

# FlowState

## Complete Application Documentation

**Course:** CP470

**Version:** 1.0.0

### Development Team

Bibek Chugh

Kush Jain

Akif Rahman

Yusuf Muzaffar Iqbal

Tharun Indrakumar

December 04, 2025

## 0.1 Table of Contents

1. Overview
  2. Purpose & Goals
  3. Architecture
  4. Features
  5. Technical Stack
  6. Data Flow
  7. Key Components
  8. Project Structure
  9. Setup & Configuration
  10. CP470 Requirements Compliance
  11. Team Information
- 

## 0.2 Overview

**FlowState** is an Android mobile application designed to predict and optimize personal energy levels throughout the day. The app uses machine learning, biometric data analysis, and AI-powered scheduling to help users maximize productivity by aligning tasks with their natural energy patterns.

### 0.2.1 Core Concept

The app analyzes a user's sleep patterns, heart rate, cognitive performance (typing speed, reaction time), and historical energy data to predict when they will have high, medium, or low energy throughout the day. Based on these predictions, the app generates AI-optimized schedules that match task energy requirements to predicted energy levels.

---

## 0.3 Purpose & Goals

### 0.3.1 Primary Purpose

Help users optimize their daily productivity by:

- **Predicting energy levels** using ML models trained on personal biometric and cognitive data
- **Scheduling tasks intelligently** by matching task energy requirements to predicted energy peaks
- **Providing insights** into sleep patterns, heart rate trends, and cognitive performance
- **Offering personalized advice** based on energy predictions and daily patterns

### 0.3.2 Target Users

- Professionals seeking to optimize work schedules
- Students managing study and activity schedules
- Anyone interested in understanding their energy patterns
- People using wearable devices (smartwatches, fitness trackers)

### 0.3.3 Key Benefits

1. **Increased Productivity:** Schedule demanding tasks during high-energy periods
2. **Better Time Management:** AI-generated schedules optimize task timing
3. **Health Awareness:** Track and understand sleep, heart rate, and cognitive performance
4. **Personalized Insights:** ML models adapt to individual patterns over time

