CityMapper 项目文档-User1

组别：大模型组

1.引言内容如下：

-目的：

本文档旨在为City Mapper交通导航软件的设计、开发和测试提供全面的需求描述。通过明确系统的功能需求、外部接口需求以及非功能需求，确保开发团队能够根据具体要求实现高效、可靠且用户友好的产品。此外，文档还详细说明了系统的关键用例和数据结构，以帮助相关利益方（如产品经理、开发人员和测试人员）对系统的目标和范围形成一致的理解。

-范围：

本文档适用于City Mapper交通导航软件的整个生命周期，包括需求分析、设计、开发、测试和维护阶段。系统的主要目标是为用户提供多种交通方式的路线规划服务，整合实时交通信息和兴趣点推荐，从而帮助用户高效地完成城市中的出行任务。文档涵盖了以下内容：

1.系统的功能需求，包括路线规划、实时交通更新、兴趣点推荐等核心模块。

2.外部接口需求，涉及第三方API和服务的对接。

3.非功能需求，例如性能、安全性和可靠性。

4.数据模型和实体关系图（ER图），用于描述系统中关键数据实体及其关系。

5.完整的用例描述，展示系统与用户之间的交互过程。

-定义、缩写与缩略语：

为了便于理解本文档内容，以下列出了一些重要术语、缩写和缩略语：

ER图：实体关系图（Entity-Relationship Diagram），用于描述数据库中的实体及其关系。

POI：兴趣点（Point of Interest），指地图上具有特定意义的地点，如餐馆、公园、商店等。

API：应用程序编程接口（Application Programming Interface），用于不同软件系统之间的交互。

UI：用户界面（User Interface），指用户与系统交互的可视化界面。

SQL：结构化查询语言（Structured Query Language），用于管理和操作关系型数据库。

GPS：全球定位系统（Global Positioning System），用于获取用户的地理位置信息。

ETA：预计到达时间（Estimated Time of Arrival），用于表示用户到达目的地的时间预测。

-参考文献：

以下文献和资源为本文档的编写提供了理论支持和技术参考：

·Sommerville, I. (2019). Software Engineering (10th Edition). Pearson Education.

提供了软件需求工程的基本理论和实践方法。

·Fowler, M. (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition). Addison-Wesley.

涵盖了UML建模的核心概念，尤其是用例图和ER图的设计原则。

·IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications.

定义了软件需求文档的标准格式和内容规范。

·City Mapper官方网站及相关技术文档。

提供了关于交通导航软件的实际案例和功能设计灵感。

·Google Maps Platform API Documentation.

为地图服务和实时交通数据的集成提供了技术支持。

1. 总体概述内容如下：

-产品视角：

City Mapper是一款面向城市用户的智能交通导航软件，旨在为用户提供高效、便捷的出行解决方案。作为一款集成多种交通方式和实时动态信息的工具，City Mapper不仅关注路线规划的核心功能，还通过兴趣点推荐、用户偏好设置等附加功能提升用户体验。系统从用户需求出发，整合了公共交通、步行、骑行等多种出行模式，并结合实时交通数据和天气信息，帮助用户在复杂的城市环境中快速找到最佳出行方案。

City Mapper在整个生态系统中扮演着连接用户与城市交通资源的角色。它依赖于第三方API（如地图服务、实时交通和天气数据）来获取动态信息，同时也需要一个稳定的后台数据库来存储用户信息、历史记录和偏好设置。

