

Software Design Document

Sleep Fixer App

Group 6

Dang Nguyen Rafael Caldera Yohei Oya Yuanwei Chen

December 2025

Contents

Revision History	3
1 Introduction	4
1.1 Purpose of This Document	4
1.2 Intended Audience	4
1.3 Overview of the System	4
2 System Architecture	5
2.1 High-Level Workflow	5
2.2 Main Components	6
2.2.1 Presentation Layer (Flutter UI)	6
2.2.2 Business Logic Layer	6
2.2.3 Data Layer	6
3 Core Design Details	7
3.1 Data Storage (Hive Database)	7
3.1.1 Example Progress Entry	7
3.1.2 Example Sleep Plan	7
3.2 Key Algorithms	7
3.2.1 Shift Size Calculation	7
3.2.2 Plan Duration	7
3.2.3 Progress Status Rules	8
4 User Interface Design	9
4.1 Using the App	9
4.2 Navigation Layout	9
Glossary	10
References	11

Revision History

Name	Date	Reason for Changes	Version
Dang Nguyen	12/02/2025	Initial SDD based on Snapshot 1	1.0.0

1 Introduction

1.1 Purpose of This Document

This Software Design Specification (SDD) describes the design for Snapshot 1 of the Sleep Fixer mobile app. It explains the system structure, major components, and how different parts of the app work together. The goal is to provide a clear and easy-to-follow guide that helps the development team build and test the MVP version.

1.2 Intended Audience

This document is written for:

- Developers who will implement the app
- QA testers creating test plans
- Project managers tracking feature progress

1.3 Overview of the System

Sleep Fixer is a mobile app that helps users slowly shift their bedtime toward a healthier schedule. Snapshot 1 focuses on basic features:

- Creating a personalized sleep plan
- Tracking daily bedtime progress
- Storing data locally using Hive
- Showing onboarding messages and simple sleep tips

The app is built using Flutter and runs on both iOS and Android.

2 System Architecture

2.1 High-Level Workflow

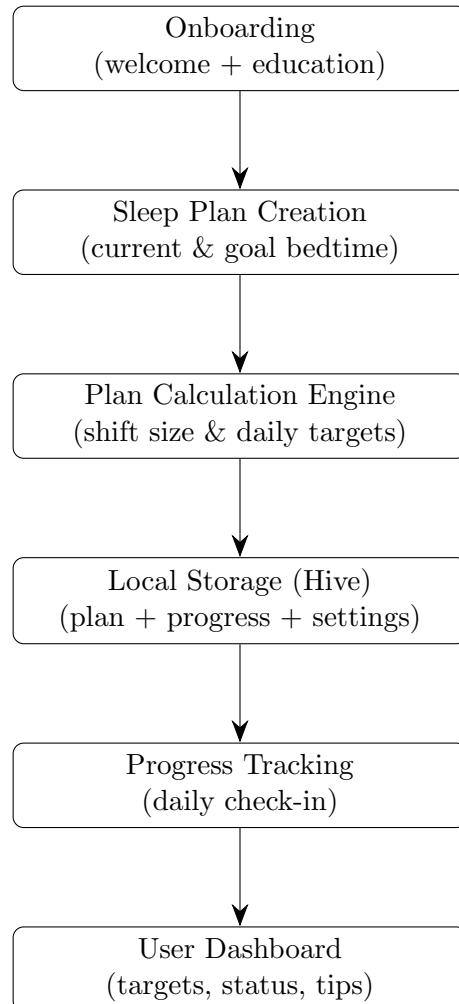


Figure 1: High-level workflow of the Sleep Fixer app

The system workflow is shown in Figure 1. The main steps are:

1. User completes onboarding screens.
2. User creates a sleep plan by entering current and goal bedtimes.
3. The app calculates a plan with daily bedtime targets.
4. The plan is saved locally in a Hive database.
5. Each day, the user enters their actual bedtime.
6. The app compares target vs. actual times and assigns a status.
7. The dashboard shows overall progress and helpful sleep tips.

2.2 Main Components

2.2.1 Presentation Layer (Flutter UI)

- **OnboardingScreen:** Introduces the app.
- **PlanCreationScreen:** Allows users to enter bedtime information.
- **FullPlanScreen:** Shows the list of bedtime targets.
- **ProgressTrackingDialog:** User inputs actual bedtime.
- **ProfileScreen:** Shows settings and plan summary.

2.2.2 Business Logic Layer

- **SleepPlanCalculator:** Creates daily bedtime targets.
- **ProgressTrackingService:** Calculates progress status.
- **TimeCalculationUtils:** Helper functions for time differences.

2.2.3 Data Layer

- **HiveService:** Opens and manages Hive database boxes.
- **SleepPlan Model:** Stores plan information.
- **SleepProgress Model:** Stores daily progress entries.

3 Core Design Details

3.1 Data Storage (Hive Database)

The app uses Hive, a lightweight NoSQL database stored on the user's device. Two boxes are used:

- **sleep_progress**: Stores daily progress (date, target time, actual time, status, notes).
- **sleep_settings**: Stores plan configuration (current bedtime, goal bedtime, shift size, etc.).

3.1.1 Example Progress Entry

```
{  
    "date": "2025-12-02",  
    "targetTime": "22:00",  
    "actualTime": "22:15",  
    "status": "within30Min",  
    "notes": "Felt good tonight"  
}
```

3.1.2 Example Sleep Plan

```
{  
    "currentBedtime": "23:30",  
    "goalBedtime": "22:00",  
    "shiftSize": 20,  
    "planLength": 9,  
    "startDate": "2025-12-01"  
}
```

3.2 Key Algorithms

3.2.1 Shift Size Calculation

```
IF time difference <= 60 minutes → shift = 15 minutes  
ELSE IF <= 180 minutes → shift = 20 minutes  
ELSE → shift = 30 minutes
```

3.2.2 Plan Duration

```
shiftsNeeded = CEILING(totalDifference / shiftSize)  
daysPerShift = 1 if difference <= 120 min else 2  
totalDays = shiftsNeeded × daysPerShift
```

3.2.3 Progress Status Rules

```
0 minutes difference → onTarget  
<= 30 minutes → within30Min  
<= 60 minutes → within1Hour  
> 60 minutes → offTarget
```

4 User Interface Design

4.1 Using the App

1. The onboarding screens explain how gradual sleep shifts work.
2. The user enters their starting and goal bedtime.
3. The app shows the calculated schedule.
4. Each day, the user taps the current day and enters their actual bedtime.
5. The dashboard displays progress and sleep tips.

4.2 Navigation Layout

- **Home:** Shows daily targets and progress.
- **Profile:** Shows plan summary and settings.

Glossary

- SDD - Software Design Document
- UI - User Interface
- MVP - Minimum Viable Product
- API - Application Programming Interface

References

- Flutter Documentation — <https://docs.flutter.dev>
- Hive Database Documentation — <https://docs.hivedb.dev>
- Dart Language Guides — <https://dart.dev/guides>
- National Sleep Foundation — <https://www.sleepfoundation.org>
- Harvard Medical School Sleep Research — <https://sleep.med.harvard.edu>
- Material Design Guidelines — <https://material.io/design>