DXW团队

眠眠萌

软件设计文档

1. 技术选型

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 睡眠检测应用 | 理由 | 备注 |
| 1 终端支持 | 🗹 Android | 安卓系统的智能手机普及度较高，功能完备，便于应用的开发、测试和分发。 | 暂无跨平台 |
| 1.1 开发语言框架 | 🗹 Java  🗹 Android SDK (25) | 使用较新的SDK 进行开发，能够获得最新的特性和最好的性能，同时使用Support Library package保持向下的兼容性。 |  |
| 1.2 使用权限 | 🗹 加速度传感器  🗹 陀螺仪  🗹 方向传感器  🗹 麦克风 | 为了检测睡眠质量，需要在睡眠过程中，使用这些传感器提供的数据。 |  |
| 2 开发平台与工具 | 🗹 Windows  🗹 macOS |  |  |
| 2.1 IDE | 🗹 Android Studio | 安卓开发的优先选择，全平台适用 | 2.2.X版本与2.3.X版本有少量不兼容 |
| 2.2 测试平台 | 🗹 AVD  🗹 genymotion  🗹 开发者的手机 | 为了性能，可以在开发时尽量使用genymotion进行评估，为了检测睡眠，必须要使用实机进行体验测试。 |  |
| 2.3 源代码管理 | 🗹 Github | 最流行的面向开源项目的托管平台，便于版本管理和工作统计。 |  |

1. 架构设计

安卓项目的主要部分分层如下：

├─androidTest

│ └─java

│ └─com

│ └─appmon

│ └─sleepmon

├─main

│ ├─assets

│ ├─java

│ │ └─com

│ │ └─appmon

│ │ └─sleepmon

│ │ ├─Fragments

│ │ ├─Model

│ │ ├─Receiver

│ │ ├─Utility

│ │ └─Widget

│ ├─jniLibs

│ └─res

│ ├─drawable

│ ├─layout

│ ├─mipmap

│ ├─raw

│ ├─values

│ └─xml

└─test

└─java

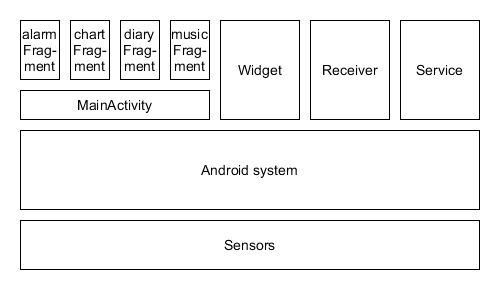
└─com

└─appmon

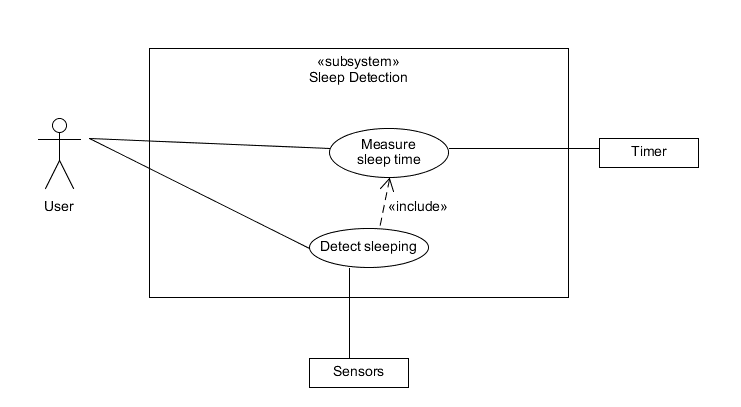
└─sleepmon

|  |  |
| --- | --- |
| **模块** | **内容** |
| androidTest | 样机测试代码 |
| Fragments | 界面片段的业务逻辑 |
| Model | 数据的容器类 |
| Receiver | 安卓中的接收器的组件类 |
| Utility | 常用的工具类，如数据库、图表组件 |
| Widget | 音乐播放器的桌面小部件 |
| jniLibs | 引用的第三方二进制文件 |
| res | 资源文件，如布局、常数、二进制资源文件等 |
| Test | 本机测试代码 |

