1) SRS – DocDesk Web Application

1.1 Purpose

DocDesk is a clinic management web application to:

- Register patients (new/revisit)
- Capture pre-consultation values
- Manage doctor queue using token numbers
- Record consultation details (diagnosis, investigation, prescription, notes, fee)
- Print consultation summary for pharmacy
- Maintain master tables (admin CRUD)
- Generate revenue & consultation reports

1.2 Scope

Modules

1. Authentication & Role-based Access
2. Reception Module (registration + pre-consultation)
3. Doctor Module (today queue + consultation + print)
4. Admin Module (masters + users + reports)
5. Reporting Module (daily/monthly/yearly revenue, doctor-wise, etc.)

1.3 User Roles

1. **Reception**
    - Register new patient
    - Revisit existing patient using Patient ID
    - Generate token number for the day
    - Enter/edit pre-consultation values
2. **Doctor**
    - View today’s token queue
    - Open patient consultation page
    - Edit pre-consultation values
    - Enter consultation details + fee
    - Select multiple diagnosis/investigation/prescription
    - Save & print consultation summary
3. **Admin**

    - Full access (masters CRUD, user management)
    - Reports (revenue, consultations, etc.)

---

2) Functional Requirements

2.1 Authentication & Authorization

FR-1 Users must login with username/password
FR-2 Role-based menu access:

- Reception → Registration, Revisit, Today Tokens, Pre-consultation
- Doctor → Today Queue, Consultation, Print
- Admin → Masters, Users, Reports, All access
  FR-3 Session timeout + secure logout
  FR-4 Password hashing (BCrypt)

---

3) Reception Module Requirements

3.1 Patient Registration (New)

FR-R1 Reception can register a patient with:

- Patient Name (required)
- Mobile Number (required, validation)
- Sex (Male/Female/Other)
- Age (optional if DOB exists)
- DOB (optional if age exists)
- Place/Address (optional/required based on clinic)
  FR-R2 After saving, generate a **unique patient_id**:
- Format suggestion: PYYYYNNNN (e.g., P20260001) or numeric auto-id + display code
  FR-R3 Patient mobile must support duplicates (family members may share), but system should show possible matches.

3.2 Patient Revisit

FR-R4 Reception can search patient by:

- Patient ID (fast)
- Mobile / Name (optional feature)
  FR-R5 For revisit, create a new **visit record** (token assigned).

3.3 Token & Queue Generation

FR-R6 For every registration/revisit, generate **token number** for that date:

- Token resets daily per clinic/doctor (choose rule; typical: daily global)
  FR-R7 Reception can view today token list and status:
- Waiting / In Consultation / Completed / Cancelled

3.4 Pre-Consultation Values (Dynamic)

FR-R8 Reception can enter pre-consultation values (e.g., BP, Weight, Temperature, Sugar, Pulse).
FR-R9 These pre-consultation fields must load dynamically from DB tables (Admin configurable).
FR-R10 Pre-consultation supports:

- Numeric fields (e.g., weight)
- Text fields (e.g., symptoms brief)
- Dropdown options (e.g., blood group)
- Units (kg, mmHg, °C)
  FR-R11 Reception can edit values before doctor completes consultation.

---

4) Doctor Module Requirements

4.1 Today’s Queue List

FR-D1 Doctor can view today’s patient list ordered by token number.
FR-D2 Each token shows:

- Token No
- Patient ID
- Patient Name, Age/Sex
- Visit status
- Registration time
- Pre-consultation completion flag

4.2 Consultation Page

FR-D3 Doctor opens a token → consultation page with:

- Patient demographic details
- Pre-consultation values (editable)
- Chief complaints (free text)
- Internal notes (private)
- Consultation fee (number)
- Diagnosis selection (multi-select from master)
- Investigation selection (multi-select from master)
- Prescription selection (multi-select from master) with dosage instructions

4.3 Diagnosis / Investigation / Prescription

FR-D4 Doctor can select multiple items from master tables:

- Diagnosis (ICD optional)
- Investigation (lab tests)
- Prescription (medicine)
  FR-D5 Prescription supports:
- Dose (e.g., 1-0-1)
- Duration (e.g., 5 days)
- Frequency
- Before/After food
- Remarks
  (You can keep it simple first and expand later.)

4.4 Save, Print & Pharmacy Slip

FR-D6 On save:

- Mark visit as Completed (or keep "Completed" after print)
- Generate printable consultation summary (PDF/HTML print)
  FR-D7 Printout includes:
- Clinic name/logo (optional)
- Patient ID, name, age/sex, visit date
- Pre-consultation values
- Chief complaints
- Diagnosis
- Investigations
- Prescription list
- Fee
- Doctor signature line

---

5) Admin Module Requirements

5.1 Master Management (CRUD)

FR-A1 Admin can manage:

- Pre-consultation field definitions
- Diagnosis master

- Investigation master
- Prescription master
- (Optional) dosage templates, unit masters, categories

5.2 User Management

FR-A2 Admin can create users:

- Reception users
- Doctor users
- Admin users
  FR-A3 Activate/deactivate users and reset passwords

5.3 Reports

FR-A4 Reports required:

1. **Today Consultation Report**
   - Total visits, completed visits, revenue
2. **Monthly Consultation Report**
   - Day-wise totals + revenue
3. **Yearly Consultation Report**
   - Month-wise totals + revenue
4. **Doctor-wise Revenue Report** (if multiple doctors)
5. **Diagnosis-wise Count Report**
6. **Investigation-wise Count Report**
   FR-A5 Reports filter:

- Date range
- Doctor
- Status (completed/cancelled)

---

6) Non-Functional Requirements

6.1 Security

- BCrypt password hashing
- JWT or Session-based auth (JWT recommended for React)
- Role-based authorization at API level
- Audit logs for critical actions (edit/delete)

6.2 Performance

- Today queue must load within 2 seconds for normal clinic usage
- Pagination for reports and patient search

6.3 Availability & Backup

- Daily DB backup
- Restore procedure documented

6.4 Usability

- Fast reception workflow (minimal clicks)
- Doctor consultation page optimized for quick selection (searchable multi-select)

8) Database Architecture (MySQL)

Below is a clean, scalable schema. (You can implement with JPA/Hibernate easily.)

