

# Michi Finance

## Documentation

Luis González (LuisGH28)

October 21, 2025

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Technologies . . . . .	2
<b>2</b>	<b>Goals</b>	<b>2</b>
2.1	General Objective . . . . .	2
2.2	Specific Objectives . . . . .	2
<b>3</b>	<b>Features and Requirements</b>	<b>2</b>
3.1	Development Requirements . . . . .	2
3.2	Navigation and Modules . . . . .	2
3.3	Key Screens . . . . .	3
<b>4</b>	<b>Architecture</b>	<b>3</b>
4.1	Pattern . . . . .	3
4.2	Project Structure . . . . .	3
4.3	Databases . . . . .	3
<b>5</b>	<b>Data Synchronization and Model</b>	<b>3</b>
5.1	Functionalities . . . . .	3
5.2	Suggested Dependencies . . . . .	4
5.3	Sync Strategy . . . . .	4
<b>6</b>	<b>UI/UX Design</b>	<b>4</b>
6.1	Color Palette . . . . .	4
6.2	Inspiration . . . . .	4
<b>7</b>	<b>Installation and Internationalization (i18n)</b>	<b>4</b>
<b>8</b>	<b>Data Export</b>	<b>4</b>
8.1	Formats . . . . .	4
8.2	Contents . . . . .	4
8.3	Suggested Dependencies . . . . .	5
8.4	UX Considerations . . . . .	5
<b>9</b>	<b>AI Assistant</b>	<b>5</b>
<b>10</b>	<b>Work Plan and Roadmap</b>	<b>5</b>
10.1	Phases . . . . .	5
10.2	Estimated Workload . . . . .	5
10.3	Deliverables . . . . .	5

## 1 Introduction

**Michi Finance** is a friendly mobile app for personal finance tracking, designed for users who seek a clear, visual, and functional experience. It allows users to record expenses, set saving goals, visualize dynamic charts, export reports, and receive AI-based suggestions. The idea arises from the need to achieve better financial stability by knowing where money goes and building savings through efficient management.

### 1.1 Technologies

- Flutter + Dart, `flutter_localizations`, `intl`
- SQLite (with Firebase option)
- `fl_chart`
- Firebase Auth (Google/Facebook)
- Obsidian for documentation

## 2 Goals

### 2.1 General Objective

Develop a cross-platform mobile app to manage personal finances in a visual, friendly, and personalized way.

### 2.2 Specific Objectives

- Allow users to easily record income and expenses.
- Display statistics and comparative charts.
- Set and track financial goals.
- Export monthly reports (CSV and PDF).
- Integrate an intelligent assistant for recommendations.
- Provide an intuitive UI with custom illustrations and icons.

## 3 Features and Requirements

### 3.1 Development Requirements

- Flutter/Dart
- SQLite
- Python with Flask (for optional AI backend)

### 3.2 Navigation and Modules

**Navigation:** Bottom Navigation (Home, Add, Reports, Goals) and a side Drawer (Settings, Account, Sign Out).

### 3.3 Key Screens

- **HomeScreen:** displays app name with logo, expense summary, remaining balance, main expenses list, expense distribution chart, floating button for AI assistant, and export options (Excel or PDF).
- **AddExpense:** includes image, title “Add Expense”, amount input, category name, and save button.
- **Goals:** lists goals with image, name, progress bar, and a detailed view showing creation date, amount, note, and status (Started, Halfway, Completed, Canceled) with edit capabilities.

## 4 Architecture

### 4.1 Pattern

Clean Architecture (layered) with state management using Riverpod (or Provider as an alternative).

### 4.2 Project Structure

```
lib/  
  ai_assistant/      # virtual assistant  
  config/  
  data/              # database configuration  
    database/  
    firebase/  
  export/            # CSV and PDF export functions  
  main.dart  
  models/  
  providers/  
  screens/  
  services/  
  utils/  
  widgets/
```

### 4.3 Databases

SQLite for local and offline data storage. Firebase for authentication and cloud synchronization.

## 5 Data Synchronization and Model

**Purpose:** synchronize local data (SQLite) with the cloud (Firebase) efficiently and safely.

### 5.1 Functionalities

- Automatic backup when connected to the Internet.
- Manual “sync now” option.
- Timestamp comparison to avoid overwriting recent data.
- Data recovery from Firebase on new devices.

## 5.2 Suggested Dependencies

`cloud_firestore`, `connectivity_plus`

## 5.3 Sync Strategy

Hybrid repository (local/cloud) logic. Triggered on:

- Sign-in
- Periodically (optional)
- Manual sync button

# 6 UI/UX Design

## 6.1 Color Palette

Primary `#FFC107` (Yellow), Secondary `#4CAF50` (Green), Background `#FFF8E1`, with soft and friendly accent tones.

## 6.2 Inspiration

Material Design principles with a kawaii/cat-inspired aesthetic for an engaging and friendly interface.

# 7 Installation and Internationalization (i18n)

**Purpose:** provide multi-language support (Spanish and English) using `flutter_localizations` and `intl`.

Listing 1: Dependencies in `pubspec.yaml`

```
dependencies:
  flutter_localizations:
    sdk: flutter
  intl: ^0.18.0
```

# 8 Data Export

**Purpose:** allow users to export their financial information in different formats for backup or review.

## 8.1 Formats

- CSV (structured table, ideal for Excel/Google Sheets)
- PDF (visual report with charts and summaries)

## 8.2 Contents

- Expenses by category
- Completed and pending goals
- Monthly comparisons (optional)
- Date, description, amount, and category

### 8.3 Suggested Dependencies

csv, pdf, printing, share\_plus

### 8.4 UX Considerations

Export button with format options; optional scheduled monthly automatic export.

## 9 AI Assistant

The AI assistant helps users organize and record their expenses more efficiently, providing savings suggestions and optimal money distribution methods. The assistant uses a custom model designed for real-time responses (Flask or Node.js backend).

## 10 Work Plan and Roadmap

### 10.1 Phases

1. Documentation and preliminary design
2. Flutter project setup
3. Navigation structure and base screens
4. Local database with SQLite
5. Expense registration and visualization
6. CSV export
7. Dynamic charts
8. Firebase Authentication
9. PDF export
10. AI assistant (Flask or Node.js)
11. Testing and deployment

### 10.2 Estimated Workload

Two hours per day (Monday–Friday), with weekly documented progress.

### 10.3 Deliverables

Source code on GitHub, Markdown/Obsidian documentation, screenshots, and demo video.