Case Study Task: "SmartClinic 2.0 – Appointment & Patient Management System"

Overview :

You are part of a software development team responsible for designing **SmartClinic 2.0**, a digital solution intended to replace the clinic's existing paper-based appointment management and patient interaction workflows. This task guides you through all fundamental stages of the software development lifecycle, as covered in the training materials, with a strong emphasis on analysis, planning, system design, and documentation.

PART 1 — System Study Task

1. Study of the Current System

a. Appointment scheduling is maintained using a paper-based register.
b. Appointment rescheduling is handled through phone calls.
c. Patient reminders are managed manually.

2. Identification of Problems and Gaps

Identify at least six key issues in the existing system, such as delays in appointment handling, miscommunication, lack of appointment history, risk of record loss, inefficient tracking, and limited accountability.

Problems & Their Effects:

1. **Appointment Scheduling Conflicts**
   Appointments are frequently mixed up, resulting in multiple patients being assigned the same time slot. This leads to patient dissatisfaction and increased stress for medical staff.
2. **Limited Access to Patient History**
   Patient medical history is not readily available. If physical files are misplaced, doctors lack access to previous diagnoses, treatments, and medications.
3. **Risk of Data Loss or Damage**
   Paper-based records are vulnerable to loss, damage, or deterioration. Once a file is lost, the patient's complete medical history may be permanently unavailable.
4. **Unreliable Manual Reminders**
   Appointment reminders rely on manual follow-ups, which are prone to human error. Missed reminders result in patient no-shows and financial losses for the clinic.

5. **Excessive Time Spent on Rescheduling Calls**
   Appointment rescheduling is handled through phone calls, consuming significant receptionist time and causing delays in other administrative tasks.
6. **Lack of Reporting and Analytical Insights**
   The clinic lacks access to essential reports and analytics, including daily patient volume, no-show rates, and peak operational hours.
7. **Inadequate Patient Data Security**
   Paper files can be accessed by unauthorized individuals, posing serious privacy and confidentiality risks.

Conclusion :

The existing system is inefficient, insecure, and prone to errors. It leads to operational delays, poor patient experience, increased administrative workload, and financial losses. A digital solution is essential to improve efficiency, data security, and overall clinic operations.

PART 2 — User Objectives

Task

Interview (hypothetical) users:

- Receptionist
- Doctor
- Patients

From this, define **clear objectives** (NOT features), such as: Reduce waiting time

- Automate reminders
- Improve accuracy
- Avoid double-booking

After analyzing the needs of receptionists, doctors, and patients, the proposed system aims to:

- Minimize patient waiting time through efficient appointment scheduling.
- Eliminate double-booking of appointments by enforcing automated scheduling controls.

- Automatically send appointment reminders to patients to reduce no-shows.
- Ensure patient records are securely stored, well-organized, and easily retrievable.
- Enable fast and hassle-free appointment rescheduling.
- Provide doctors with quick access to complete patient medical history.
- Reduce the administrative workload of clinic staff through process automation.
- Enhance overall clinic service quality and patient satisfaction.

PART 3 — User Requirements
Break requirements into:
Functional Requirements
Examples:
- Book and reschedule appointments
- Send SMS/email reminders
- Maintain patient history
- Doctor availability calendar (based on the training content)

Functional Requirements (What the system must do)
1. Patient can book appointment
2. Receptionist can add, cancel, or change appointment
3. System should stop double booking
4. System should send confirmation message
5. System should send reminder message
6. Store patient details safely
7. Store patient medical history
8. Doctor can see daily appointment list
9. Doctor can add treatment note
10. Only authorized user can access data
11. Search patient by name or phone
12. Create daily appointment report

Non-Functional(How the system should behave)
1. System should be fast
2. Patient data must be safe

3. System should work on mobile also
4. System should not crash easily
5. System should be easy to use
6. Data backup should be done regularly

PART 4 — Prepare the SRS (Software Requirements Specification).
Task
Create a **mini SRS** containing:

1. Purpose
SmartClinic 2.0 is made to replace the paper system with a digital system for managing appointments and patient records easily and safely.