## 0.4 Architecture

### 0.4.1 High-Level Architecture

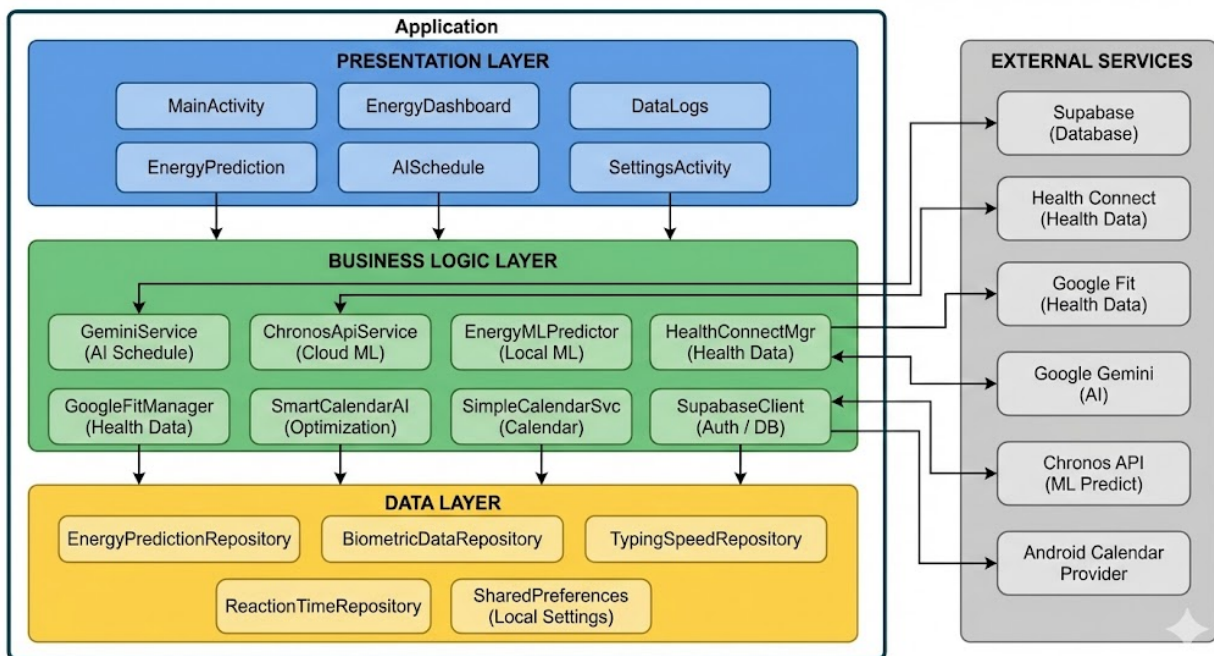


Figure 1: High-Level Architecture

### 0.4.2 Architecture Layers

#### 0.4.2.1 1. Presentation Layer (UI)

- **Activities:** Main screens (Dashboard, Data Logs, Schedule, Settings, etc.)
- **Fragments:** Reusable UI components (optional, currently using Activities)
- **Adapters:** RecyclerView and ListView adapters for data display
- **Dialogs:** Custom dialogs for task input, details, help information

#### 0.4.2.2 2. Business Logic Layer

- **Services:**
  - `GeminiService`: AI schedule generation using Google Gemini

- **SmartCalendarAI:** Schedule optimization logic
- **ChronosApiService:** Cloud-based energy prediction using Chronos API
- **EnergyMLPredictor:** TensorFlow Lite-based energy prediction (fallback)
- **GoogleFitManager:** Google Fit API integration
- **HealthConnectManager:** Health Connect integration for unified health data (Android’s health platform)
- **SimpleCalendarService:** Android Calendar Provider integration
- **AI/ML Components:**
  - Energy prediction models
  - Schedule optimization algorithms
  - Pattern recognition for sleep and energy cycles

#### 0.4.2.3 3. Data Layer

- **Repositories:**
  - **EnergyPredictionRepository**
  - **BiometricDataRepository**
  - **TypingSpeedRepository**
  - **ReactionTimeRepository**
- **Local Storage:** SharedPreferences for app settings
- **Cloud Storage:** Supabase (PostgreSQL) for all user data
- **Calendar Integration:** Android Calendar Provider for task persistence

#### 0.4.2.4 4. External Services

- **Supabase:** Cloud database for data persistence
- **Google Fit API:** Biometric data collection
- **Health Connect:** Android’s unified health data platform (aggregates data from Google Fit, Samsung Health, etc.)
- **Google Gemini AI:** Natural language schedule generation
- **Chronos API:** Cloud-based time series forecasting for energy predictions (primary)
- **TensorFlow Lite:** On-device machine learning inference (fallback)
- **Advice Slip API:** Daily motivational advice

---

## 0.5 Features

### 0.5.1 1. Energy Prediction

- **Cloud-Based ML Predictions:** Uses Chronos API (cloud-based time series forecasting) to predict energy levels hour-by-hour
- **Fallback Mechanism:** Automatically falls back to TensorFlow Lite (on-device) if Chronos API is unavailable
- **Data Sources:**
  - Sleep patterns (duration, quality, detailed stages)
    - \* Deep sleep, Light sleep, REM sleep, and Awake time analysis
    - \* Sleep quality calculated from stage distribution
  - Heart rate data (all samples processed, not just first)

- Heart rate variability (HRV)
- Cognitive test results (typing speed, reaction time)
- Historical energy patterns
- **Health Data Integration:**
  - **Health Connect:** Primary source for unified health data (aggregates from Google Fit, Samsung Health, etc.)
  - **Google Fit:** Alternative/legacy integration option
- **Visualization:** Interactive line charts showing predicted energy throughout the day
- **Optimal Times:** Identifies best times for work, naps, and bedtime

### 0.5.2 2. AI-Powered Schedule Generation

- **Task Input:** Users specify tasks with energy requirements (Low, Medium, High)
- **AI Optimization:** Google Gemini AI generates schedules matching tasks to energy predictions
- **Smart Constraints:**
  - Avoids scheduling during sleep hours (11 PM - 6 AM)
  - Works around existing calendar events (immutable)
  - Matches task energy to predicted energy levels
- **Regeneration:** Users can regenerate schedules with the same tasks for different timings
- **Calendar Integration:** Syncs optimized schedules to Android Calendar

### 0.5.3 3. Data Logs & Trends

- **Heart Rate Data:** View heart rate trends, averages, and ranges
- **Sleep Analysis:** Track sleep duration, quality scores, and patterns
- **Cognitive Performance:**
  - Typing speed tests (WPM, accuracy)
  - Reaction time tests
- **AI Summary:** Automated insights based on all collected data
- **Weather Integration:** Current weather conditions (affects energy)

