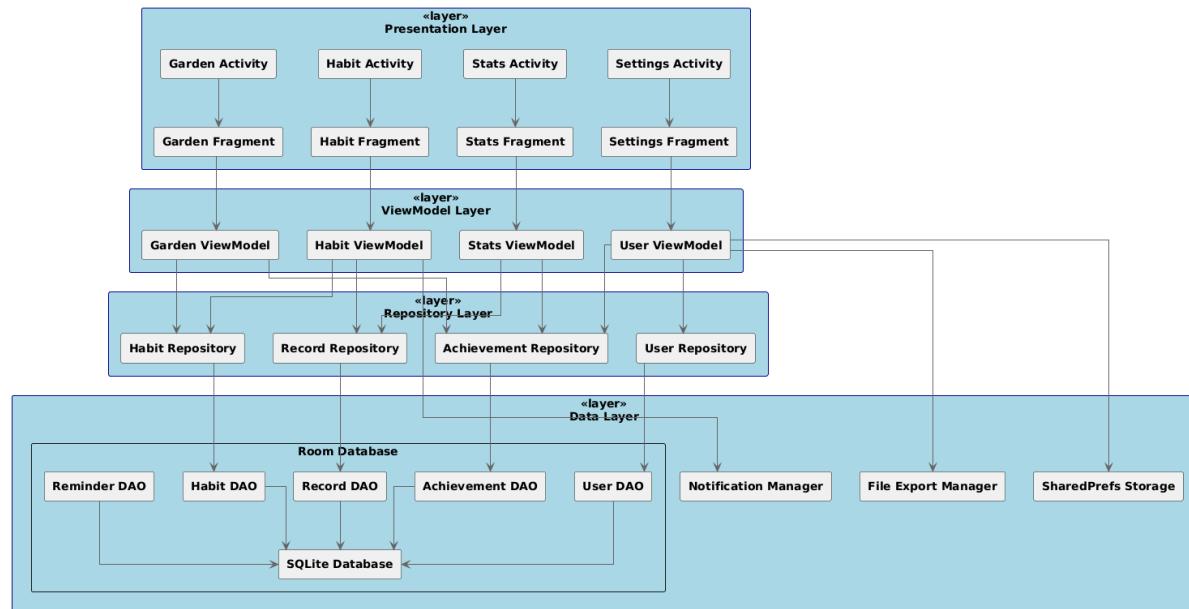


HabitBloom 系统设计说明书

1. 系统架构设计

1.1 总体架构图



1.2 架构说明

1.2.1 架构模式：MVVM (Model-View-ViewModel)

采用 Android 推荐的 MVVM 架构模式，实现关注点分离：

层级	职责	组件
View	负责UI展示和用户交互	Activity, Fragment, XML Layout
ViewModel	负责UI逻辑和状态管理	ViewModel + LiveData
Model	负责数据存储和业务逻辑	Repository, Room, Entity

1.2.2 分层设计

表现层 (Presentation Layer)

- 包含所有 Activity 和 Fragment
- 通过 DataBinding 实现视图绑定
- 观察 ViewModel 中的 LiveData 更新UI

ViewModel 层

- 持有 LiveData 状态数据
- 处理用户交互事件
- 调用 Repository 进行数据操作
- 生命周期感知，避免内存泄漏

数据仓库层 (Repository Layer)

- 统一数据访问接口
- 协调本地数据源
- 处理数据转换和业务逻辑

数据层 (Data Layer)