-产品功能：

City Mapper的主要功能模块包括：

1.路线规划

支持多种交通方式（公共交通、步行、骑行）的路线计算。

提供按距离、时间和用户偏好排序的多种推荐方案。

显示预计到达时间（ETA）和实际时间对比。

2.实时交通更新

动态显示交通拥堵状况、公共交通延误信息。

根据实时数据调整路线规划，提供最优建议。

3.兴趣点推荐

基于用户位置推荐附近的餐馆、公园、商店等兴趣点。

提供兴趣点的详细信息（地址、评分、分类）。

4.用户偏好设置

允许用户自定义出行偏好（如优先选择步行或公共交通）。

保存常用起点和终点，提升使用效率。

5.多语言支持

界面支持多语言切换，满足国际化用户需求。

-用户特征：

City Mapper的目标用户主要包括以下几类：

城市通勤者：日常工作、学习需要频繁往返于城市不同区域的人群，他们对高效的路线规划和实时交通信息有较高需求。

游客：初到城市的游客，需要快速熟悉当地的交通网络并探索周边兴趣点。

环保倡导者：倾向于选择步行、骑行等低碳出行方式的用户群体。

技术熟练型用户：习惯使用智能手机和导航软件的年轻用户，能够快速上手复杂的软件功能。

用户普遍具备一定的智能手机操作能力，并对导航软件的基本功能有初步了解。此外，部分用户可能对个性化设置和实时动态信息有较高的期待。

-竞品分析：

当前市场上存在多个成熟的交通导航软件，以下是主要竞品及其特点分析：

Google Maps

优势：覆盖范围广，地图数据详尽，实时交通更新准确。

劣势：对某些小众交通方式（如共享单车）支持有限；界面相对复杂，新手用户可能需要较长时间适应。

Apple Maps

优势：与苹果生态系统深度集成，界面简洁直观。

劣势：在非苹果设备上的兼容性较差；实时交通数据更新速度稍逊于Google Maps。

高德地图（中国）

优势：本地化程度高，特别适合中国市场；支持多种支付方式（如打车服务）。

劣势：国际化程度较低，海外城市的覆盖率不足。

Waze

优势：以社区驱动为核心，用户可以报告路况、事故等信息。

劣势：对公共交通的支持较弱；界面设计较为繁杂。

City Mapper的优势在于专注于城市出行场景，整合了多种交通方式和实时动态信息，同时提供了个性化的兴趣点推荐功能。相比上述竞品，City Mapper更注重用户体验的定制化和智能化。