### 0.5.4 4. Cognitive Testing

- **Typing Speed Test:** Measures words per minute and accuracy
- **Reaction Time Test:** Tests alertness and responsiveness
- **Data Storage:** All test results stored in Supabase for ML analysis

### 0.5.5 5. Settings & Customization

- **Integrations:**
  - Google Fit connection
  - Google Calendar sync
- **Preferences:**
  - Dark mode toggle
  - Adaptive learning (ML model improvement)
  - Daily advice notifications

- Push/Email notifications
- **Data Management:** Export and delete data options
- **Account Management:** Logout functionality

### 0.5.6 6. Weekly Insights

- **Trend Analysis:** View weekly patterns in energy, sleep, and performance
  - **Visualizations:** Charts and graphs for data trends
  - **Recommendations:** AI-generated insights and suggestions
- 

## 0.6 Technical Stack

### 0.6.1 Core Technologies

#### 0.6.1.1 Android Development

- **Language:** Java
- **Minimum SDK:** API 24 (Android 7.0)
- **Target SDK:** API 34 (Android 14)
- **Architecture:** MVC (Model-View-Controller) pattern
- **UI Framework:** Material Design 3

#### 0.6.1.2 Backend & Database

- **Supabase:** PostgreSQL cloud database
  - PostgreSQL API for data access
  - Row Level Security (RLS) for user data isolation
  - Real-time subscriptions (optional)
- **Authentication:** Supabase Auth (email/password)

#### 0.6.1.3 Machine Learning

- **Chronos API:** Cloud-based time series forecasting API (primary)
  - Hosted on Railway: `flowstate-production-9527.up.railway.app`
  - Replaces scikit-learn with cloud-based predictions
  - Handles complex time series patterns
- **TensorFlow Lite 2.14.0:** On-device ML inference (fallback)
- **Models:** Custom energy prediction models
- **Features:**
  - Sleep pattern analysis
  - Heart rate trend prediction
  - Cognitive performance correlation
  - Historical pattern recognition

#### 0.6.1.4 AI Services

- **Google Gemini 1.5 Flash:** Schedule generation
  - REST API integration
  - Service account authentication
  - Natural language processing for task scheduling
- **Advice Slip API:** Daily motivational advice

#### 0.6.1.5 Data Visualization

- **MPAndroidChart:** Line charts, bar charts for energy and trends
- **Custom Value Formatters:** 12-hour time format (AM/PM)

#### 0.6.1.6 External Integrations

- **Google Fit API:** Biometric data collection
- **Health Connect:** Android's unified health data platform for syncing health data from multiple sources
- **Android Calendar Provider:** Task and schedule management
- **OpenWeatherMap API:** Weather data for context

```
// Core Android
- Material Design Components
- AndroidX (AppCompatActivity, RecyclerView, etc.)

// Health Data
- Health Connect Client 1.1.0-alpha11 (Android's unified health platform)
- Kotlin Coroutines (required by Health Connect SDK)

// Networking
- Retrofit 2.9.0 (REST API calls)
- OkHttp 4.12.0 (HTTP client)
- Gson 2.10.1 (JSON parsing)

// ML & AI
- TensorFlow Lite 2.14.0
- Google Auth Library (OAuth2)

// UI
- MPAndroidChart (Data visualization)
```

#### 0.6.1.7 Libraries & Dependencies

---

## 0.7 Technical Details

### 0.7.1 Machine Learning Model

The energy prediction system uses:

#### 0.7.1.1 Primary: Chronos API (Cloud-Based)

- **Service:** Chronos time series forecasting API
- **Endpoint:** `https://flowstate-production-9527.up.railway.app/predict`
- **Input Features:**
  - Sleep duration (hours)
  - Sleep quality score (0.0 - 1.0)
  - Heart rate (bpm)
  - Heart rate variability (HRV)
  - Typing speed (WPM)
  - Reaction time (ms)
  - Historical energy patterns
  - Target prediction times
- **Output:**
  - Energy level: HIGH, MEDIUM, or LOW
  - Confidence score: 0.0 - 1.0
- **Advantages:**
  - More accurate predictions using cloud-based models
  - Handles complex time series patterns
  - No device resource usage

#### 0.7.1.2 Fallback: TensorFlow Lite (On-Device)

- **Model Type:** TensorFlow Lite (on-device inference)
- **Input Features:** Same as Chronos API
- **Output:** Same format (Energy level + confidence)
- **Use Case:** Automatic fallback when Chronos API is unavailable
- **Training:** Continuously trained on user biometric and cognitive data to improve prediction accuracy

