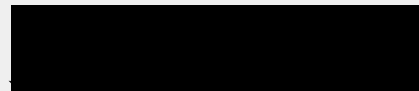


软件工程实践课程汇报

软件实现汇报：华为Sports



2024年4月29日

大纲

项目概况

开发过程

功能展示

项目概况

项目名称：华为Sports--智能骑行助手

主要功能：

1. 骑行记录

- 开始骑行：用户开始骑行时手动启动骑行时长、时速、均速、距离。
- 自动暂停：骑行过程中自动识别停止情况，如等红灯、喝水等，暂停运动数据记录。
- 暂停\继续骑行：用户可以手动暂停\恢复骑行记录。
- 结束骑行：用户结束骑行时停止记录。

2. 记录查看

- 骑行结束后查看本次骑行活动的数据

主要非功能需求：安全性、可靠性、易用性。



开发过程—权限问题及解决方案

开发过程： 采用敏捷开发，每个迭代结束都交付可运行的软件增量。

* 权限

设备能力类

- ☒ 设备基础信息 ! 包含手机与穿戴设备应用间数...
获取已配对穿戴设备的随机标识符、名称、电量、连接状态、应用安装状态等信息，可发送音乐等文件到穿戴设备
- ☒ 消息通知 ! 本消息通知为模板化的消息推...
发送消息通知给已配对穿戴设备，或接受来自已配对穿戴设备的消息通知

* 申请数据权限及使用说明

[点击上传](#)

请按照申请数据权限及使用说明，按照[样例](#)填写。如果附应用模拟截图可加快审核，请确认截图可能包含的第三方信息已经获取了权利人授权和许可。

请上传小于10M的EXCEL文件

* 用户授权路径说明

[点击上传](#)

Wear Engine

查看协议 | 开发指南 | FAQ | 示例代码 | CodeLab

申请 Wear Engine 服务

3. 点击“移动应用”并选择产品后，勾选必需申请的权限（个人开发者只可申请设备基础信息、消息通知两个基本的权限）。

说明
请选择手机侧应用，穿戴设备侧应用默认不需要申请权限。

图3 勾选需要申请的权限（个人开发者只可申请设备基础信息、消息通知两个基本的权限）

权限

设备能力类

- ☒ 设备基础信息 ! 包含手机与穿戴设备应用间数...
获取已配对穿戴设备的随机标识符、名称、电量、连接状态、应用安装状态等信息，可发送音乐等文件到穿戴设备
- ☒ 消息通知 ! 本消息通知为模板化的消息推...
发送消息通知给已配对穿戴设备，或接受来自已配对穿戴设备的消息通知
- ☐ 穿戴用户状态
获取穿戴设备用户的佩戴状态、佩戴状态等信息
- ☐ 运动传感器
控制，获取已配对穿戴设备的加速度、陀螺仪等运动传感器信息
- ☐ 人体传感器 ! 仅支持Android 10及以上版本设备
获取已配对穿戴设备的人体传感器信息