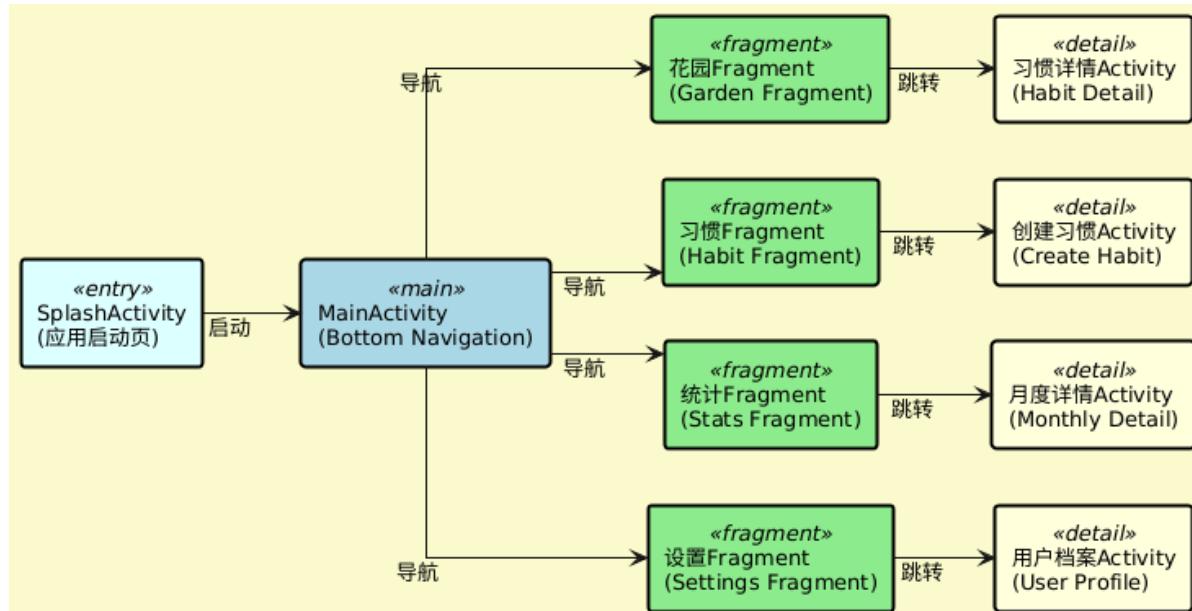
- Room 数据库持久化存储
- SharedPreferences 配置存储
- 文件系统用于数据导出/导入

1.3 技术选型

类别	技术方案	说明
开发语言	Java	Android 原生开发
最低版本	Android 8.0 (API 26)	兼容性考虑
数据库	Room	Android 官方 ORM
异步处理	LiveData + Executor	响应式数据
UI 组件	Material Design 3	现代化设计
导航	Navigation Component	统一导航管理
通知	WorkManager + AlarmManager	定时提醒

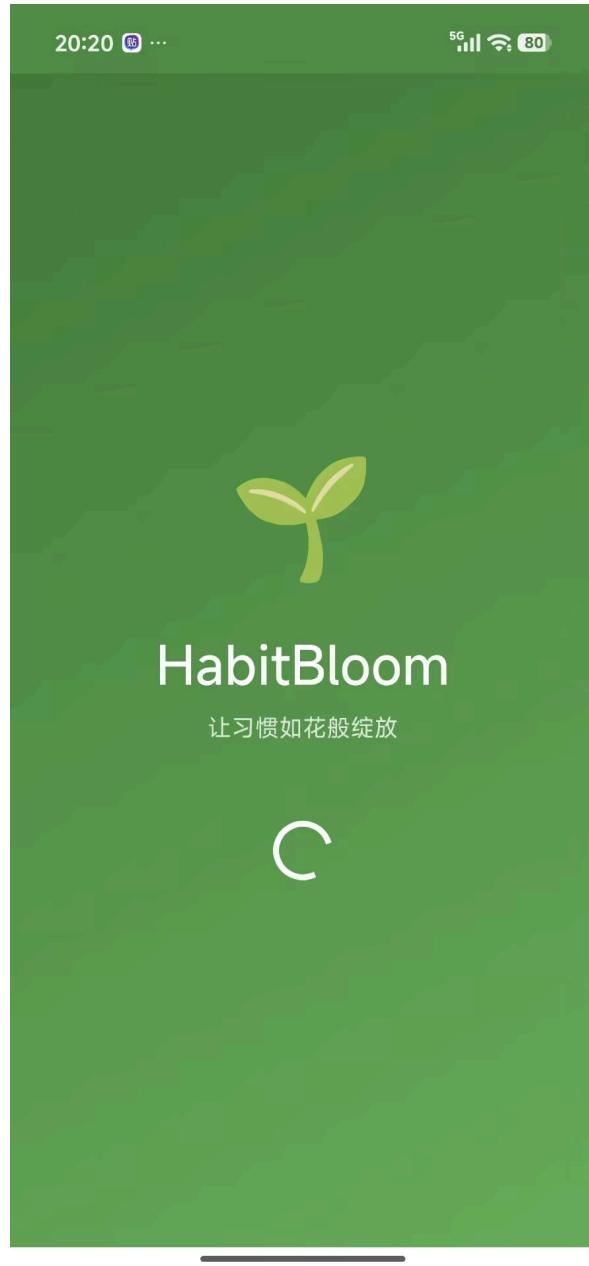
2. UI/UX 设计

2.1 页面导航结构



2.2 页面原型设计

2.2.1 启动页 (SplashActivity)