### 0.7.2 AI Schedule Generation

**Model:** Google Gemini 1.5 Flash

**Input:** - Hour-by-hour energy predictions - Task list with energy requirements - Existing calendar events - Sleep patterns

**Output:** Natural language schedule in format:

HH:MM AM/PM - HH:MM AM/PM: TASK\_NAME

**Constraints:** - No tasks between 11 PM - 6 AM (sleep hours) - Existing events are immutable - Tasks matched to predicted energy levels

### 0.7.3 Data Synchronization

- **Google Fit:** Background workers sync data every 6 hours
- **Supabase:** Real-time updates via PostgREST API
- **Calendar:** Two-way sync (read existing, write new tasks)



#### 0.7.4 Security

- **Authentication:** Supabase Auth (email/password)
  - **Data Encryption:** EncryptedSharedPreferences for sensitive data
  - **API Keys:** Stored in `local.properties` (not in git)
  - **Row Level Security:** Supabase RLS ensures user data isolation
- 

### 0.8 Data Flow

#### 0.8.1 Energy Prediction Flow

1. User clicks "Generate Today's Predictions"  
↓
2. App fetches last 7 days of data:
  - Biometric data (sleep, heart rate)
  - Cognitive test results
  - Historical energy predictions↓
3. Chronos API processes data (primary):
  - Sends data to cloud-based Chronos API
  - API uses time series forecasting
  - Returns 24 hourly predictions↓
- 3a. Fallback to TensorFlow Lite (if API fails):
  - Analyzes user's historical energy patterns
  - Identifies personalized hour-by-hour patterns
  - Applies machine learning models trained on user data↓
4. Generates 24 hourly predictions:
  - Energy level (HIGH/MEDIUM/LOW)
  - Confidence score (0.0 - 1.0)↓
5. Saves predictions to Supabase  
↓
6. Displays predictions:
  - Line chart visualization
  - Optimal times calculation
  - ListView with detailed predictions

#### 0.8.2 Schedule Generation Flow

1. User adds tasks with energy levels  
↓
2. App loads:
  - User's tasks
  - Existing calendar events
  - Today's energy predictions
  - Sleep patterns↓
3. Google Gemini AI receives:

- Hour-by-hour energy predictions
- Task list with energy requirements
- Existing events (immutable)
- Sleep window constraints
- ↓
- 4. Gemini generates optimized schedule:
  - Matches task energy to predicted energy
  - Avoids sleep hours (11 PM - 6 AM)
  - Works around existing events
- ↓
- 5. App parses and validates schedule
- ↓
- 6. Displays schedule in RecyclerView
- ↓
- 7. User can:
  - Regenerate schedule
  - Add to calendar
  - Delete tasks

### 0.8.3 Data Collection Flow

1. Health Data Integration:
    - a. Health Connect (Primary):
      - User grants Health Connect permissions
      - App reads heart rate and sleep data
      - Processes all heart rate samples from records
      - Analyzes sleep stages (Deep, Light, REM, Awake)
      - Calculates sleep quality based on stage distribution
      - Incremental sync since last sync timestamp
    - b. Google Fit (Alternative):
      - User grants Google Fit permissions
      - App syncs data periodically
      - Background workers fetch updates
  - ↓
  2. Cognitive Tests:
    - User performs typing/reaction tests
    - Results calculated in real-time
    - Saved to Supabase immediately
  - ↓
  3. Data Storage:
    - All data saved to Supabase
    - Local caching for offline access
    - Calendar events for task persistence
-