问题及解决方案：

1. 华为穿戴设备SDK不给个人开放陀螺仪、gps等权限。于是该用百度地图SDK。

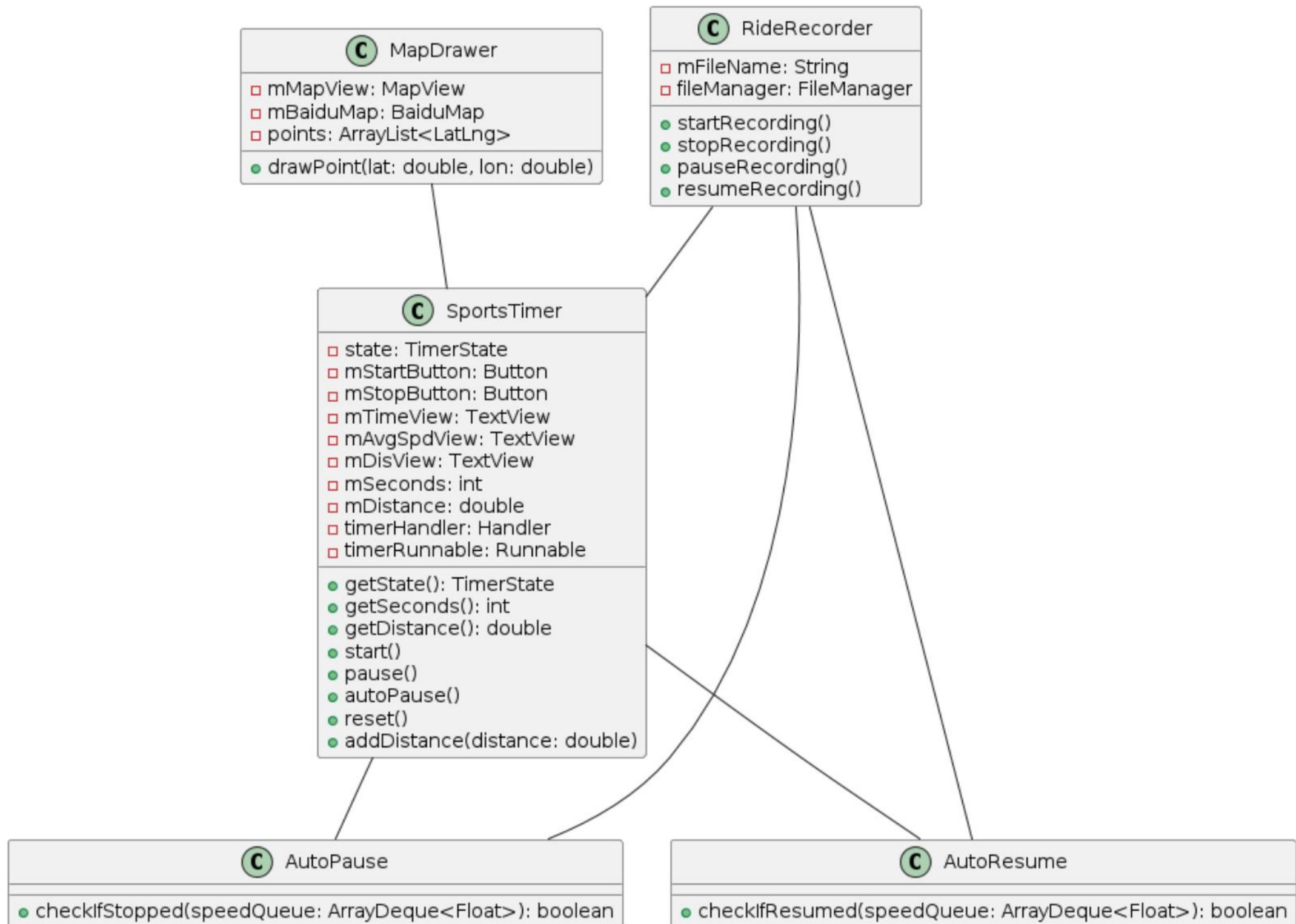
2. 我们需要的地图SDK只有Harmony NEXT版本，不支持现在的鸿蒙设备。想在鸿蒙设备上使用SDK，只能用安卓SDK。

开发过程：用例1

用户故事：

作为一个骑行爱好者，我想要能够在开始骑行记录时实时查看地图轨迹、速度、距离和时间等信息，同时能够手动或自动暂停记录，并在需要时恢复记录，以便于更好地了解自己的骑行状态、提高骑行效率和享受骑行乐趣。