2.2.2 主花园页面 (GardenFragment)



2.2.3 习惯详情页 (HabitDetailActivity)



2.2.4 创建习惯页面 (CreateHabitActivity)



2.2.5 统计页面 (StatsFragment)



2.2.6 设置页面 (SettingsFragment)



2.3 设计规范

2.3.1 色彩方案

用途	颜色代码	说明
主色调	#4CAF50	绿色，代表生长
主色调深	#388E3C	深绿色
主色调浅	#C8E6C9	浅绿色背景
强调色	#FF9800	橙色，用于高亮
背景色	#F5F5F5	浅灰背景
卡片色	#FFFFFF	白色卡片
文字主色	#212121	深灰文字

用途	颜色代码	说明
文字次色	#757575	灰色文字
成功色	#4CAF50	绿色
警告色	#FFC107	黄色
错误色	#F44336	红色

2.3.2 字体规范

用途	字号	字重
大标题	24sp	Bold
标题	20sp	Medium
子标题	16sp	Medium
正文	14sp	Regular
辅助文字	12sp	Regular
按钮文字	14sp	Medium

2.3.3 间距规范

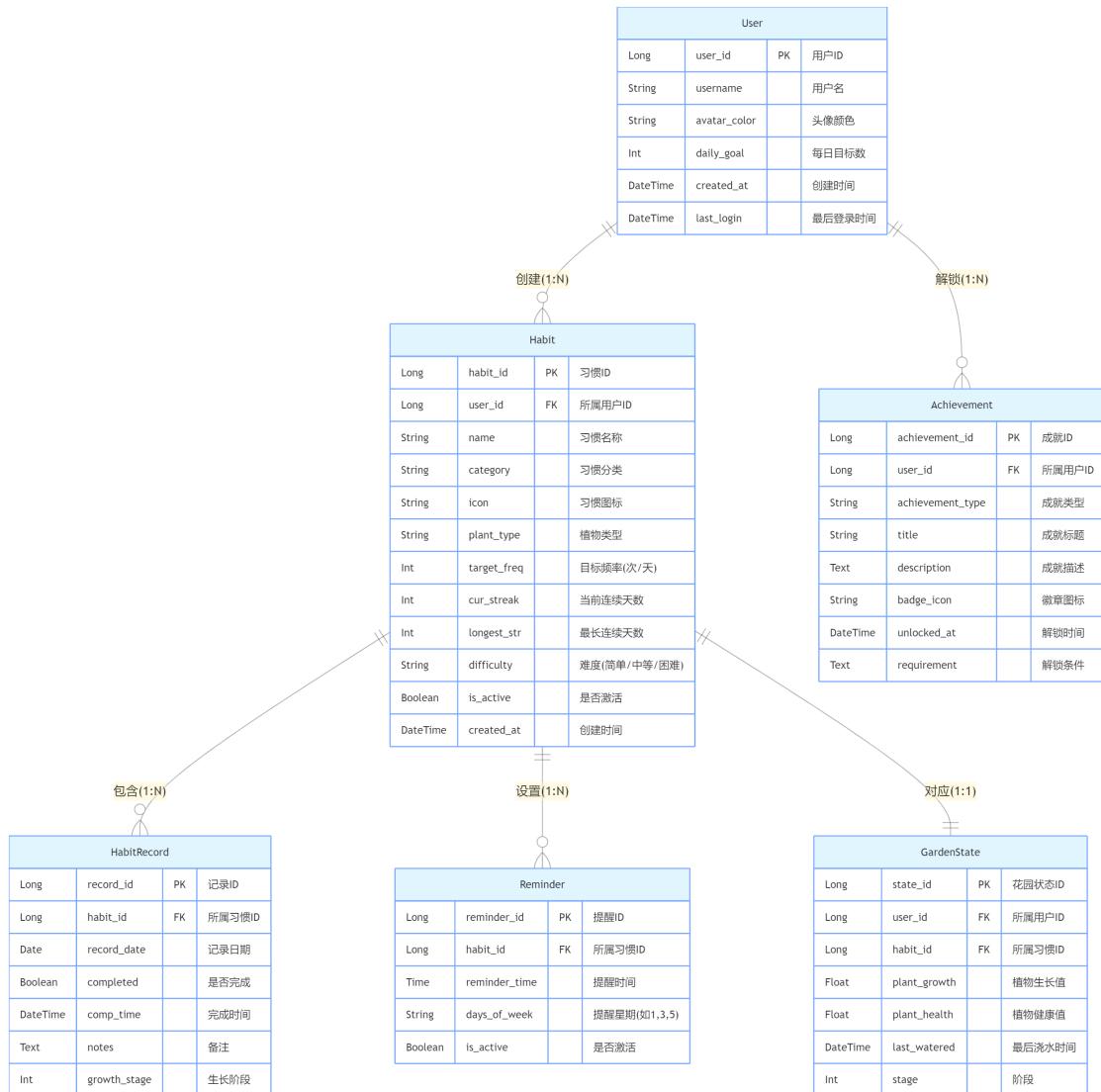
用途	尺寸
页面边距	16dp
卡片边距	12dp
组件间距	8dp
小间距	4dp
圆角	12dp

