**Software Design Specification (SDS)**

**Smart Queue Management System (SQMS2)**

**1. System Architecture**

* **MVC Pattern**: Flask-based
  + models.py – ORM models (SQLAlchemy)
  + routes.py – All views and route logic
  + templates/ – Jinja2 HTML templates
  + simulation.py – Optional simulation module
  + main.py – Entry point
  + \_\_init\_\_.py – App factory

**2. Database Schema**

**Tables:**

* **Patient:**  
  id (PK), patient\_id, name, phone, category, registered\_at
* **Doctor:**  
  id (PK), name, is\_available
* **Queue:**  
  queue\_id (PK), patient\_id (FK), doctor\_id (FK), priority\_level, session\_end\_time, arrival\_time, served
* **Receipt:**  
  receipt\_id (PK), patient\_id (FK), doctor\_id (FK), estimated\_wait, created\_at
* **Log:**  
  log\_id (PK), patient\_id (FK), action, timestamp

**3. UI Design**

* **Homepage:** Check-in form and dynamic receipt
* **Admin Dashboard:**
  + Filter, search, serve buttons
  + Realtime refresh + scroll-to-top
  + Dark mode
* **Doctor Dashboard:**
  + View assigned patients
  + Serve button
  + Personalized filtering
* **Logs Page:**
  + Search, reset, export (CSV, PDF, Excel)

**4. Key Modules & Logic**

* **Doctor Assignment Logic:**  
  Assign least busy doctor using SQL COUNT() ordering
* **Simulation Logic:**
  + Threads simulate check-ins
  + Doctor auto assignment
  + Patients auto served after delay
* **Export Functions:**
  + PDF: FPDF
  + Excel: openpyxl
  + CSV: native CSV

**5. Security Features**

* Session-based admin/doctor authentication
* Route protection via @login\_required logic
* HTML input sanitization
* Error handling with get\_or\_404, try/except blocks