开发过程: 用例1-UML用例图



开发过程：用例1-UML顺序图



开发过程： 用例1-设计原则

该用例满足了以下面向对象的设计原则：

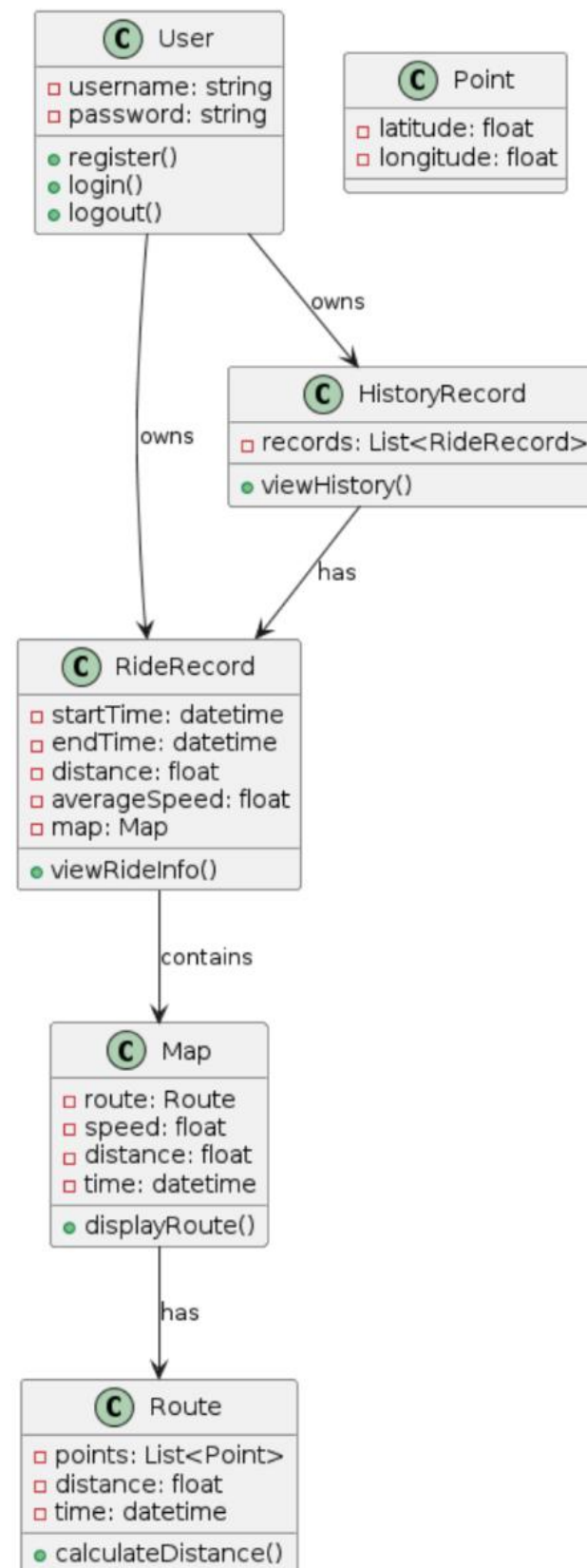
- 单一职责原则 (**SRP**)：每个类都有一个单一的责任。例如，`MainActivity` 负责处理用户界面和与用户交互，`SportsTimer` 负责计时器逻辑，`RecordActivity` 负责记录骑行数据。
- 开放-封闭原则 (**OCP**)：系统应该对扩展开放，对修改封闭。用例描述了用户希望实时看到骑行数据并控制记录的需求，这些功能可以通过添加新的功能模块来扩展，而不需要修改现有代码。
- 里氏替换原则 (**LSP**)：父类的对象可以被其子类的对象替换而不影响程序的正确性。例如，`MyLocationListener` 是 `BDAbstractLocationListener` 的子类，可以替换使用，而不会影响程序的其他部分。
- 依赖倒置原则 (**DIP**)：高层模块不应依赖于低层模块，二者都应该依赖于抽象。例如，`MainActivity` 通过接口与 `SportsTimer` 进行通信，而不是直接依赖于具体的实现类。
- 接口隔离原则 (**ISP**)：客户端不应该强制依赖于它们不使用的接口。在这个用例中，`MainActivity` 和 `RecordActivity` 只依赖于他们需要使用的接口和功能，而不受其他模块的影响。