2.3.4 动画规范

动画类型	时长	曲线
页面过渡	300ms	ease-in-out
植物生长	500ms	ease-out
浇水效果	200ms	ease-in
按钮反馈	100ms	linear
成就解锁	800ms	bounce

3. 数据库设计

3.1 ER图



3.2 表结构设计

3.2.1 users 表 - 用户信息表

```
CREATE TABLE users (
    user_id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT NOT NULL DEFAULT '用户',
    avatar_color TEXT DEFAULT '#4CAF50',
    daily_goal_time TEXT DEFAULT '08:00',
    created_at INTEGER NOT NULL DEFAULT (strftime('%s', 'now') * 1000),
    last_login INTEGER
);
```

字段	类型	约束	说明
user_id	INTEGER	PK, AUTO	用户唯一标识

字段	类型	约束	说明
username	TEXT	NOT NULL	用户昵称
avatar_color	TEXT	DEFAULT	头像颜色
daily_goal_time	TEXT	DEFAULT	每日目标时段
created_at	INTEGER	NOT NULL	创建时间戳
last_login	INTEGER	NULL	最后登录时间

3.2.2 habits 表 - 习惯信息表

```

CREATE TABLE habits (
    habit_id INTEGER PRIMARY KEY AUTOINCREMENT,
    user_id INTEGER NOT NULL DEFAULT 1,
    name TEXT NOT NULL,
    category TEXT CHECK(category IN ('health', 'study', 'work', 'life')) DEFAULT
    'life',
    icon TEXT DEFAULT '鄢',
    plant_type TEXT CHECK(plant_type IN ('flower', 'tree', 'cactus', 'herb')) DEFAULT
    'flower',
    target_frequency INTEGER DEFAULT 7,
    current_streak INTEGER DEFAULT 0,
    longest_streak INTEGER DEFAULT 0,
    total_completed INTEGER DEFAULT 0,
    difficulty INTEGER CHECK(difficulty BETWEEN 1 AND 5) DEFAULT 3,
    is_active INTEGER DEFAULT 1,
    created_at INTEGER NOT NULL DEFAULT (strftime('%s', 'now') * 1000),
    FOREIGN KEY (user_id) REFERENCES users(user_id) ON DELETE CASCADE
);

CREATE INDEX idx_habits_user_id ON habits(user_id);
CREATE INDEX idx_habits_category ON habits(category);

```

字段	类型	约束	说明
habit_id	INTEGER	PK, AUTO	习惯唯一标识
user_id	INTEGER	FK, NOT NULL	关联用户
name	TEXT	NOT NULL	习惯名称
category	TEXT	CHECK	分类(health/study/work/life)
icon	TEXT	DEFAULT	图标emoji
plant_type	TEXT	CHECK	植物类型
target_frequency	INTEGER	DEFAULT 7	每周目标次数
current_streak	INTEGER	DEFAULT 0	当前连续天数
longest_streak	INTEGER	DEFAULT 0	最长连续天数

