# Hospital OPD Queue Management & SDLC Activity Sheets

SDLC ACTIVITY SHEETS

**Hospital Queue Management and OPD Guidance System**

**ACTIVITY SHEET 1 - Requirements Gathering (SRS Development)**

Project Title: Hospital Queue Management and OPD Guidance System

Task 1: Stakeholder Interview

Identified Stakeholders:

1. Patients

2. Doctors

3. Reception / OPD Staff

4, Hospital Administrator

5. System Developer

Interview Questions:

1, How long do patients usually wait in OPD queues?

2. Do patients face confusion in identifying correct OPDs?

3. Should the system show real-time queue status?

4. ls token-based appointment helpful for patients?

5. Should elderly and emergency patients get priority?

6. Do you want digital display boards for OPD guidance?

7. Should SMS notifications be sent to patients?

8. Is multilingual support required?

9. How important is reducing waiting time?

10.Should Patient History be stored for future visits?

**ACTIVITY SHEET 2 — System Design (High-Level & Low-Level)**

**Project Title:**

**Smart OPD Queue Management & Navigation System**

---

## Problem Statement

Hospitals face long waiting queues and confusion regarding OPD locations, doctor availability, and appointment timings. This system aims to reduce waiting time and confusion using digital queue management.

---

## Task 1: System Architecture Diagram

Components include:

- Patient Mobile App
- Admin Web App
- Backend Server
- Database
- Notification Service
- Hospital Information System

The patient books a token, the backend manages queues, and notifications are sent in real time.

---

## Task 2: UI Wireframes

Screens include:

- Home Screen
- Login/Signup Screen
- OPD Queue Status Screen
- Settings/Profile Screen

These screens help patients book tokens, track queues, and navigate OPDs easily.

---

### Task 3: Database Design (ERD)

Entities include:

- Patient
- Doctor
- OPD
- Appointment
- Queue Token
- Admin/Staff

Relationships connect patients with appointments, doctors with OPDs, and queue tokens with OPDs and patients.

---

### Outcome

The system reduces OPD confusion, minimizes long queues, and improves overall hospital efficiency.

## ACTIVITY SHEET 3 – Development Phase

### Task 1: Break Work Into Tasks

**Fill the development backlog:**

| Task ID | Feature | Description | Assignee | Status |
|---|---|---|---|---|
| T1 | User Interface & Menu Design | Created basic menu for booking and managing appointments. | Team Member 1 | Done |
| T2 | Appointment Booking | Implemented booking using patient details and time slot. | Team Member 2 | Done |
| T3 | Slot Check | Added logic to avoid double booking. | Team Member 3 | In Progress |
| T4 | View & Cancel Appointments | Display and cancel booked appointments. | Team Member 4 | To Do |
| T5 | Testing | Tested and fixed errors. | Team Member 5 | To Do |

---

### Task 2: Code Walkthrough Notes

**Features implemented:**
Basic menu and appointment booking module.

**Blockers faced:**
Time slot overlap issue.

**Solutions / Next steps:**
Added validation checks and will complete cancellation and testing.

---

### Deliverable: Updated backlog + coded feature demo

- Working appointment booking module demonstrated
- Basic user interface for scheduling appointments completed
- Initial validation implemented to avoid time slot conflicts

## ACTIVITY SHEET 4 – Testing Phase

### Task 1: Create Test Cases

| Test Case ID | Description | Steps | Expected Result | Actual Result | Status (P/F) |
|---|---|---|---|---|---|
| TC1 | Book appointment | Choose doctor and date, then book. | Appointment is booked successfully | Appointment booked | P |
| TC2 | Book without date | Choose doctor but no date, then book. | Error message shown | Error message shown | P |
| TC3 | Cancel appointment | Open appointment and cancel | Appointment is cancelled | Appointment cancelled | P |

---

### Task 2: Bug Report

| Bug ID | Description | Severity | Steps to Reproduce | Screenshot | Status |
|---|---|---|---|---|---|
| B1 | No email after booking | Medium | 1. Book appointment<br>2. Check email | Not attached | Open / Fixed |
| B2 | App crashes on past date | High | 1. Open app<br>2. Select past date | Not attached | Open / Fixed |

---

### Task 3: Test Summary

- Total test cases: **3**
- Passed: **3**

## ACTIVITY SHEET 5 – Deployment & Release Notes

### Task 1: Deployment Checklist

✓ Tick as completed:

- ✓ Code merged
- ✓ Database configured
- ✓ Environment variables set
- ✓ Build successful
- ✓ Final testing done
- ✓ Version tagged (v1.0)

---

### Task 2: Release Notes

**Release Version: v1.0**

**Features Included:**

- Allows patients to register, log in, and book appointments with doctors
- Enables doctors/admin to view, approve, reschedule, or cancel appointments
- Appointment scheduling based on available dates and time slots
- Patient details and appointment data stored securely in the database
- User-friendly interface for easy navigation and booking

**Known Issues:**

- Email/SMS notifications for appointment confirmation are not yet implemented
- Limited error handling for invalid input in some forms

**Next Update Goals:**

- Add email and SMS notification system for appointment reminders
- Implement online payment integration
- Improve UI design and add role-based access control

**Deliverable: Deployment & Release Document completed successfully**

**ACTIVITY SHEET 6 – Maintenance & Reflection**

**Patch ID: P1**

**Issue: Task 1 – Fix & Patch**

**Issue:**
**Appointment time clashes for multiple patients**

**Root Cause:**
**No real-time slot availability check in the scheduling module**

**Fix Implemented:**
**Added slot validation and auto-blocking of booked time slots**

**Status:**
**Done**

---

**Task 2: Team Retrospective**

**What worked well?**

- **Team coordination was good during issue identification.**
- **Appointment booking flow was easy to understand and use.**
- **Testing helped detect scheduling conflicts early.**

**What needs improvement?**

- **Better validation for peak-hour appointments.**
- **Faster update of appointment status after booking.**
- **More user feedback messages.**

**What will we change next time?**

- **Implement real-time notifications for appointment confirmation.**
- **Add automated testing for scheduling conflicts.**
- **Improve UI for better patient experience.**

---

**Deliverable: Final Reflection**

**The maintenance phase helped us identify and fix critical scheduling issues.**

**By applying proper patch management, the Hospital Appointment Scheduler became more reliable and efficient. The retrospective allowed the team to reflect on improvements and plan better enhancements for future iterations.**