**Kidney Health Tracking App - Project Documentation**

**1. Software Development Life Cycle (SDLC)**

**1.1 Planning**

* **Objective**: Design an app to help users with kidney conditions manage water intake and diet.
* **Stakeholders**: Patients, dietitians, nephrologists, developers, testers.
* **Tools**: Flutter/React Native, Firebase/Django, Figma, GitHub.

**1.2 Requirement Analysis**

* Functional: Water tracker, diet plan suggestions, reminders, notifications, user profile.
* Non-functional: Responsive UI, high availability, easy navigation, secure data.
* Technical: Mobile app (Android/iOS), cloud backend.

**1.3 Design**

* UI/UX wireframes and mockups (created in Figma).
* Architecture: MVVM (Model-View-ViewModel), REST API.
* Database: Firestore or MySQL.

**1.4 Development**

* Setup of environment (IDE, SDKs).
* Frontend: UI components, navigation, and logic.
* Backend: API endpoints, user data handling, reminder logic.

**1.5 Testing**

* Unit tests (data validation, input checks).
* Integration tests (API and database).
* User Acceptance Testing (feedback from real patients).

**1.6 Deployment**

* Android: Google Play Store
* iOS: Apple App Store
* Hosting: Firebase, AWS, or Heroku

**1.7 Maintenance**

* Bug fixing, updates, security patches, adding new features (e.g., wearable integration).

**2. Software Requirements Specification (SRS)**

**2.1 Introduction**

* Purpose: Track water intake and suggest kidney-friendly meals.
* Scope: Individual users with kidney conditions.
* Definitions: CKD – Chronic Kidney Disease; UI – User Interface.

**2.2 Functional Requirements**

* R1: User login and registration
* R2: Track and log water intake
* R3: Provide meal suggestions
* R4: Set and send reminders
* R5: View daily and weekly reports

**2.3 Non-Functional Requirements**

* NFR1: App must be accessible to users 60+
* NFR2: Response time < 2 seconds
* NFR3: 99% uptime

**3. Architecture Design Document (ADD)**

**3.1 Overview**

* Client-Server architecture
* Mobile frontend, cloud backend

**3.2 Components**

* Frontend: Built in Flutter/React Native
* Backend: Firebase/Django with REST APIs
* Database: Firestore (NoSQL) or MySQL (Relational)
* Notification Service: Firebase Cloud Messaging

**3.3 Diagrams**

* UI Mockup (see design above)
* Use Case Diagram: Login, Log Water, View Plan, Get Reminder
* Sequence Diagram: User action -> API -> Database -> UI update

**4. Test Plan**

**4.1 Testing Types**

* Unit Testing: Logic and UI inputs
* Integration Testing: API + Frontend/Backend
* System Testing: Full user flow
* UAT: End-user testing with real patients

**4.2 Test Cases**

* TC1: Login validation
* TC2: Water log increments
* TC3: Reminder firing
* TC4: Meal plan display correctness

**5. User Manual**

**5.1 Setup**

* Download from Play Store/App Store
* Create account or login

**5.2 Daily Use**

* Open app to view dashboard
* Tap 'Log Water' every time you drink
* Tap 'Meal Plan' to see what to eat
* Wait for reminder alerts throughout the day

**5.3 Tips**

* Customize reminder times in settings
* Review weekly report with your doctor

**6. Deployment Plan**

**6.1 Pre-deployment**

* Finalize tests, clean code
* Prepare App Store listings

**6.2 Deployment Steps**

* Build release APK/IPA
* Submit to stores
* Monitor analytics and crash reports

**6.3 Post-deployment**

* Respond to feedback
* Schedule regular updates

**7. Maintenance and Support Plan**

* Weekly check for bug reports
* Monthly review of user analytics
* Bi-annual update cycles
* Emergency hotfix capability

**8. Future Enhancements**

* AI-based meal suggestion system
* Chatbot for health Q&A
* Sync with fitness trackers
* Multi-language support

End of Documentation