-设计约束：

在City Mapper的设计和开发过程中，需遵循以下约束条件：

性能要求：路线规划的响应时间不得超过3秒，以确保用户体验流畅。

数据隐私：用户的位置数据和个人信息必须加密存储，并符合GDPR等相关隐私法规。

第三方API依赖：系统的实时交通、地图服务和天气信息均依赖于第三方API，因此需确保API接口的稳定性和可靠性。

跨平台兼容性：软件需支持iOS和Android两大主流操作系统，并保证在不同设备上的界面一致性。

可扩展性：系统架构需具备良好的可扩展性，以便未来新增功能模块（如拼车服务、共享单车支持）。

-假设与依赖关系：

在City Mapper的设计和实现过程中，存在以下假设和依赖关系：

假设：

用户设备具备GPS功能，并允许应用访问地理位置信息。

第三方API（如地图服务、实时交通和天气数据）能够持续提供稳定的服务。

用户具备基本的智能手机操作能力，能够理解并使用核心功能。

依赖关系：

后台数据库的稳定性：用户信息、历史记录和偏好设置需存储在可靠的数据库中。

第三方服务的可用性：例如Google Maps API、OpenWeatherMap API等，若这些服务中断，将直接影响系统的功能表现。

网络连接：系统依赖互联网连接以获取实时数据和动态信息，因此需假设用户在大多数情况下能够访问稳定的网络环境。

通过明确以上假设和依赖关系，City Mapper能够在设计阶段充分考虑潜在风险，并制定相应的应对措施，以确保系统在实际运行中的可靠性和稳定性。

1. 功能需求

1. 路线规划

描述：用户输入起点和终点，系统根据交通方式偏好生成多种出行方案。

输入：

起点位置（手动输入或通过GPS获取）。

终点位置（手动输入或通过地图选择）。

用户偏好设置（如优先选择公共交通、步行或骑行）。

输出：

多种路线方案列表，包括每条路线的详细步骤、预计耗时、距离及交通方式组合。

2. 实时交通更新

描述：系统动态获取实时交通数据，并根据当前路况调整路线规划。

输入：

当前用户的地理位置（通过GPS获取）。

第三方API提供的实时交通信息（如拥堵路段、公交延误等）。

输出：

更新后的路线规划方案，标注受影响路段及其替代路径。

预计到达时间（ETA）的动态调整。

3. 兴趣点推荐

描述：基于用户当前位置和历史行为，系统推荐附近的兴趣点（POI）。

输入：

用户的当前位置（通过GPS获取）。

用户的历史行为数据（如常访问的兴趣点类别）。

第三方API提供的兴趣点数据（如餐馆、公园、商店等）。

输出：

列表形式的兴趣点推荐，包含名称、地址、评分及分类信息。

4. 用户偏好设置

描述：用户可自定义出行偏好，系统根据设置优化路线规划。

输入：

用户选择的交通方式优先级（如步行 > 公共交通 > 骑行）。

常用起点和终点的保存选项。

输出：

个性化设置存储至用户账户中，供后续路线规划使用。

5. 多语言支持

描述：系统支持多语言切换，满足国际化用户需求。

输入：

用户选择的语言选项（如中文、英文、法语等）。

输出：

界面语言切换为用户选择的语言，包括菜单、提示信息及兴趣点详情。

6. 数据导入与同步

描述：用户可将常用地址或历史记录从外部文件导入系统，并支持云端同步。

输入：

Excel文件（格式：.xlsx），包含字段如地址名称、经纬度、标签等。

用户登录账户信息（用于云端同步）。

输出：

导入的地址数据存储至用户账户中，并可在不同设备间同步。

7. 历史记录查询

描述：用户可查看过去的出行记录，包括起点、终点、路线及时间。

输入：

用户登录账户信息。

查询条件（如日期范围、特定地点）。

输出：

历史记录列表，包含每次出行的详细信息。

8. 实时天气信息展示

描述：系统结合天气数据，为用户提供出行相关的天气建议。

输入：

用户的当前位置（通过GPS获取）。

第三方API提供的实时天气信息（如温度、降水概率、风速等）。

输出：

当前及未来几小时的天气预报，结合出行建议（如是否带伞、是否适合骑行）。

9. 通知提醒功能

描述：系统根据用户行程发送提醒通知，确保用户按时出发或调整计划。

输入：

用户设定的出发时间或到达时间。

实时交通数据（用于动态调整提醒时间）。

输出：

推送通知，内容包括建议出发时间、路线变更提醒等。

10. 地图交互功能

描述：用户可通过地图界面直接选择起点、终点或兴趣点。

输入：

用户在地图上的点击操作。

输出：

所选位置的详细信息（如地址、经纬度）显示在界面上，并作为路线规划的输入参数。

11. 路线分享功能

描述：用户可将规划好的路线分享给其他用户或平台。

输入：

已生成的路线方案。

分享目标（如短信、邮件、社交媒体链接）。