8.1 Core Tables

users

Stores all logins (admin/reception/doctor).

- id (PK)
- username (unique)
- password_hash
- role ENUM('ADMIN','RECEPTION','DOCTOR')
- full_name
- mobile
- is_active TINYINT(1)
- created_at
- last_login_at

patients

- id (PK, auto)
- patient_code (unique) → generated patient id like P20260001
- name
- mobile
- sex ENUM('M','F','O')
- age INT NULL
- dob DATE NULL

- place VARCHAR
- created_at
- updated_at

Note: If DOB is available, age can be calculated on UI, but keeping age helps if DOB unknown.

8.2 Visit / Token / Queue

visits

One row per consultation day.

- id (PK)
- visit_date DATE (indexed)
- token_no INT (indexed)
- patient_id (FK → patients.id)
- doctor_id (FK → users.id, role=DOCTOR) *(optional if single doctor, but recommended)*
- status ENUM('WAITING','IN_CONSULTATION','COMPLETED','CANCELLED')
- registered_by (FK → users.id, reception)
- registered_at DATETIME
- completed_at DATETIME NULL

Unique constraint suggestion: (visit_date, token_no, doctor_id) OR (visit_date, token_no) based on your rule.

8.3 Pre-Consultation (Dynamic Fields)

preconsult_fields

Admin configurable fields.

- id (PK)
- field_key (unique) e.g., 'bp', 'weight'
- label e.g., 'Blood Pressure'
- input_type ENUM('NUMBER','TEXT','DROPDOWN','BOOLEAN')
- unit VARCHAR NULL (kg, mmHg)
- is_active TINYINT(1)
- sort_order INT
- created_at

preconsult_field_options

Only for DROPDOWN types.

- id (PK)
- field_id (FK → preconsult_fields.id)
- option_value
- sort_order
- is_active

visit_preconsult_values

Stores values for each visit (not patient lifetime).

- id (PK)
- visit_id (FK → visits.id)
- field_id (FK → preconsult_fields.id)
- value_text VARCHAR NULL
- value_number DECIMAL(10,2) NULL
- value_bool TINYINT(1) NULL
- updated_by (FK → users.id)
- updated_at

Unique constraint: (visit_id, field_id)

8.4 Consultation Details

consultations

One row per visit completion.

- id (PK)
- visit_id (FK → visits.id, unique)
- chief_complaints TEXT
- internal_notes TEXT
- consultation_fee DECIMAL(10,2) DEFAULT 0
- created_by (FK → users.id, doctor)
- created_at
- updated_at

8.5 Masters (Admin CRUD)

diagnosis_master

- id (PK)
- name (unique)
- code VARCHAR NULL (ICD optional)
- is_active
- created_at

investigation_master

- id (PK)
- name (unique)
- category VARCHAR NULL
- is_active
- created_at

prescription_master

Medicine master.

- id (PK)
- medicine_name (unique)
- strength VARCHAR NULL (500mg)
- default_instruction VARCHAR NULL
- is_active
- created_at

8.6 Many-to-Many for Consultation Selections

consultation_diagnosis

- id (PK)
- consultation_id (FK → consultations.id)
- diagnosis_id (FK → diagnosis_master.id)

Unique: (consultation_id, diagnosis_id)

consultation_investigations

- id (PK)
- consultation_id (FK → consultations.id)

- investigation_id (FK → investigation_master.id)

Unique: (consultation_id, investigation_id)

consultation_prescriptions

Stores selected medicines + instructions.

- id (PK)
- consultation_id (FK → consultations.id)
- medicine_id (FK → prescription_master.id)
- dosage VARCHAR NULL (e.g., 1-0-1)
- duration_days INT NULL
- timing ENUM('BEFORE_FOOD','AFTER_FOOD','ANY') NULL
- remarks VARCHAR NULL

Unique: (consultation_id, medicine_id) *(or allow duplicates by removing unique if needed)*

8.7 Optional but Recommended

audit_logs

- id (PK)
- user_id (FK → users.id)
- action VARCHAR (e.g., "UPDATE_PATIENT", "DELETE_DIAGNOSIS")
- entity_type VARCHAR
- entity_id BIGINT
- old_value JSON NULL
- new_value JSON NULL
- created_at

clinic_settings

- id (PK)
- setting_key UNIQUE
- setting_value

Example: token reset rule, clinic name, print header/footer.

---

9) ER Relationship Summary (Simple)

- patients (1) → (M) visits

- visits (1) → (1) consultations
- visits (1) → (M) visit_preconsult_values
- consultations (1) → (M) consultation_diagnosis → diagnosis_master
- consultations (1) → (M) consultation_investigations → investigation_master
- consultations (1) → (M) consultation_prescriptions → prescription_master
- users (doctor/reception/admin) relate to visits/consultations/audits