs and available slot.
TC-O2	Validate rescheduling of an appointment with valid appointment ID but doctor unavailable at new time.
TC-03	Validate cancellation with invalid appointment ID.

Test Case ID	Input(s)	Expected Output	Result (Pass/Fail)
TC-01	patient_id = 123, doctor_id = 456, appointment_time = "2025-09-15 10:00"	Appointment created successfully, confirmation message sent to patient and doctor.	PASS
TC-02	appointment_id = 789, new_time = "2025-09-15 11:00" (doctor not available)	Error: Doctor Not Available	PASS
TC-03	appointment_id = "invalid123"	Error: Invalid Appointment ID	PASS

6. Reflection

1. Which SDLC phase was the most challenging? Why?

The design phase was the most difficult bcz it required translating abstract requirements into a concrete system model ensuring both database design and UI structure aligned with functional needs

2. Which SDLC model (Waterfall, Agile) best fits this project? Why?

The agile model best fits the project since it allows iterative development, continuous feedbacks and easier handling of requirement changes compared to rigid waterfall approach

3. How you determine functional and non-functional requirements?

Fuctional requirements were determined using user needs,system test cases and required features

Non functional requirements were identified by performance,security,usability and scalability expectations

7. Attachments

Along with this filled worksheet, provide supported document of design methodologies/diagrams and document of types of testing techniques.