Diabetic Neuropathy Monitoring Smart Shoe - Project File Structure

Root Directory Structure

```
diabetic-smart-shoe/
├── README.md
├── LICENSE
├── .gitignore
├── docker-compose.yml
├── docs/
├── hardware/
├── firmware/
├── mobile-app/
├── backend/
├── ml-models/
├── clinical-validation/
├── regulatory/
├── manufacturing/
├── deployment/
└── scripts/
```

1. Documentation (`docs/`)

```
docs/
├── README.md
├── technical-specifications/
│   ├── hardware-specs.md
│   ├── firmware-architecture.md
│   ├── app-requirements.md
│   ├── backend-api.md
│   └── system-integration.md
├── clinical/
│   ├── test-protocols.md
│   ├── validation-plan.md
│   ├── clinical-studies/
│   └── medical-literature-review.md
├── user-guides/
│   ├── patient-manual.md
│   ├── clinician-guide.md
│   └── troubleshooting.md
├── design/
│   ├── ui-mockups/
│   ├── hardware-schematics/
│   ├── mechanical-drawings/
│   └── user-flow-diagrams/
└── project-management/
    ├── roadmap.md
    ├── milestones.md
    └── risk-assessment.md
```

2. Hardware Design (`hardware/`)

```
hardware/
├── README.md
├── pcb-design/
│   ├── smart-insole-main/
│   │   ├── schematic.sch
│   │   ├── layout.pcb
│   │   ├── gerber-files/
│   │   └── bill-of-materials.csv
│   ├── charging-base/
│   │   ├── schematic.sch
│   │   ├── layout.pcb
│   │   └── gerber-files/
│   └── test-boards/
├── mechanical/
│   ├── insole-design/
│   │   ├── base-structure.step
│   │   ├── component-housings/
│   │   └── size-variants/
│   ├── materials/
│   │   ├── specifications.md
│   │   └── testing-reports/
│   └── assembly-drawings/
├── components/
│   ├── actuators/
│   │   ├── micro-solenoid/
│   │   ├── peltier-plates/
│   │   └── vibration-motors/
│   ├── sensors/
│   │   ├── temperature/
│   │   ├── pressure/
│   │   └── accelerometer/
│   ├── microcontroller/
│   │   ├── selection-criteria.md
│   │   └── pin-assignments.md
│   └── power-management/
│       ├── battery-specs.md
│       └── charging-circuit/
└── testing/
    ├── test-procedures/
    ├── validation-results/
    └── calibration-protocols/
```

3. Firmware (`firmware/`)

```
firmware/
├── README.md
├── src/
│   ├── main.c
│   ├── drivers/
│   │   ├── solenoid-driver.c/.h
│   │   ├── peltier-driver.c/.h
│   │   ├── vibration-driver.c/.h
│   │   ├── temperature-sensor.c/.h
│   │   ├── pressure-sensor.c/.h
│   │   └── bluetooth-stack.c/.h
│   ├── tests/
│   │   ├── pinprick-test.c/.h
│   │   ├── temperature-test.c/.h
│   │   └── vibration-test.c/.h
│   ├── safety/
│   │   ├── emergency-stop.c/.h
│   │   ├── temperature-monitor.c/.h
│   │   └── battery-safety.c/.h
│   ├── communication/
│   │   ├── bluetooth-handler.c/.h
│   │   ├── protocol-definitions.h
│   │   └── data-packaging.c/.h
│   ├── power-management/
│   │   ├── sleep-modes.c/.h
│   │   ├── battery-monitor.c/.h
│   │   └── charging-control.c/.h
│   └── utils/
│       ├── calibration.c/.h
│       ├── data-storage.c/.h
│       └── diagnostics.c/.h
├── include/
│   ├── config.h
│   ├── hardware-definitions.h
│   └── protocol.h
├── tests/
│   ├── unit-tests/
│   ├── integration-tests/
│   └── hardware-in-loop/
├── tools/
│   ├── firmware-updater/
│   ├── calibration-tools/
│   └── debugging-utilities/
└── build/
    ├── Makefile
    ├── cmake/
    └── toolchain/
```