输出：

包含路线信息的分享链接或文本内容，发送至指定目标。

1. 外部接口

-描述软件与外部系统接口需求：

1. 地图服务API

描述：获取地图数据、地理编码及逆地理编码服务。

接口类型：RESTful API

输入：

用户提供的地址或经纬度坐标。

请求的地理范围（如城市、区域）。

输出：

标准化的地图瓦片数据。

地址与经纬度之间的转换结果。

依赖方：Google Maps API 或 OpenStreetMap API。

2. 实时交通API

描述：获取实时交通状况，包括道路拥堵、公共交通延误等信息。

接口类型：RESTful API 或 WebSocket

输入：

用户当前位置或指定路线的起点和终点。

查询的时间范围。

输出：

道路拥堵等级、预计通行时间。

公共交通（如地铁、公交）的实时到站时间和延误信息。

依赖方：TomTom Traffic API 或 HERE Traffic API。

3. 天气数据API

描述：获取用户所在位置的实时天气信息及未来几小时的天气预报。

接口类型：RESTful API

输入：

用户当前位置的经纬度。

查询的时间范围（如当前天气或未来3小时）。

输出：

温度、湿度、降水概率、风速等详细天气信息。

是否适合特定出行方式的建议（如是否适合骑行）。

依赖方：OpenWeatherMap API 或 WeatherAPI。

4. 兴趣点（POI）数据API

描述：获取用户附近的兴趣点信息（如餐馆、公园、商店等）。

接口类型：RESTful API

输入：

用户当前位置的经纬度。

查询的兴趣点类别（如餐饮、娱乐、购物）。

输出：

列表形式的兴趣点信息，包括名称、地址、评分、分类及距离。

依赖方：Foursquare API 或 Google Places API。

5. 用户账户管理API

描述：支持用户登录、注册及云端数据同步功能。

接口类型：RESTful API

输入：

用户登录信息（如邮箱、密码）。

用户上传的数据（如常用地址、偏好设置）。

输出：

登录状态及用户账户信息。

同步后的用户数据（如历史记录、偏好设置）。

依赖方：Firebase Authentication 或自建用户管理系统。

6. 推送通知服务API

描述：向用户发送实时通知，包括出发提醒、路线变更等。

接口类型：RESTful API 或 SDK 集成

输入：

用户设备的唯一标识符（如推送令牌）。

通知内容（如建议出发时间、交通延误警告）。

输出：

推送通知成功发送至用户设备。

依赖方：Firebase Cloud Messaging (FCM) 或 Apple Push Notification Service (APNs)。

7. 数据导入与导出API

描述：支持用户上传Excel文件或将数据导出至外部系统。

接口类型：RESTful API 或文件解析库

输入：

用户上传的Excel文件（格式：.xlsx），包含字段如地址名称、经纬度、标签等。

输出：

解析后的结构化数据存储至数据库中。

导出的文件（如CSV格式）供用户下载。

依赖方：第三方文件解析库（如 Apache POI 或 pandas）。

8. 社交媒体分享API

描述：允许用户将路线信息分享至社交媒体平台。

接口类型：OAuth 2.0 授权 + RESTful API

输入：

用户授权的社交媒体账户信息。

路线信息（如起点、终点、预计耗时）。

输出：

成功发布至社交媒体平台的分享内容。

依赖方：Twitter API、Facebook Graph API 或微信分享SDK。

9. 支付网关API（可选）

描述：支持用户通过应用内支付购买高级功能或服务（如无广告模式、高级路线分析）。

接口类型：RESTful API

输入：

用户选择的支付方式（如信用卡、PayPal）。

订单信息（如金额、服务类型）。

输出：

支付成功或失败的状态反馈。

更新后的用户权限（如解锁高级功能）。

1. 非功能需求

-列出软件需要满足的性能指标：

响应时间：

路线规划请求的平均响应时间不超过2秒。

实时交通更新和天气信息的刷新间隔不超过30秒。

并发支持：

系统应支持至少10,000名用户同时在线使用，每秒处理不少于500个API请求。

数据同步延迟：

用户账户数据的云端同步延迟不得超过5秒。

-描述软件的质量属性：

可用性

系统可用性：

系统全年可用性（uptime）不低于99.9%。

关键功能（如路线规划、实时交通更新）在高峰期的故障率低于0.1%。

离线模式：

在无网络连接的情况下，用户可访问最近一次缓存的地图数据和路线规划结果。

多平台支持：