开发过程：用例2

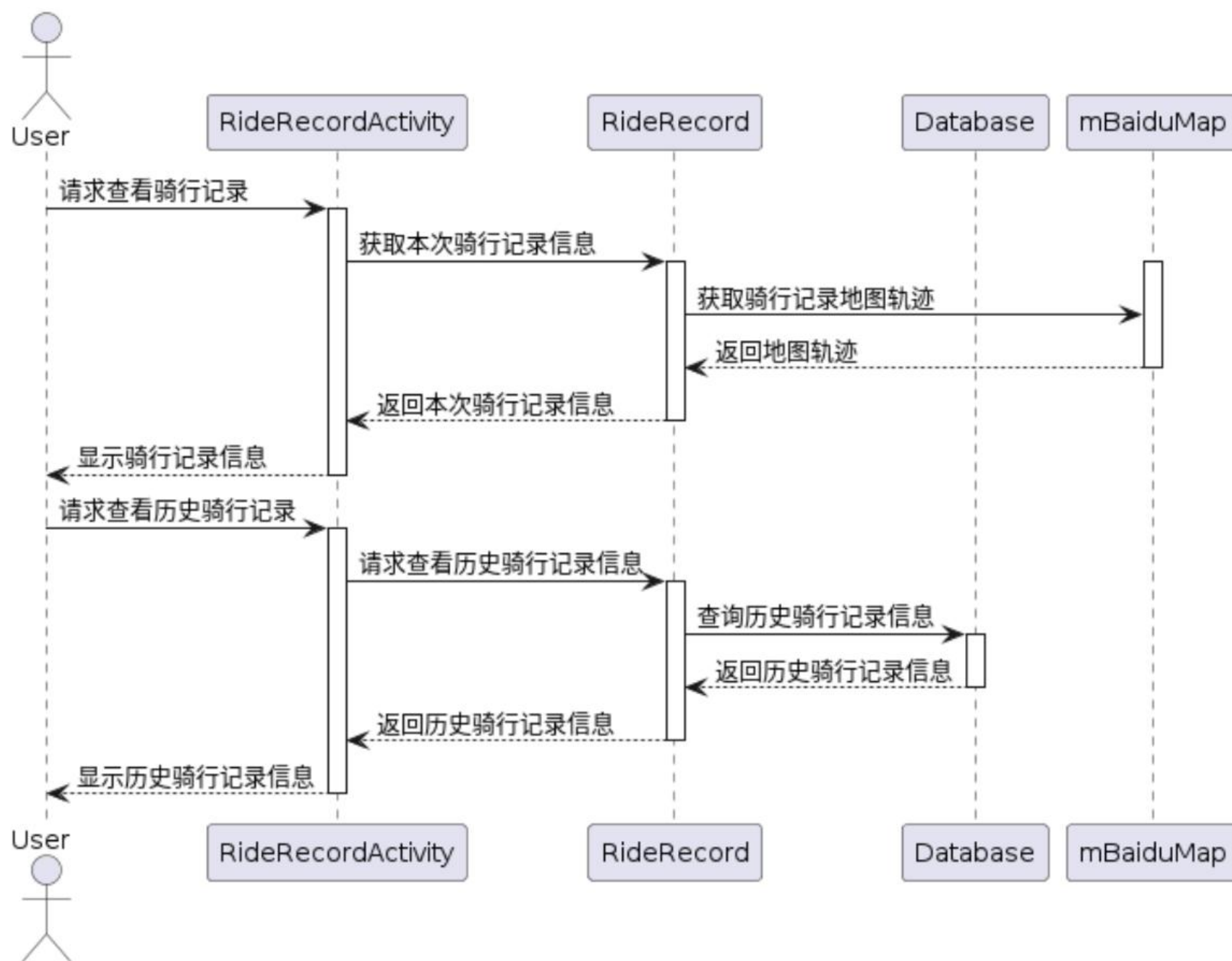
用户故事：

作为一个骑行爱好者，我想要在结束骑行后能够方便地查看本次骑行的轨迹地图、速度、距离、时间等信息，以便于了解自己的骑行表现和记录骑行历史。这样能够帮助我评估自己的骑行进步，并且在需要时能够分享和回顾骑行经历，提升骑行体验和享受。

开发过程: 用例2-UML用例图



开发过程：用例2-UML顺序图



开发过程：用例2-设计原则

该用例满足了以下面向对象的设计原则：

- **单一职责原则**：每个类都专注于实现单一的功能，例如RecordActivity.java负责展示骑行记录的详细信息，FileManager.java负责管理文件的读写操作，各个类之间的职责清晰明确。
- **开放封闭原则**：系统中的各个模块都是可扩展的，例如在RecordActivity.java中，可以通过添加新的方法或组件来增强用户查看骑行记录信息的功能，而不需要修改已有的代码。
- **依赖倒置原则**：高层模块不应该依赖于低层模块，而是应该依赖于抽象。在这个用例中，RecordActivity.java等高层模块依赖于抽象的数据访问接口，而不是依赖于具体的实现，这样可以降低模块之间的耦合度。
- **接口隔离原则**：接口应该是小而专门的，不应该包含不必要的方法。在这个用例中，各个类都只暴露了必要的接口，例如FileManager.java只提供了文件读写操作的接口，而不包含其他无关的方法。

