**应用系统设计文档**

1. 引言

本文档描述了番茄钟程序的设计，包括软件架构、模块设计、数据库设计和用户界面设计等方面的内容。

2. 软件架构

番茄钟程序采用MVC（Model-View-Controller）架构，以实现界面、逻辑和数据的分离。

2.1 模块划分

番茄钟程序包含以下主要模块：

用户界面模块（View）：负责展示用户界面，接收用户输入，并将用户操作传递给控制器模块。

控制器模块（Controller）：负责处理用户操作，管理任务和计时器，与模型模块进行交互。

模型模块（Model）：负责管理任务数据、历史数据和成就信息，提供对数据库的读写操作。

2.2 模块间通信

模块间的通信通过消息传递方式进行，使用回调函数和事件驱动的机制实现模块间的交互和数据传递。

3. 模块设计

3.1 用户界面模块（View）

设计用户界面，包括任务创建、系统设置、计时器和统计图表等界面。

实现用户输入的响应，将用户操作事件传递给控制器模块。

3.2 控制器模块（Controller）

响应用户操作事件，处理用户的任务创建、计时器控制和系统设置等操作。

调用模型模块的接口，读取和更新任务数据、历史数据和成就信息。

控制计时器的启动、暂停和重置。

3.3 模型模块（Model）

设计数据库结构，包括任务表、历史数据表和成就表。

提供数据库的读写接口，包括任务的创建、更新和查询，历史数据的记录和查询，成就信息的获取和更新。

实现数据的持久化，将数据保存到SQLite数据库中。

4. 数据库设计

番茄钟程序使用SQLite数据库来存储用户的任务数据、历史数据和成就信息。

4.1 任务表

字段：任务ID、任务名称、任务类型、任务时长、休闲时长、开始时间、结束时间、状态等。

主键：任务ID。

4.2 历史数据表

字段：日期、番茄数量、完成任务数量等。

主键：日期。

4.3 成就表

字段：成就ID、成就名称、获得条件、获得时间等。

主键：成就ID。

5. 用户界面设计

设计用户界面以提供用户友好的操作和交互体验。

6. 总体流程

整体流程如下：

用户创建任务并设置相关属性。

用户启动计时器开始任务，倒计时进行。

用户可以暂停、继续和重置计时器。

如果用户未暂停或退出计时器，则该任务将被计为一个番茄。

用户可以点击统计图标查看历史数据统计信息。

系统根据用户获得的番茄数量颁发成就勋章。

7. 接口定义

详细描述模块之间的接口和函数调用方式，包括视图层、控制层和模型层之间的接口。

以上是一个设计文档的示例，你可以根据自己的实际需求和技术选择进行调整和补充。设计文档应该清晰地描述软件的架构、模块划分、接口设计和数据库设计等方面的内容，以便团队成员理解和实现。