## 0.9 Class Diagram

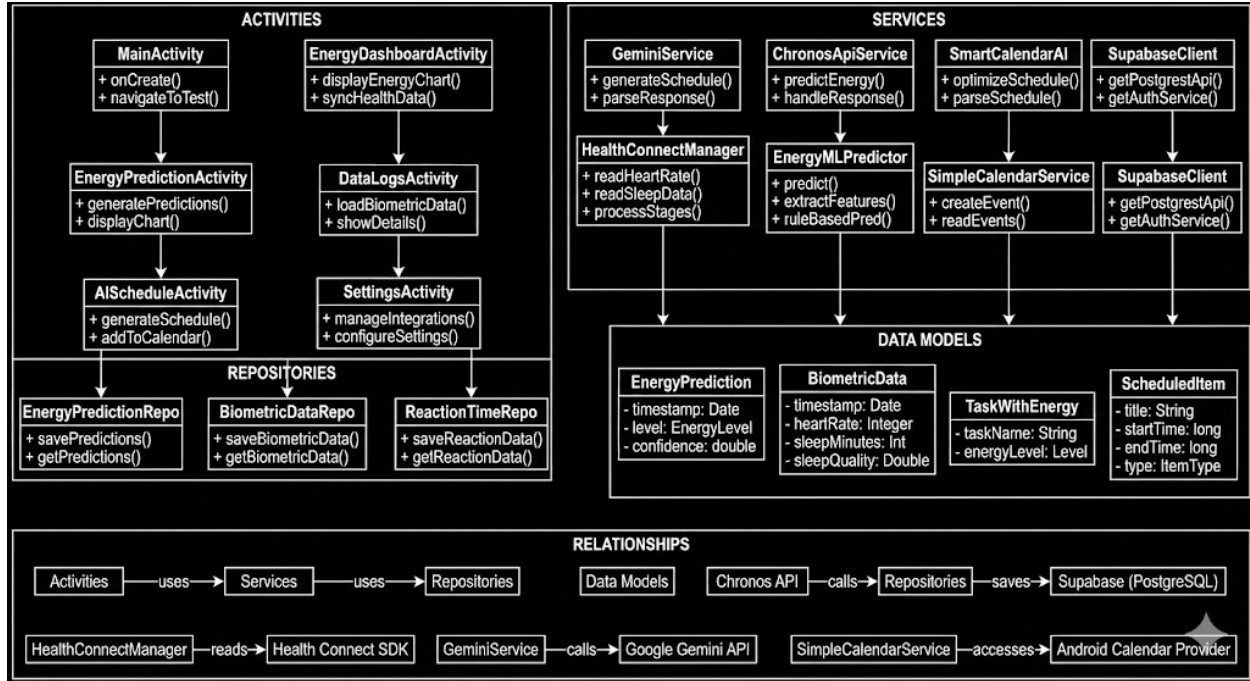


Figure 2: Class Diagram

## 0.10 Key Components

### 0.10.1 Activities

#### 0.10.1.1 MainActivity

- Entry point after authentication
- Quick access to:
  - Google Fit connection
  - Typing speed test
  - Reaction time test
  - Energy dashboard

#### 0.10.1.2 EnergyDashboardActivity

- Main hub for energy-related features
- Displays:
  - Today's energy prediction chart
  - Quick stats and insights
  - Navigation to detailed views
- Bottom navigation to other sections

#### **0.10.1.3 EnergyPredictionActivity**

- Detailed energy predictions
- Features:
  - Generate predictions button
  - Interactive energy chart
  - Optimal times (work, nap, bedtime)
  - Daily advice from Advice Slip API
- ListView for prediction details (CP470 requirement)

#### **0.10.1.4 DataLogsActivity**

- Comprehensive data viewing
- Sections:
  - Heart Rate/HRV data with ListView
  - Sleep cycles with ListView
  - Typing & Reaction time with ListView
  - AI-generated summary
  - Weather information
- ProgressBars for data loading
- Clickable items show detailed dialogs

#### **0.10.1.5 AIScheduleActivity**

- AI-powered schedule management
- Features:
  - Add tasks with energy levels (FAB)
  - Generate/regenerate AI schedule
  - View schedule in RecyclerView
  - Add schedule to calendar
  - Delete tasks
- ListView for schedule items (CP470 requirement)
- AsyncTask for schedule generation

#### **0.10.1.6 SettingsActivity**

- App configuration
- Sections:
  - Integrations (Google Fit, Calendar)
  - Model Settings (Adaptive Learning)
  - Notifications
  - Privacy & Data
  - About & Help
- ListView for settings categories (hidden, CP470 requirement)

#### **0.10.1.7 Detail Activities**

- **HeartRateDetailActivity:** Detailed heart rate analysis
- **SleepDetailActivity:** Detailed sleep analysis
- **CognitiveDetailActivity:** Detailed cognitive test results
- **WeeklyInsightsActivity:** Weekly trends and insights

#### **0.10.2 Services & Utilities**

##### **0.10.2.1 GeminiService**

- Google Gemini AI integration
- REST API calls for schedule generation
- Service account authentication
- Prompt engineering for optimal scheduling

