MS Companion: AI-Powered Multiple Sclerosis Management System

Hackathon Team

September 30, 2025

Abstract

This document outlines the technical specification for MS Companion, an AI-powered mobile application designed to predict relapse risk for Multiple Sclerosis patients. The system leverages machine learning and context-aware nudges to provide personalized, non-medical support, helping users maintain independence and well-being. Our minimum viable product (MVP) focuses on a core predictive early-warning system with mocked supporting features for demonstration purposes.

Contents

1 Introduction **3**
1.1 Problem Statement . . . 3
1.2 Our Solution . . . 3

2 System Architecture **3**
2.1 High-Level Overview . . . 3
2.2 Data Flow . . . 4

3 Core Features **4**
3.1 Feature 1: Relapse & Setback Predictor . . . 4
3.1.1 Multi-Modal Data Fusion . . . 4
3.1.2 Machine Learning Implementation . . . 4
3.1.3 API Endpoints . . . 5
3.2 Feature 2: Context-Aware Nudge Engine . . . 6
3.2.1 Micro-Context Rules . . . 6

4 Minimum Viable Product (MVP) Scope **6**
4.1 Actual Implementation . . . 6
4.2 Mocked Features for Demonstration . . . 8

5 Technical Implementation **9**
5.1 Tech Stack . . . 9
5.2 Flutter UI Components . . . 9

6 Data Privacy & Security **10**
6.1 Compliance Measures . . . 10
6.2 Privacy by Design . . . 10

7 Competitive Advantages **10**
7.1 Technical Differentiators 10
7.2 Market Differentiators 10

8 Conclusion **11**

1 Introduction

1.1 Problem Statement

People with Multiple Sclerosis (MS) face significant challenges including loss of autonomy during relapses, emotional strain, and fragmented rehabilitation programs. Current solutions fail to integrate health, lifestyle, and emotional data, missing crucial opportunities for early intervention and personalized support.

1.2 Our Solution

MS Companion addresses these challenges through:

- Multi-modal early warning system for relapse prediction
- Context-aware personalized suggestions
- Adaptive wellness planning
- Voice-enabled mood tracking
- Culturally sensitive mental health support