字段	类型	约束	说明
total_completed	INTEGER	DEFAULT 0	总完成次数
difficulty	INTEGER	CHECK 1-5	难度等级
is_active	INTEGER	DEFAULT 1	是否启用
created_at	INTEGER	NOT NULL	创建时间戳

3.2.3 habit_records 表 - 打卡记录表

```

CREATE TABLE habit_records (
    record_id INTEGER PRIMARY KEY AUTOINCREMENT,
    habit_id INTEGER NOT NULL,
    record_date TEXT NOT NULL,
    completed INTEGER DEFAULT 0,
    completed_time TEXT,
    notes TEXT,
    plant_growth_stage INTEGER DEFAULT 0,
    UNIQUE(habit_id, record_date),
    FOREIGN KEY (habit_id) REFERENCES habits(habit_id) ON DELETE CASCADE
);

CREATE INDEX idx_records_habit_id ON habit_records(habit_id);
CREATE INDEX idx_records_date ON habit_records(record_date);

```

字段	类型	约束	说明
record_id	INTEGER	PK, AUTO	记录唯一标识
habit_id	INTEGER	FK, NOT NULL	关联习惯
record_date	TEXT	NOT NULL	记录日期(yyyy-MM-dd)
completed	INTEGER	DEFAULT 0	是否完成(0/1)
completed_time	TEXT	NULL	完成时间(HH:mm)
notes	TEXT	NULL	备注/日记
plant_growth_stage	INTEGER	DEFAULT 0	植物生长阶段

3.2.4 reminders 表 - 提醒设置表

```

CREATE TABLE reminders (
    reminder_id INTEGER PRIMARY KEY AUTOINCREMENT,
    habit_id INTEGER NOT NULL,
    reminder_time TEXT NOT NULL,
    days_of_week TEXT DEFAULT '1,2,3,4,5,6,7',
    is_active INTEGER DEFAULT 1,
    FOREIGN KEY (habit_id) REFERENCES habits(habit_id) ON DELETE CASCADE
);

CREATE INDEX idx_reminders_habit_id ON reminders(habit_id);

```

字段	类型	约束	说明
reminder_id	INTEGER	PK, AUTO	提醒唯一标识
habit_id	INTEGER	FK, NOT NULL	关联习惯
reminder_time	TEXT	NOT NULL	提醒时间(HH:mm)
days_of_week	TEXT	DEFAULT	周几提醒(1-7逗号分隔)
is_active	INTEGER	DEFAULT 1	是否启用

3.2.5 achievements 表 - 成就表

```

CREATE TABLE achievements (
    achievement_id INTEGER PRIMARY KEY AUTOINCREMENT,
    user_id INTEGER NOT NULL DEFAULT 1,
    achievement_type TEXT NOT NULL,
    title TEXT NOT NULL,
    description TEXT,
    badge_icon TEXT NOT NULL,
    unlocked_at INTEGER,
    requirement_value INTEGER,
    FOREIGN KEY (user_id) REFERENCES users(user_id) ON DELETE CASCADE
);

CREATE INDEX idx_achievements_user_id ON achievements(user_id);

```

字段	类型	约束	说明
achievement_id	INTEGER	PK, AUTO	成就唯一标识
user_id	INTEGER	FK, NOT NULL	关联用户
achievement_type	TEXT	NOT NULL	成就类型
title	TEXT	NOT NULL	成就标题
description	TEXT	NULL	成就描述
badge_icon	TEXT	NOT NULL	徽章图标
unlocked_at	INTEGER	NULL	解锁时间(null=未解锁)
requirement_value	INTEGER	NULL	达成条件值

3.2.6 garden_states 表 - 花园状态表

```

CREATE TABLE garden_states (
    state_id INTEGER PRIMARY KEY AUTOINCREMENT,
    user_id INTEGER NOT NULL DEFAULT 1,
    habit_id INTEGER NOT NULL,
    plant_growth INTEGER DEFAULT 0,
    plant_health INTEGER DEFAULT 100,
    last_watered TEXT,
    stage INTEGER DEFAULT 1,