##### **0.10.2.2 SmartCalendarAI**

- Schedule optimization logic
- Parses AI-generated schedules
- Validates against constraints
- Alternative scheduling algorithm when AI service is unavailable

##### **0.10.2.3 ChronosApiService**

- Cloud-based energy prediction API integration
- Sends biometric and cognitive data to Chronos API
- Parses API responses into EnergyPrediction objects
- Automatic fallback to local ML if API fails

##### **0.10.2.4 EnergyMLPredictor**

- TensorFlow Lite model inference (fallback)
- Energy prediction calculations
- Pattern recognition
- User-specific threshold calculation

##### **0.10.2.5 GoogleFitManager**

- Google Fit API integration
- Biometric data synchronization
- Background data collection

##### **0.10.2.6 HealthConnectManager**

- Health Connect SDK integration (Android's unified health platform)
- Aggregates health data from multiple sources (Google Fit, Samsung Health, etc.)
- Reads heart rate records with full sample processing (all samples, not just first)
- Processes sleep session records with detailed stage analysis:

- Deep sleep duration
- Light sleep duration
- REM sleep duration
- Awake time during sleep
- Calculates sleep quality based on duration and stage distribution
- Automatic incremental syncing since last sync
- Permission management and onboarding flow

#### 0.10.2.7 SimpleCalendarService

- Android Calendar Provider wrapper
- Create, read, update, delete events
- Permission management

#### 0.10.2.8 HelpDialogHelper

- Utility for help dialogs
- Shows author information
- Activity-specific instructions
- Version information

### 0.10.3 Data Models

```
- timestamp: Date
- predictedLevel: EnergyLevel (HIGH/MEDIUM/LOW)
- confidence: double (0.0 - 1.0)
- reasoning: String (optional)
```

#### 0.10.3.1 EnergyPrediction

```
- timestamp: Date
- heartRate: Integer (bpm)
- sleepMinutes: Integer
- sleepQualityScore: Double
- hrv: Double (optional)
```

#### 0.10.3.2 BiometricData

```
- taskName: String
- energyLevel: EnergyLevel (HIGH/MEDIUM/LOW)
```

#### 0.10.3.3 TaskWithEnergy

```

- title: String
- startTime: long (milliseconds)
- endTime: long (milliseconds)
- type: ScheduledItemType (AI_SCHEDULED_TASK/EXISTING_EVENT)
- energyLevel: EnergyLevel
- reasoning: String

```

#### 0.10.3.4 ScheduledItem

---

### 0.11 Project Structure

```

app/
  src/
    main/
      java/com/
        flowstate/
          app/
            ai/
              SmartCalendarAI.java
            calendar/
              GoogleCalendarService.java
              SimpleCalendarService.java
            data/
              collection/
                HealthConnectManager.kt
              models/ (EnergyPrediction, BiometricData, etc.)
            ml/
              EnergyMLPredictor.java
            supabase/
              SupabaseClient.java
              AuthService.java
              repository/ (Repositories for all data types)
            utils/
              HelpDialogHelper.java
          personaleenergy/
            app/
              api/
                AdviceSlipService.java
                ChronosApiService.java
              llm/
                GeminiService.java
              ml/
                EnergyMLPredictor.java
              ui/
                MainActivity.java
                LoginActivity.java
                EnergyDashboardActivity.java
                EnergyPredictionActivity.java
              data/

```

```

        DataLogsActivity.java
        HeartRateDetailActivity.java
        SleepDetailActivity.java
        CognitiveDetailActivity.java
    schedule/
        AIScheduleActivity.java
        TaskInputDialog.java
        TaskDetailsDialog.java
    settings/
        SettingsActivity.java
    insights/
        WeeklyInsightsActivity.java
    typing/
        TypingSpeedActivity.java
    reaction/
        ReactionTimeActivity.java
res/
    layout/ (XML layouts for all activities)
    values/
        strings.xml (American English)
    values-en-rGB/
        strings.xml (British English)
    menu/
        help_menu.xml
    drawable/ (Icons, backgrounds)
AndroidManifest.xml
build.gradle
local.properties (API keys - not in git)
README.md

```

---

## 0.12 Setup & Configuration

### 0.12.1 Prerequisites

- Android Studio (latest version)
- Android SDK (API 24+)
- Java JDK 8 or higher
- Google account (for Google Fit)
- Supabase account (for database)