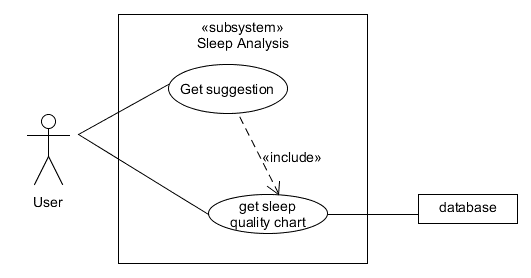
架构层次图：



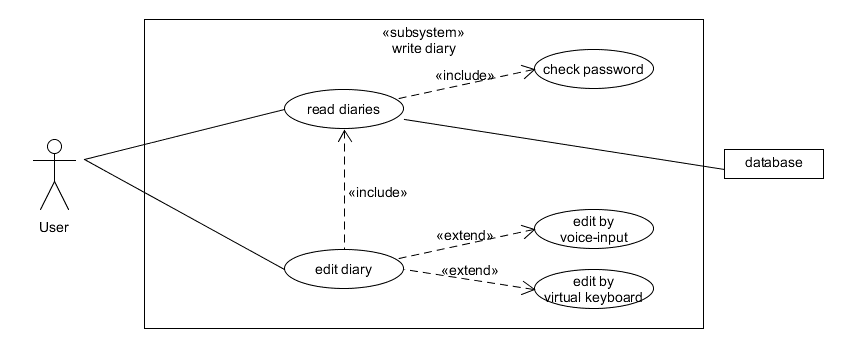
1. 用例
2. 睡眠检测



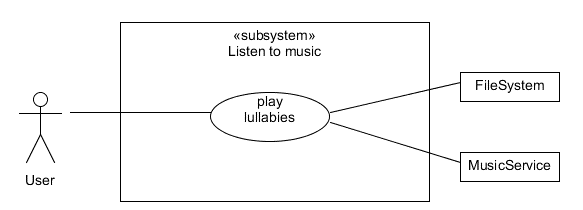
1. 睡眠分析



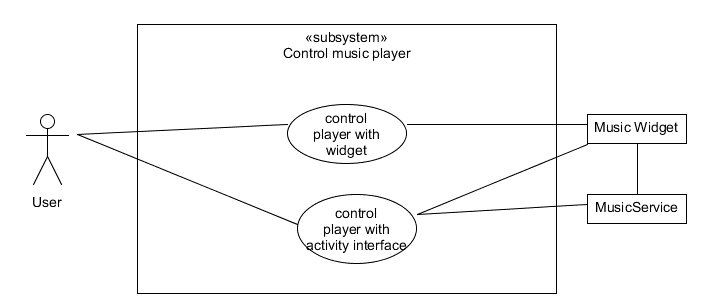
1. 写日记



1. 播放催眠曲



1. 使用桌面小部件



1. 软件设计技术
2. Structure Programming

使用了android.support.v4.app.Fragment进行功能模块和界面的划分，使得划分更清晰，同时易于分工，并通过函数调用的方式，使得每个类的方法尽量专能化，缩短代码块的长度，增加可读性。如各个Fragment的init()或prepare()函数

1. Object-Oriented Programming

通过对MusicService的封装，在Fragments进行访问的时候，仅使用其暴露在外的接口，来达到在后台播放音乐的同时能使界面上的操作传递给MusicService的效果，隐藏了其他不必要的信息。又如音乐播放中使用的music类

1. Design Patterns

如安卓中常见的onClickListener为监听者模式，databaseHelper实际上是一个单例模式。以及安卓中，各种实现接口的类，是一种模板模式，如调用了super.onCreate()的数据库、activity等。

版本更新记录

|  |  |
| --- | --- |
| 17/05/18 | 初始版本 |
| 17/06/24 | 最终更新 |