PRODUCT REQUIREMENTS DOCUMENT (PRD)

Product Name (Working)

MediQueue – Lightweight Digital Queue Management for Hospitals

---

1. Technology Orientation (Sprint Theme)

Tech Stack

- **Frontend:** Flutter (Dart)
- **Backend:** Firebase
  - Firebase Authentication
  - Cloud Firestore
  - Firebase Storage
  - Firebase Cloud Messaging (FCM)
  - Cloud Functions
- **Design:** Figma
- **Maps & Location:** Google Maps SDK
- **State Management:** Provider / Riverpod

Real-World Application

Build a **mobile-first digital queue & appointment management system** for Tier-2/3 hospitals and clinics with:

- QR-based patient onboarding
- Real-time queue updates
- Notifications
- Simple admin panel for doctors/reception

---

2. Problem Statement

In Tier-2/3 cities:

- Patients wait in long physical queues
- No transparency of waiting time
- Manual registers cause inefficiency

- Hospitals can’t afford heavy HIS software
- Poor internet reliability

---

3. Product Vision

> Create a **PhonePe-like lightweight platform for OPD queue management** that any clinic can adopt in under 5 minutes using just a smartphone and QR code.

---

4. Goals & Objectives

MVP Goals

- Digitize OPD queue
- Reduce crowding
- Provide real-time queue visibility
- Simple UI for low-tech users
- No expensive hardware

Success Metrics

- Clinic onboarding < 10 minutes
- 60% patients use QR
- Reduce waiting time by 30–40%
- Doctor adoption rate
- App stability

---

5. Target Users

Primary Users

- Doctors
- Receptionists
- Small clinics
- Diagnostic centers

Secondary Users

- Patients (smartphone users)

6. Key Features (MVP Scope)

6.1 Authentication (Firebase Auth)

- Email & password login for staff
- Session persistence
- Logout
- Role-based access (Doctor / Reception)

6.2 Clinic Setup Module

- Create clinic profile
- Add doctors
- Set OPD timings
- Generate QR code

6.3 Patient Queue Module

- Scan QR
- Enter name + phone
- Join queue
- Get token number
- View estimated wait time

6.4 Queue Management Dashboard

- View current queue
- Next / Skip / Complete patient
- Manual patient entry
- Reset daily queue

6.5 Notifications

- SMS / Push notifications
- Events:
    - Queue joined
    - 2 patients remaining
    - Turn now

6.6 Real-time Sync (Firestore)

- Live updates across devices
- Snapshot listeners

6.7 Maps & Location (Optional MVP+)

- Show clinic location
- Patient navigation

6.8 UI/UX

- Responsive UI
- Dark mode
- Consistent theme
- Error states
- Loaders
- Empty states

---

7. Non-Functional Requirements

- Low bandwidth support
- Works on Android devices
- Secure data storage
- High availability
- Fast load times
- Scalable architecture

---

8. Out of Scope (Future Features)

- Payments
- EMR
- Insurance
- Lab reports
- AI diagnosis
- Video consultation
- ABHA integration

🏗️ 9. High-Level Design (HLD)

Architecture

Flutter App (Client)
⬇️
Firebase Authentication
Cloud Firestore (DB)
Firebase Storage
Cloud Functions
FCM (Notifications)
Google Maps SDK

Main Modules

1. Authentication Module
2. Clinic Management Module
3. Queue Management Module
4. Notification Module
5. UI Module
6. Maps Module
7. State Management Module

🧱 10. Low-Level Design (LLD)

Core Classes

User

None

```
User {
  uid
  name
  role (doctor/reception)
  clinicId
}
```

Clinic

```
None
Clinic {
  clinicId
  name
  address
  location
}
```

Doctor

```
None
Doctor {
  doctorId
  clinicId
  name
  timing
}
```

QueueToken

```
None
QueueToken {
  tokenId
  patientName
  phone
  status (waiting, skipped, completed)
  createdAt
}
```

Services

- AuthService
- FirestoreService
- NotificationService
- LocationService
- StorageService

📱 11. UI Screens

1. Splash Screen
2. Login / Signup
3. Clinic Setup Screen
4. Doctor List Screen
5. QR Code Screen
6. Patient Join Queue Screen
7. Queue Dashboard
8. Token Status Screen
9. Map Screen
10. Settings Screen

🔄 12. Data Flow

1. User logs in → Firebase Auth
2. Creates clinic → Firestore
3. Patient scans QR → Web/App screen
4. Token added → Firestore
5. Queue updated → snapshot listener
6. Notification sent → Cloud Function

🔐 13. Security

- Firebase Authentication
- Firestore rules by clinicId
- No medical records stored
- Data auto-delete after 24 hours
- HTTPS enforced

🎨 14. Design Principles (Figma → Flutter)

- Responsive layouts
- Widget-based UI

- Material Design
- Adaptive layout using MediaQuery & LayoutBuilder
- Consistent theme & colors
- Dark mode support

🧪 15. Testing

- Emulator testing
- Physical device testing
- Error handling
- Empty states
- Network failure cases

🚀 16. Deployment

- Build APK/App Bundle
- Play Store deployment
- Firebase production environment
- Monitoring logs & crashes

📅 17. Project Roadmap (Sprint-Based)

Sprint 1

- Flutter setup
- UI screens
- Navigation
- Theme

Sprint 2

- Firebase Auth
- Firestore integration
- CRUD queue flow

Sprint 3

- Notifications
- Maps
- Cloud Functions
- Testing

Sprint 4

- Deployment
- Feedback
- Iteration

📚 18. Alignment With Your Topic List

Your provided syllabus topics map naturally to this PRD:

Topic Group	Product Module
Flutter UI (2.3–2.25)	UI screens, widgets, navigation
Firebase (2.26–2.39)	Auth, Firestore, notifications
Maps (2.40–2.41)	Clinic location
CRUD Flow (2.42)	Queue management
State Management (2.43)	App state
Forms (2.44)	Patient & clinic forms
Navigation (2.45)	Bottom nav
Themes (2.46)	Dark mode
Error handling (2.47)	Loaders & empty states
Testing (2.48)	QA
Release (2.49–2.50)	Play Store

🏁 19. Final MVP Definition

The MVP will allow:
✅ Clinic to register
✅ Doctor to create OPD queue
✅ Patient to join via QR
✅ Live queue updates
✅ Notifications
✅ Secure authentication
✅ Responsive UI
✅ Play Store deployable app