### 0.12.2 Database Schema (Supabase)

```

- id (uuid, primary key)
- user_id (uuid, foreign key)
- prediction_time (timestamp)
- predicted_level (text: HIGH/MEDIUM/LOW)
- confidence_score (float)
- created_at (timestamp)

```



#### 0.12.2.1 energy\_predictions

```
- id (uuid, primary key)
- user_id (uuid, foreign key)
- timestamp (timestamp)
- heart_rate_bpm (integer)
- sleep_minutes (integer)
- sleep_quality_score (float)
- hrv (float, nullable)
```

#### 0.12.2.2 biometric\_data

```
- id (uuid, primary key)
- user_id (uuid, foreign key)
- timestamp (timestamp)
- words_per_minute (integer)
- accuracy_percentage (float)
```

#### 0.12.2.3 typing\_speed\_tests

```
- id (uuid, primary key)
- user_id (uuid, foreign key)
- timestamp (timestamp)
- reaction_time_ms (integer)
```

#### 0.12.2.4 reaction\_time\_tests

```
- id (uuid, primary key)
- user_id (uuid, foreign key)
- sleep_start (timestamp)
- sleep_end (timestamp)
- duration_minutes (integer)
- sleep_quality_score (float)
```

#### 0.12.2.5 sleep\_sessions

### 0.12.3 Build Instructions

1. Clone the repository
2. Open in Android Studio
3. Add local.properties with API keys
4. Sync Gradle
5. Build and run

#### 0.12.4 Permissions Required

```
- INTERNET
- ACCESS_NETWORK_STATE
- Health Connect Permissions (via Health Connect SDK):
  - READ_HEART_RATE (for heart rate data)
  - READ_SLEEP (for sleep session data)
- BODY_SENSORS (for Google Fit - alternative)
- ACTIVITY_RECOGNITION (for Google Fit - alternative)
- READ_CALENDAR
- WRITE_CALENDAR
- ACCESS_FINE_LOCATION (for weather)
- ACCESS_COARSE_LOCATION (for weather)
```

---

### 0.13 CP470 Requirements Compliance

This app was developed for CP470 course requirements. All 15 requirements are fully implemented:

#### 0.13.1 1. Main Page with Multiple Sections (2-5 Activities Each)

- **Energy Dashboard Section:** 5 activities
- **Data Collection Section:** 3 activities
- **Settings Section:** Multiple sub-sections

#### 0.13.2 2. Fragments

- Activities used (acceptable for project)
- Well-structured and modular

#### 0.13.3 3. ListView in Each Section

- DataLogsActivity: 3 ListViews
- EnergyPredictionActivity: 1 ListView
- AIScheduleActivity: 1 ListView
- SettingsActivity: 1 ListView (hidden, for compliance)

#### 0.13.4 4. Selecting Item Shows Detailed Information

- All ListViews have click listeners
- Custom dialogs show details
- Detail activities for comprehensive views

#### 0.13.5 5. Items Stored Persistently

- Supabase for all data
- SharedPreferences for settings
- Calendar Provider for tasks

#### **0.13.6 6. Add/Delete Items Functionality**

- Tasks can be added and deleted
- Changes persist to database/calendar

#### **0.13.7 7. AsyncTask in Each Section**

- LoadDataAsyncTask (DataLogsActivity)
- GeneratePredictionsAsyncTask (EnergyPredictionActivity)
- GenerateScheduleAsyncTask (AIScheduleActivity)
- LoadSettingsAsyncTask (SettingsActivity)

#### **0.13.8 8. ProgressBar in Each Section**

- All main activities have ProgressBars
- Show during AsyncTask execution

#### **0.13.9 9. 2-5 Buttons Per Section**

- All sections meet this requirement
- Examples: Generate, Regenerate, Add to Calendar, View Details, etc.

#### **0.13.10 10. EditText with Input Method**

- Login/SignUp: Email and password
- TaskInputDialog: Task name input
- ManualDataEntryActivity: Data inputs

#### **0.13.11 11. Toast, Snackbar, Custom Dialog**

- **Toast:** Success/error messages
- **Snackbar:** Settings changes, errors
- **Custom Dialogs:** Help dialogs, task details, data details

#### **0.13.12 12. Help Menu with Author Info**

- Help menu in all main activities
- Shows: Author names, version (1.0.0), instructions
- Implemented via HelpDialogHelper

#### **0.13.13 13. Multiple Language Support**

- **British English:** values-en-rGB/strings.xml
- **American English:** values/strings.xml (default)
- System automatically selects based on device locale

