Date

SDLC Case Study

Worksheet :-

Project Title: Hospital Patient Management System

Team name: The Merge Minds

Team Members	Roles
M. Farooq (068)	Project Manager
Anosha Hafeez (014)	Designer + Frontend dev
Khubaib Ahmed (052)	Requirement Gathering
Sarim-ul-Haq (149)	Backend Developer
M. Umer (117)	Testing

Date

1. Requirement Phase:

→ 5 functional and 2 non-functional requirements for our Project :

Functional Requirement :

→ Patient registrations & Profile management.

→ Appointment scheduling & calendar

→ Biling and invoice generation.

→ Medical history and records.

→ Reporting and audit logs

→ Role-based access → Receptionist, Doctor, Admin, Patient.

Non-functional Requirements

→ Security & Privacy → Patient data must be encrypted and accesible only to authorized users-

→ Performance and Scalibility. → The system must handle multiple users concurrently with fast response time:

2. Design Phase:-

→ UML class diagram and

→ WBS Tree diagram are designed

using Draw.io

note: Screenshots are attached and files are also uploaded in folder.

4. Development Phase:-

→ The most important & key functionality of our system is **Appointment Scheduling**

Pseudo code for Appointment scheduling.

```
                          name                      parameters
Function schedule~~Applica~~Appointment (PatientID, doctorId,
    appointment Date Time)
  Input → from users (Patients):
  → Patient ID (unique identifier of Patient)
  → doctorId (unique identifier of doctor)
  → appointment Date Time (requested date & Time).

  Process (steps):
    1. Check if doctor ID exists in System
    2. Check if PatientID exists in system
    3. Search database for existing appointment
         Where doctorId = given doctorID
         AND appointment Date Time overlaps.
    4. If conflicts exists Return the message " Doctor already
         booked . Choose another slot." (and show them available
         slots for ease).
    5. Else
         Save new appointment in database with patient ID,
         doctorId, appointment Date Time. and
         Return "Appointment Scheduled successfully".
  END functions.
```

6. Reflection:-

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

→ The Design Phase was the most Challenging because it required translating abstract requirements into a Concrete System system model, ensuring both database design and UI structure aligned with functional need.

2. Which SDLC model best fits this project? :Why?

→ The Agile model best fits this Project since it allows Iterative development, continuous feedback and easier handling of requirement changes compared to the rigid waterfall approach.

3. How do you determine functional & non-functional requirements?

* functional requirements: Determined by analyzing user needs, system use-cases and required features (login, appointment booking, search).

* Nonfunctional Requirements: Identified by considering Performance, Security, usability, reliability and scalability expectations.