2 System Architecture

2.1 High-Level Overview

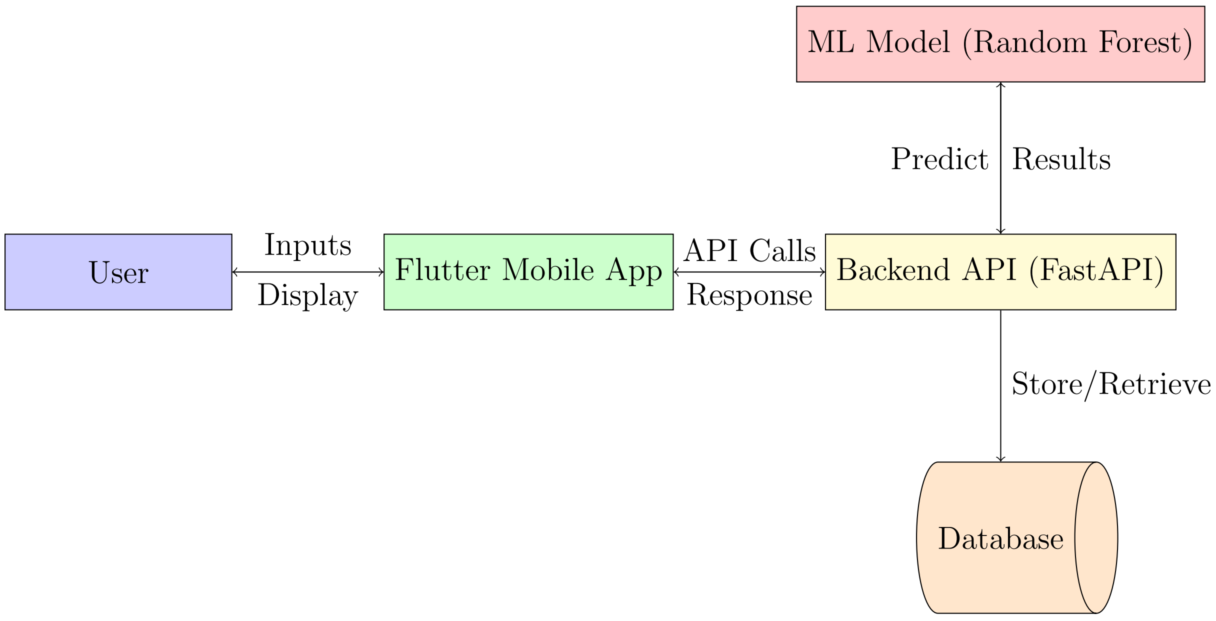

Figure 1: System Architecture Diagram

2.2 Data Flow

1. User inputs data via Flutter mobile app (sleep, mood, activity, fatigue)
2. App sends data to FastAPI backend via REST API
3. Backend processes data through pre-trained Random Forest model
4. ML model returns risk score and category
5. Backend generates personalized suggestions based on risk
6. Results displayed to user in dashboard

3 Core Features

3.1 Feature 1: Relapse & Setback Predictor

3.1.1 Multi-Modal Data Fusion

The AI creates a composite **Relapse Risk Score (0-100%)** by fusing multiple data sources:

Table 1: Data Inputs for Risk Prediction

Data Type	**Parameters**
Self-Reported Data	• Sleep Quality (1-5 scale) • Sleep Duration (hours) • Fatigue Level (1-5 scale) • Mood Score (1-5 scale) • Activity Steps (count)
Behavioral Data (Mocked)	• Voice journaling fatigue analysis • App interaction patterns • Keystroke dynamics

3.1.2 Machine Learning Implementation

Listing 1: ML Model Backend Code Skeleton

```python
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
import joblib
```

```
class MSRiskPredictor:
    def __init__(self):
        self.model = RandomForestClassifier(n_estimators=100, random_state=

    def train_model(self, X_train, y_train):
        self.model.fit(X_train, y_train)

    def predict_risk(self, features):
        """Predict relapse risk from input features"""
        probability = self.model.predict_proba([features])[0][1]
        return probability

    def get_risk_category(self, probability):
        if probability <= 0.3:
            return "Low"
        elif probability <= 0.7:
            return "Medium"
        else:
            return "High"
```

3.1.3 API Endpoints

Listing 2: REST API Endpoints

```
POST /api/predict
Request: {
    "user_id": "string",
    "sleep_quality": int,
    "sleep_duration": float,
    "fatigue_level": int,
    "mood_score": int,
    "activity_steps": int
}
Response: {
    "risk_score": float,
    "risk_category": "Low|Medium|High",
    "suggestion": "string",
    "timestamp": "ISO8601"
}

GET /api/history/{user_id}
Response: {
    "history": [
        {"date": "YYYY-MM-DD", "risk_score": float},
        ...
    ]
}
```

3.2 Feature 2: Context-Aware Nudge Engine

3.2.1 Micro-Context Rules

Table 2: Context-Aware Nudge Rules

Context	Condition	Action
Time & Sedentary	2:30 PM - 3:30 PM	”You’ve been stationary. How about a 2-minute standing stretch?”
Weather & Environment	Temperature ¿ 35°C	”High temperature today. Prioritize hydration and cool environments.”
Social Context	Mood Score 2 for 2 days	”Tough week? Want suggestions for low-energy social connection?”
Energy Dip	Risk score increases by ¿20%	”Noticed energy shift. Want to talk about how you’re feeling?”

4 Minimum Viable Product (MVP) Scope

4.1 Actual Implementation

Table 3: MVP Features - Actual Implementation

Feature	Purpose	Implementation Details
Relapse Risk Prediction	Core predictive engine	• Backend: FastAPI with scikit-learn • Model: RandomForest trained on synthetic data • Inputs: Sleep, Activity, Mood, Fatigue • Output: Risk score (0-100%) + category

Feature	**Purpose**	**Implementation Details**
User Input Interface	Data collection	• Flutter form with sliders and inputs • Mood/Fatigue: 1-5 sliders • Sleep: Number input (hours) • Activity: Step counter input
Personalized Suggestion Engine	Risk-based advice	• Low risk: "Maintain routine, stay hydrated" • Medium risk: "Light activities, gentle stretching" • High risk: "Rest, mindfulness, voice journal"
Dashboard UI	Risk visualization	• Color-coded risk category • Risk percentage display • Personalized suggestion text • Weekly trend button
Weekly Trend Chart	Historical data view	• Flutter chart library (fl_chart) • 7-day risk score history • Interactive trend visualization