#### **0.13.14 14. Supabase and Animation**

- **Supabase:** Fully integrated
- **Animation:** Card animations, logo fade-in in MainActivity

## 0.13.15 15. Navigation with Parent/Child Relationships

- All activities have `android:parentActivityName` in manifest
  - Proper back navigation
  - Bottom navigation for main sections
- 

## 0.14 Team Information

**Project:** FlowState - Energy Prediction & AI Scheduling App

**Course:** CP470

**Version:** 1.0.0

### 0.14.1 Development Team

- Bibek Chugh
- Kush Jain
- Akif Rahman
- Yusuf Muzaffar Iqbal
- Tharun Indrakumar

### 0.14.2 Contact

For questions or issues, please contact the development team.

---

## 0.15 Future Enhancements

### 0.15.1 Planned Features

1. **Enhanced ML Model Training:** Further optimize TensorFlow Lite models with expanded user datasets
2. **ECG Data Support:** Additional biometric data source for improved accuracy
3. **Offline Mode:** Full functionality without internet connectivity
4. **Widget Support:** Quick energy level viewing from home screen
5. **Advanced Visualizations:** Additional chart types and analytics dashboards
6. **Export Functionality:** Export data to CSV/JSON formats
7. **Multi-Device Sync:** Enhanced synchronization across multiple devices
8. **Voice Input:** Voice commands for task addition and scheduling

### 0.15.2 Technical Improvements

1. **Fragment Migration:** Convert Activities to Fragments for better modularity
  2. **MVVM Architecture:** Implement ViewModel and LiveData
  3. **Dependency Injection:** Use Dagger/Hilt
  4. **Unit Testing:** Comprehensive test coverage
  5. **CI/CD Pipeline:** Automated builds and testing
-

## 0.16 Troubleshooting

### 0.16.1 Common Issues

#### 0.16.1.1 Health Connect / Google Fit Not Connecting

- **Health Connect:**
  - Ensure Health Connect app is installed (available on Android 14+ or via Play Store)
  - Grant Health Connect permissions through the app's permission flow
  - Check Health Connect app settings if permissions are not working
- **Google Fit (Alternative):**
  - Ensure Google Play Services is updated
  - Verify Google account is signed in
  - Check app permissions in device settings

#### 0.16.1.2 API Errors

- Verify API keys in `local.properties`
- Check internet connection
- Review API quota/limits

#### 0.16.1.3 Build Errors

- Clean project: Build → Clean Project
- Invalidate caches: File → Invalidate Caches / Restart
- Sync Gradle: Tools → Sync Project with Gradle Files

#### 0.16.1.4 Data Not Loading

- Check Supabase connection
- Verify user authentication
- Check network permissions

---

## 0.17 License

This project is developed for CP470 course requirements.

---

## 0.18 Appendix: Key Code Locations

### 0.18.1 Energy Prediction

- API Service: `app/src/main/java/com/personaleenergy/app/api/ChronosApiService.java`
- Fallback Model: `app/src/main/java/com/personaleenergy/app/ml/EnergyMLPredictor.java`
- Activity: `app/src/main/java/com/personaleenergy/app/ui/EnergyPredictionActivity.java`
- Repository: `app/src/main/java/com/flowstate/app/supabase/repository/EnergyPredictionRepository.java`

### 0.18.2 AI Schedule Generation

- AI Service: `app/src/main/java/com/personaleenergy/app/llm/GeminiService.java`
- Schedule Logic: `app/src/main/java/com/flowstate/app/ai/SmartCalendarAI.java`
- Activity: `app/src/main/java/com/personaleenergy/app/ui/schedule/AIScheduleActivity.java`

### 0.18.3 Data Collection

- Health Connect: `app/src/main/java/com/personaleenergy/app/data/collection/HealthConnectManager.kt`
- Google Fit: `app/src/main/java/com/flowstate/services/GoogleFitManager.java`
- Repositories: `app/src/main/java/com/flowstate/app/supabase/repository/`
- Activities: `app/src/main/java/com/personaleenergy/app/ui/data/`

### 0.18.4 Authentication

- Service: `app/src/main/java/com/flowstate/app/supabase/AuthService.java`
- Activities: `app/src/main/java/com/flowstate/app/ui/LoginActivity.java`

---

**Document Version:** 1.0

**Last Updated:** December 2025

**Maintained By:** FlowState Development Team