4. Mobile Application (`mobile-app/`)

```
mobile-app/
├── README.md
├── ios/
│   ├── SmartShoe.xcodeproj
│   ├── SmartShoe/
│   │   ├── AppDelegate.swift
│   │   ├── SceneDelegate.swift
│   │   ├── ViewControllers/
│   │   │   ├── LoginViewController.swift
│   │   │   ├── DashboardViewController.swift
│   │   │   ├── TestViewController.swift
│   │   │   ├── HistoryViewController.swift
│   │   │   └── SettingsViewController.swift
│   │   ├── Models/
│   │   │   ├── User.swift
│   │   │   ├── TestResult.swift
│   │   │   └── Device.swift
│   │   ├── Services/
│   │   │   ├── BluetoothManager.swift
│   │   │   ├── APIClient.swift
│   │   │   └── DataManager.swift
│   │   └── Resources/
│   └── Podfile
├── android/
│   ├── app/
│   │   ├── build.gradle
│   │   └── src/main/
│   │       ├── AndroidManifest.xml
│   │       ├── java/com/smartshoe/
│   │       │   ├── MainActivity.java
│   │       │   ├── activities/
│   │       │   │   ├── LoginActivity.java
│   │       │   │   ├── DashboardActivity.java
│   │       │   │   ├── TestActivity.java
│   │       │   │   └── HistoryActivity.java
│   │       │   ├── models/
│   │       │   │   ├── User.java
│   │       │   │   ├── TestResult.java
│   │       │   │   └── Device.java
│   │       │   ├── services/
│   │       │   │   ├── BluetoothService.java
│   │       │   │   ├── ApiService.java
│   │       │   │   └── DatabaseService.java
│   │       │   └── utils/
│   │       └── res/
│   │           ├── layout/
│   │           ├── values/
│   │           └── drawable/
│   └── build.gradle
├── shared/
│   ├── models/
│   ├── protocols/
│   └── utilities/
├── assets/
│   ├── images/
│   ├── icons/
│   └── sounds/
└── tests/
    ├── unit-tests/
    ├── integration-tests/
    └── ui-tests/
```

5. Backend Services (`backend/`)

```
backend/
├── README.md
├── api/
│   ├── src/
│   │   ├── main/
│   │   │   ├── java/com/smartshoe/api/
│   │   │   │   ├── SmartShoeApplication.java
│   │   │   │   ├── controllers/
│   │   │   │   │   ├── UserController.java
│   │   │   │   │   ├── DeviceController.java
│   │   │   │   │   ├── TestDataController.java
│   │   │   │   │   └── AnalyticsController.java
│   │   │   │   ├── services/
│   │   │   │   │   ├── UserService.java
│   │   │   │   │   ├── DeviceService.java
│   │   │   │   │   ├── TestDataService.java
│   │   │   │   │   └── NotificationService.java
│   │   │   │   ├── repositories/
│   │   │   │   ├── models/
│   │   │   │   │   ├── User.java
│   │   │   │   │   ├── Device.java
│   │   │   │   │   ├── TestResult.java
│   │   │   │   │   └── ProgressionData.java
│   │   │   │   ├── config/
│   │   │   │   └── security/
│   │   │   └── resources/
│   │   │       ├── application.yml
│   │   │       └── db/migration/
│   │   └── test/
│   ├── build.gradle
│   └── Dockerfile
├── data-processing/
│   ├── src/
│   │   ├── ingestion/
│   │   ├── transformation/
│   │   ├── validation/
│   │   └── storage/
│   └── requirements.txt
├── analytics/
│   ├── progression-analysis/
│   ├── risk-assessment/
│   ├── reporting/
│   └── visualization/
├── infrastructure/
│   ├── kubernetes/
│   │   ├── api-deployment.yaml
│   │   ├── database-deployment.yaml
│   │   └── ingress.yaml
│   ├── terraform/
│   │   ├── main.tf
│   │   ├── variables.tf
│   │   └── outputs.tf
│   └── monitoring/
│       ├── prometheus/
│       └── grafana/
└── database/
    ├── schemas/
    ├── migrations/
    ├── seeds/
    └── backup-scripts/
```

6. Machine Learning Models (`ml-models/`)

```
ml-models/
├── README.md
├── data/
│   ├── raw/
│   ├── processed/
│   ├── training/
│   └── validation/
├── notebooks/
│   ├── exploratory-analysis.ipynb
│   ├── baseline-establishment.ipynb
│   ├── progression-modeling.ipynb
│   └── risk-prediction.ipynb
├── src/
│   ├── data-preprocessing/
│   │   ├── cleaning.py
│   │   ├── feature-extraction.py
│   │   └── normalization.py
│   ├── models/
│   │   ├── baseline-model.py
│   │   ├── progression-tracker.py
│   │   ├── anomaly-detector.py
│   │   └── risk-predictor.py
│   ├── training/
│   │   ├── train-baseline.py
│   │   ├── train-progression.py
│   │   └── hyperparameter-tuning.py
│   ├── evaluation/
│   │   ├── model-validation.py
│   │   ├── performance-metrics.py
│   │   └── clinical-correlation.py
│   └── deployment/
│       ├── model-serving.py
│       ├── batch-inference.py
│       └── api-integration.py
├── trained-models/
│   ├── baseline-v1.pkl
│   ├── progression-v1.pkl
│   └── risk-prediction-v1.pkl
├── experiments/
│   ├── experiment-configs/
│   └── results/
└── requirements.txt
```