Feature	Purpose	Implementation Details
Voice Input (Optional)	Speech-based input	• Flutter speech_to_text plugin • Keyword matching for mood/fatigue • Automatic risk recalculation

4.2 Mocked Features for Demonstration

Table 4: MVP Features - Mocked for Demo

Feature	Purpose	Implementation Details
Micro-context Awareness	Situational intelligence	• Hardcoded time-based rules • Mock location/weather data • Push notification triggers
Pre-Hab Report	Clinical utility demo	• PDF generation with placeholder data • Standardized clinical summary • Doctor-friendly format
Emergency Layer	Severe scenario handling	• Red emergency button UI • Mock caregiver alert popup • No actual SMS/email sent

Feature	Purpose	Implementation Details
Behavioral Analytics	Advanced data fusion	• Mock voice fatigue analysis • Placeholder for app usage patterns • Synthetic interaction data

5 Technical Implementation

5.1 Tech Stack

- **Frontend:** Flutter (Dart) - Cross-platform mobile development
- **Backend:** FastAPI (Python) - REST API with automatic documentation
- **ML:** Scikit-learn - Random Forest classification
- **Database:** SQLite (development) / PostgreSQL (production)
- **Deployment:** Docker containerization

5.2 Flutter UI Components

Listing 3: Flutter Dashboard Skeleton

```
class DashboardScreen extends StatefulWidget {
  @override
  _DashboardScreenState createState() => _DashboardScreenState();
}

class _DashboardScreenState extends State<DashboardScreen> {
  MSRiskData? riskData;

  Widget _buildRiskIndicator() {
    Color color = _getRiskColor(riskData?.category);
    return Container(
      color: color,
      child: Text(riskData?.category ?? 'Unknown'),
    );
  }

  Widget _buildSuggestion() {
    return Text(riskData?.suggestion ?? '');
  }
}
```

6 Data Privacy & Security

6.1 Compliance Measures

- **Data Encryption:** AES-256 for data at rest, TLS 1.3 for data in transit
- **User Consent:** Granular permissions for each data type
- **Data Anonymization:** User data pseudonymization for ML training
- **Regional Compliance:** Designed to meet UAE and US health data regulations

6.2 Privacy by Design

1. Minimal data collection - only essential parameters
2. Local processing where possible (voice-to-text on device)
3. Regular data purging policies
4. Transparent data usage explanations

7 Competitive Advantages

7.1 Technical Differentiators

1. **Personalized Risk Modeling:** User-specific baseline adaptation vs. population averages
2. **Multi-Modal Data Fusion:** Combining subjective and objective data sources
3. **Contextual Intelligence:** Environmental and situational awareness
4. **Proactive Intervention:** Early warning vs. reactive tracking

7.2 Market Differentiators

- **Cultural Sensitivity:** Arabic/English bilingual support with culturally appropriate content
- **Accessibility First:** Voice-first interface, high contrast, large text options
- **Non-Medical Focus:** Complementary to clinical care without overstepping boundaries
- **Trust-Based Design:** Transparent algorithms and user-controlled data sharing

8 Conclusion

MS Companion represents a significant advancement in digital health solutions for Multiple Sclerosis by combining predictive AI with empathetic, context-aware support. Our MVP demonstrates the core value proposition while laying the foundation for a comprehensive patient empowerment platform.

The system's unique approach to relapse prediction and personalized intervention addresses critical gaps in current MS management strategies, potentially improving quality of life and independence for people living with MS.

Appendices

Appendix A: Risk Calculation Formula

The relapse risk score is calculated using a weighted ensemble approach:

$$R = \alpha \cdot S + \beta \cdot M + \gamma \cdot F + \delta \cdot A + \epsilon \tag{1}$$

Where:

- R = Final risk score
- S = Sleep quality composite score
- M = Mood stability metric
- F = Fatigue progression indicator
- A = Activity pattern deviation
- $\alpha, \beta, \gamma, \delta$ = Feature weights learned by Random Forest
- ϵ = Error term

Appendix B: API Response Schema

Listing 4: Complete API Response Structure

```
{
  "success": boolean,
  "data": {
    "risk_assessment": {
      "score": float,
      "category": "Low|Medium|High",
      "confidence": float
    },
    "suggestions": {
      "immediate": string,
      "daily_plan": string,
      "watchlist": [string]
    },
    "contextual_nudges": [{
      "type": "time|weather|social",
      "message": string,
      "action": string
    }]
  },
  "timestamp": "ISO8601"
}
```