2. Scope
The system will:

- Book appointments
- Send reminders
- Store patient history
- Show doctor schedule
  It will NOT handle billing or insurance.

3. Definitions

- **Patient Records:** Patient personal and medical details
- **Appointment:** Fixed time for doctor visit
- **Reminder:** Message sent before appointment

4. Functional requirements
(Same as Part 3 – 12 points)

5. Non-functional requirements

(Same as Part 3 – 6 points)

6. Inputs (ex: user details, appointment details)

- Patient name, phone
- Appointment date & time
- Doctor name
- Login details

7. Outputs (confirmation, calendar view, notifications)

- Appointment confirmation
- Reminder messages
- Doctor appointment list
- Patient history report

8. Constraints (ex: clinic hours, authentication)

- Works only during clinic hours
- User must login
- Internet connection needed for reminders

PART 5 — Feasibility Study

Technical :

Do interns know the required tech (React, Node, DB)?

Yes.
The interns have basic knowledge of:

- **React** for frontend
- **Node.js** for backend
- **Database (MongoDB / PostgreSQL)** for storing data

With the help of one **project lead**, this system can be developed easily.

Required hardware/software?

Hardware Needed:

- Laptops or computers
- Internet connection

Software Needed:

- VS Code (free)
- Node.js (free)
- React (free)
- Database software (free)
- Web browser

All required tools are **free and easily available**.

Economic :

Cost of development

- Development cost is low because interns are working on it.
- Only small cost for:
- Internet
- Hosting the website
- SMS service for reminders

vs. expected return

- Fewer patients will miss appointments
- More patients will be handled daily
- Clinic will save staff time
- Clinic income will increase

Licensing tools

- React, Node, Database → Free & Open Source
- No expensive software license required
- Conclusion (Economic): FEASIBLE

Operational

Will staff adopt digital tools?

Yes.
At first they may feel small difficulty, but after using it:

- Work becomes faster
- No need for registers
- No fear of losing records

Any training required?

Yes, but only basic training for 1–2 days is enough.
Conclusion (Operational): FEASIBLE

FINAL CONCLUSION

SmartClinic 2.0 Project is:
FEASIBLE (Technically, Economically & Operationally)

This system is:

- Easy to build
- Low cost
- Useful for clinic
- Easy for staff to learn

PART 6 — Time & Resource Estimation
Task
Create a **high-level project estimation**:

- UI: 1 week
- Backend APIs: 1.5 weeks

- Database + models: 0.5 week
- Notifications: 0.5 week
- Testing + deployment: 1 week

TASK	TIME REQUIRED	WEEK
UI Design & Development	1 weeks	Week1
Backend API Development	1.5 weeks	Week 2 & Half of Week 3
Database & Models	0.5 weeks	Half of Week 3
Notifications (SMS/Email)	0.5 weeks	Week 4
Testing & Development	1 weeks	Week 5

Week 1: [UI Design & Development]

Week 2: [Backend APIs]

Week 3: [Backend APIs] [Database]

Week 4: [Notifications]

Week 5: [Testing & Deployment

Include:

Team structure (2 interns + 1 lead)

ROLE	NUMBER	RESPONSIBILITY
Project Lead	1	Planning, guidance, code review, final deployment
Intern	2	UI development, backend APIs , database,

		testing

Tools & technologies

CATEGORY	TOOLS USED
Frontend (UI)	React
Backend	Node.js
Database	MongoDB / PostgreSQL
Notifications	SMS & Email Service
Testing	Manual Testing
Deployment	Free Cloud Hosting
Code Management	GitHub
Editor	VS Code

All tools are **free, open-source, and easily available**.

FINAL CONCLUSION

The project can be completed in 5 weeks
2 interns + 1 lead are enough for this system
The tools are free and suitable
The plan is realistic and manageable

PART 7 — Final Workflow Diagram

Design the final workflow for SmartClinic 2.0:
Patient → Appointment Booking → Confirmation → Reminder → Consultation → Follow-up
Deliverable
A simple workflow diagram (can be drawn in Figma, PowerPoint, or even hand-drawn & scanned).

SMARTCLINIC 2.0