7. Clinical Validation (`clinical-validation/`)

```
clinical-validation/
├── README.md
├── protocols/
│   ├── phase1-technical-validation.md
│   ├── phase2-clinical-pilot.md
│   └── phase3-extended-trial.md
├── data-collection/
│   ├── forms/
│   │   ├── patient-consent.pdf
│   │   ├── baseline-assessment.pdf
│   │   └── follow-up-forms/
│   ├── scripts/
│   │   ├── data-entry.py
│   │   ├── validation.py
│   │   └── anonymization.py
│   └── templates/
├── statistical-analysis/
│   ├── power-analysis.R
│   ├── comparative-analysis.R
│   ├── correlation-studies.R
│   └── longitudinal-modeling.R
├── results/
│   ├── phase1-results/
│   ├── phase2-results/
│   ├── phase3-results/
│   └── publications/
└── ethics/
    ├── irb-submissions/
    ├── consent-forms/
    └── safety-monitoring/
```

8. Regulatory Affairs (`regulatory/`)

```
regulatory/
├── README.md
├── fda/
│   ├── pre-submission/
│   ├── 510k-submission/
│   │   ├── device-description/
│   │   ├── predicate-analysis/
│   │   ├── performance-testing/
│   │   └── clinical-data/
│   └── quality-system/
│       ├── design-controls/
│       ├── risk-management/
│       └── post-market-surveillance/
├── international/
│   ├── ce-marking/
│   ├── health-canada/
│   └── other-markets/
├── standards-compliance/
│   ├── iso-13485/
│   ├── iso-14971/
│   ├── iec-62304/
│   └── hipaa-compliance/
└── documentation/
    ├── device-master-record/
    ├── design-history-file/
    └── technical-files/
```

9. Manufacturing (`manufacturing/`)

```
manufacturing/
├── README.md
├── processes/
│   ├── pcb-assembly/
│   ├── mechanical-assembly/
│   ├── calibration/
│   └── testing/
├── quality-control/
│   ├── incoming-inspection/
│   ├── in-process-testing/
│   ├── final-inspection/
│   └── packaging/
├── supply-chain/
│   ├── vendor-qualification/
│   ├── component-sourcing/
│   └── inventory-management/
├── equipment/
│   ├── production-line-setup/
│   ├── test-equipment/
│   └── maintenance-schedules/
└── documentation/
    ├── work-instructions/
    ├── test-procedures/
    └── quality-records/
```

10. Deployment & DevOps (`deployment/`)

```
deployment/
├── README.md
├── environments/
│   ├── development/
│   ├── staging/
│   ├── production/
│   └── disaster-recovery/
├── ci-cd/
│   ├── github-actions/
│   ├── jenkins/
│   └── deployment-scripts/
├── monitoring/
│   ├── application-monitoring/
│   ├── infrastructure-monitoring/
│   └── log-aggregation/
├── security/
│   ├── vulnerability-scanning/
│   ├── penetration-testing/
│   └── compliance-auditing/
└── backup-recovery/
    ├── backup-strategies/
    ├── disaster-recovery-plans/
    └── testing-procedures/
```

11. Utility Scripts (`scripts/`)

```
scripts/
├── README.md
├── setup/
│   ├── development-environment.sh
│   ├── database-setup.sql
│   └── dependencies-install.sh
├── data-migration/
│   ├── legacy-data-import.py
│   ├── schema-migration.py
│   └── data-validation.py
├── testing/
│   ├── automated-testing.sh
│   ├── performance-testing.py
│   └── load-testing.js
├── deployment/
│   ├── deploy-api.sh
│   ├── deploy-mobile-app.sh
│   └── rollback.sh
└── maintenance/
    ├── log-cleanup.sh
    ├── database-maintenance.sql
    └── system-health-check.py
```

File Naming Conventions

- Use kebab-case for directories and files where possible
- Use meaningful, descriptive names
- Include version numbers for major document revisions
- Prefix test files with `test_` or suffix with `.test`
- Use appropriate file extensions (.md, .py, .java, .swift, etc.)

Version Control Strategy

- Main branch for production-ready code
- Develop branch for integration
- Feature branches for individual development
- Release branches for version preparation
- Hotfix branches for critical fixes
- Tag releases with semantic versioning (v1.0.0)

This comprehensive file structure provides organization for all aspects of the smart shoe project, from initial hardware design through clinical validation, regulatory approval, and commercial deployment.