开发过程--界面设计

地图		
0.00 时速(km/h)		
0.00 里程(km)	00:00:00 运动时间	0.00 运动均速(km/h)
开始		

地图		
0.00 时速(km/h)		
0.00 里程(km)	00:00:00 运动时间	0.00 运动均速(km/h)
结束	暂停	

地图		
0.00 时速(km/h)		
0.00 里程(km)	00:00:00 运动时间	0.00 运动均速(km/h)
结束	继续	

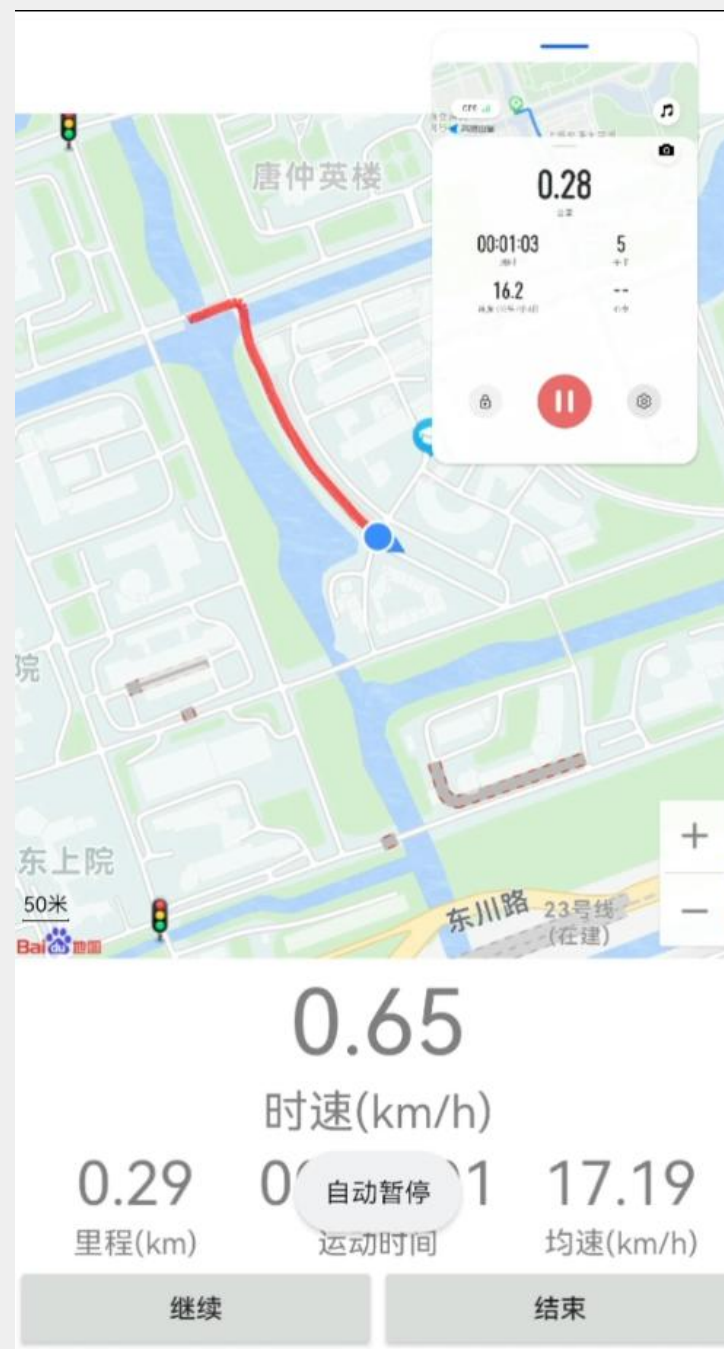


功能展示(与Keep、华为运动健康对比)

keep



华为运动健康



骑行记录查看



*具体演示视频参见canvas作业-第四次实践课分组Demo提交内容

谢谢大家