```

```

    FOREIGN KEY (user_id) REFERENCES users(user_id) ON DELETE CASCADE,
    FOREIGN KEY (habit_id) REFERENCES habits(habit_id) ON DELETE CASCADE
);

CREATE INDEX idx_garden_user_id ON garden_states(user_id);
CREATE INDEX idx_garden_habit_id ON garden_states(habit_id);

```

字段	类型	约束	说明
state_id	INTEGER	PK, AUTO	状态唯一标识
user_id	INTEGER	FK, NOT NULL	关联用户
habit_id	INTEGER	FK, NOT NULL	关联习惯
plant_growth	INTEGER	DEFAULT 0	生长进度(0-100)
plant_health	INTEGER	DEFAULT 100	健康度(0-100)
last_watered	TEXT	NULL	最后浇水日期
stage	INTEGER	DEFAULT 1	生长阶段(1-5)

4. 模块设计

4.1 包结构

```

com.habitbloom
├── HabitBloomApplication.java      # Application类
├── ui/
│   ├── splash/                   # 启动页
│   │   └── SplashActivity.java
│   ├── main/                     # 主页面
│   │   ├── MainActivity.java
│   │   └── MainviewModel.java
│   ├── garden/                  # 花园模块
│   │   ├── GardenFragment.java
│   │   ├── GardenViewModel.java
│   │   └── PlantCardAdapter.java
│   ├── habit/                  # 习惯模块
│   │   ├── HabitListFragment.java
│   │   ├── HabitDetailActivity.java
│   │   ├── CreateHabitActivity.java
│   │   ├── HabitViewModel.java
│   │   └── HabitAdapter.java
│   ├── stats/                  # 统计模块
│   │   ├── StatsFragment.java
│   │   ├── StatsViewModel.java
│   │   └── AchievementAdapter.java
│   └── settings/                # 设置模块
│       ├── SettingsFragment.java
│       └── SettingsViewModel.java
└── data/                      # 数据层
    └── database/                # 数据库

```

```

|   |   └── AppDatabase.java
|   |   └── entity/          # 实体类
|   |       |   └── User.java
|   |       |   └── Habit.java
|   |       |   └── HabitRecord.java
|   |       |   └── Reminder.java
|   |       |   └── Achievement.java
|   |       └── GardenState.java
|   └── dao/                # 数据访问
|       └── UserDao.java
|       └── HabitDao.java
|       └── HabitRecordDao.java
|       └── ReminderDao.java
|       └── AchievementDao.java
|       └── GardenStateDao.java
└── repository/           # 数据仓库
    └── HabitRepository.java
    └── RecordRepository.java
    └── AchievementRepository.java
    └── UserRepository.java
└── model/                # 数据模型
    └── HabitwithRecords.java
    └── StatsData.java
└── notification/         # 通知模块
    └── ReminderManager.java
    └── ReminderReceiver.java
    └── NotificationHelper.java
└── utils/                # 工具类
    └── DateUtils.java
    └── PlantUtils.java
    └── ExportUtils.java
    └── Constants.java
└── widget/               # 自定义控件
    └── PlantView.java
    └── Progresscircleview.java
    └── CalendarView.java

```

4.2 核心类设计

4.2.1 HabitRepository - 习惯数据仓库

```

public class HabitRepository {
    private final HabitDao habitDao;
    private final HabitRecordDao recordDao;
    private final GardenStateDao gardenDao;
    private final ExecutorService executor;

    // 获取所有活跃习惯
    public LiveData<List<Habit>> getAllActiveHabits();

    // 创建习惯
    public void createHabit(Habit habit, callback callback);

    // 打卡
    public void checkIn(long habitId, String notes, callback callback);
}

```

```

// 获取习惯详情
public LiveData<HabitwithRecords> getHabitwithRecords(long habitId);

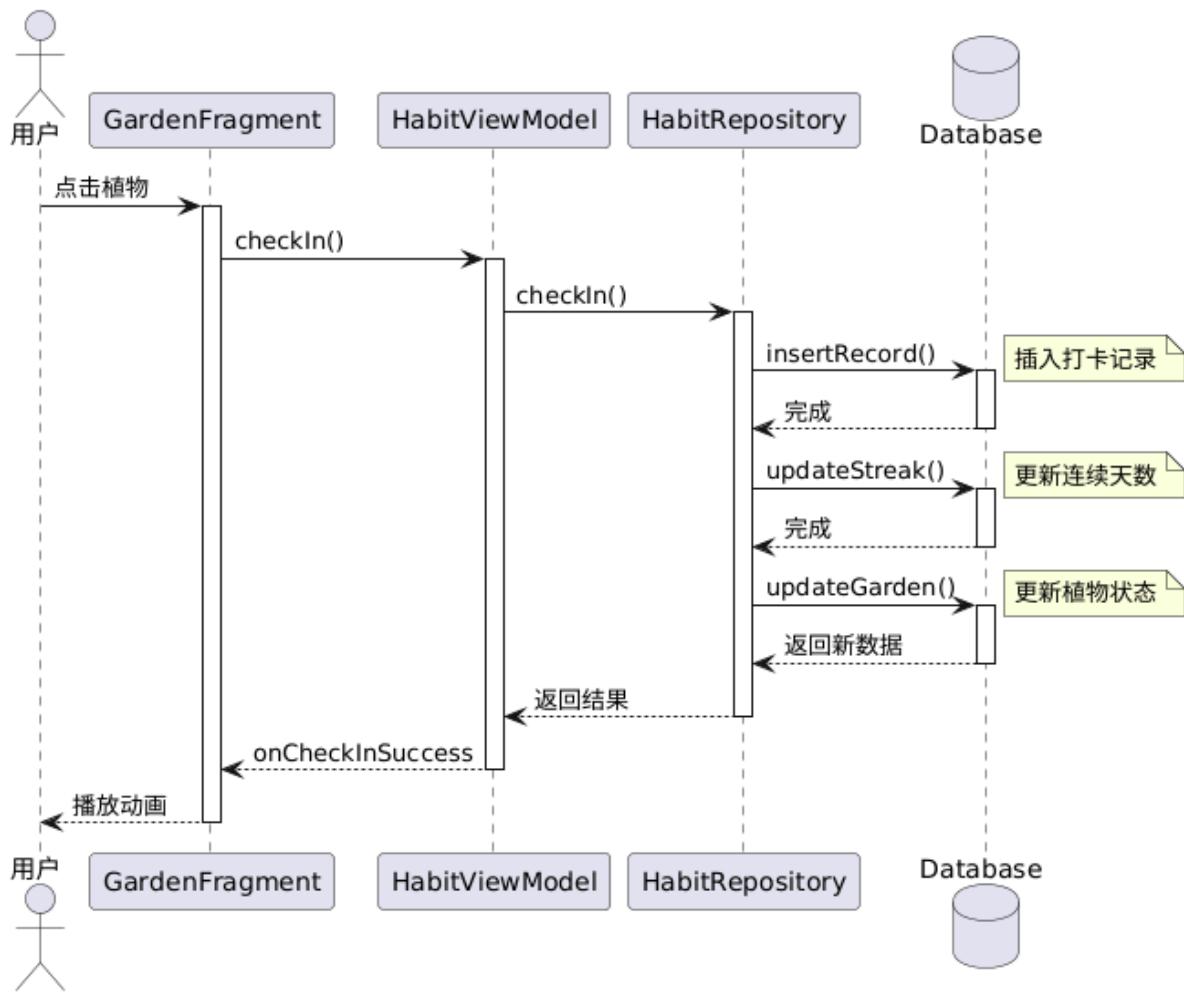
// 更新连续天数
private void updateStreak(long habitId);

// 更新植物状态
private void updatePlantGrowth(long habitId, int streak);
}

```

4.3 关键流程设计

4.3.1 打卡流程时序图



5. 安全设计

5.1 数据安全

- 所有数据存储在本地，不上传至服务器
- 导出数据使用 JSON 格式，便于用户备份
- 数据库使用 SQLite，无需额外加密

5.2 权限说明

权限	用途
RECEIVE_BOOT_COMPLETED	开机启动提醒服务
SCHEDULE_EXACT_ALARM	精确定时提醒
POST_NOTIFICATIONS	发送通知
WRITE_EXTERNAL_STORAGE	导出数据文件
READ_EXTERNAL_STORAGE	导入数据文件
VIBRATE	震动反馈

6. 版本历史

版本	日期	说明
1.0	2026-01-13	初始版本