支持iOS、Android及Web端，确保各平台用户体验一致。

3. 安全性

数据加密：

用户敏感数据（如登录信息、支付信息）需在传输过程中使用TLS 1.2及以上协议进行加密。

存储在数据库中的密码需采用强哈希算法（如bcrypt）加密。

身份验证：

使用OAuth 2.0协议进行第三方登录授权，确保用户账户安全。

提供双因素认证（2FA）选项以增强账户保护。

隐私保护：

用户位置数据仅在获得明确同意后收集，并提供随时关闭位置共享的选项。

符合GDPR（通用数据保护条例）或CCPA（加州消费者隐私法案）等隐私法规要求。

4. 可扩展性

模块化设计：

系统采用微服务架构，每个功能模块（如路线规划、实时交通）独立部署，便于扩展和维护。

第三方集成：

外部接口设计灵活，支持快速接入新的地图服务、交通数据源或支付网关。

5. 可靠性

容错能力：

当外部API（如实时交通、天气数据）不可用时，系统应自动切换至备用数据源或显示缓存数据。

提供友好的错误提示信息（如“当前无法获取实时交通信息，请稍后再试”）。

数据备份：

用户数据每日备份至云端存储，确保数据丢失时可快速恢复。

6. 可维护性

日志记录：

记录所有关键操作和错误信息，便于问题排查和性能优化。

日志级别分为DEBUG、INFO、WARN、ERROR，支持按需调整。

监控与报警：

使用监控工具（如Prometheus或Grafana）实时跟踪系统性能指标。

设置阈值报警机制，当响应时间或错误率超标时自动通知运维团队。

7. 用户体验

界面响应速度：

页面加载时间不超过3秒，交互操作（如点击按钮）的反馈时间不超过500毫秒。

无障碍支持：

界面设计符合WCAG 2.1标准，支持屏幕阅读器及高对比度模式。

本地化：

支持至少10种语言，包括但不限于中文、英文、法语、西班牙语、德语等。

根据用户地理位置自动推荐默认语言。

8. 兼容性

设备兼容性：

支持主流移动设备分辨率（如iPhone 14 Pro Max、Samsung Galaxy S23）及浏览器（如Chrome、Safari、Firefox）。

操作系统兼容性：

支持iOS 12及以上版本、Android 8.0及以上版本。

1. 完整的用例描述：

用例1：路线规划

用例名称：用户请求路线规划

参与者：普通用户、地图服务API、实时交通API

前置条件：

用户已打开City Mapper应用。

用户已授权位置访问权限或手动输入起点和终点。

后置条件：

用户获得从起点到终点的最佳路线规划建议。

系统缓存当前查询结果以供离线查看。

主事件流：

用户在界面输入起点和终点信息。

系统调用地图服务API获取地理编码数据。

系统调用实时交通API获取当前交通状况。

系统根据交通状况计算最佳路线（包括步行、骑行、驾车、公共交通等选项）。

系统将路线规划结果显示在地图上，并提供预计行程时间。

用户确认路线并选择出行方式。

异常事件流：

异常1：地图服务API不可用。

系统提示“无法获取地图数据，请检查网络连接或稍后再试”。

用户可尝试重新输入起点和终点。

异常2：实时交通API超时或返回错误。

系统默认使用历史交通数据进行路线规划，并提示“当前交通数据可能不准确”。

用例2：实时交通更新

用例名称：用户接收实时交通更新

参与者：普通用户、实时交通API、推送通知服务API

前置条件：

用户已规划一条路线并开始导航。

用户已授权推送通知权限。

后置条件：

用户接收到实时交通更新，系统动态调整推荐路线。

主事件流：

用户开始导航后，系统每隔30秒向实时交通API发送请求。

实时交通API返回当前交通状况（如拥堵、事故）。

如果检测到更优路线，系统动态调整导航路径。

系统通过推送通知提醒用户路线变更。

异常事件流：

异常1：实时交通API不可用。

系统提示“暂时无法获取实时交通信息”。

导航继续沿用原定路线。

异常2：推送通知失败。

系统在应用内显示路线变更提示，而非通过推送通知。

用例3：兴趣点搜索

用例名称：用户搜索附近的兴趣点

参与者：普通用户、兴趣点数据API

前置条件：

用户已打开City Mapper应用并允许位置访问权限。

后置条件：

用户获得附近兴趣点的列表及详细信息。

主事件流：

用户点击搜索按钮，选择“兴趣点”类别（如餐馆、公园）。

系统调用兴趣点数据API，传递用户当前位置及搜索类别。

兴趣点数据API返回符合条件的兴趣点列表。

系统将兴趣点显示在地图上，并提供名称、地址、评分等详细信息。

用户选择某个兴趣点，系统显示其详细信息（如营业时间、联系方式）。

异常事件流：

异常1：兴趣点数据API返回空结果。

系统提示“未找到符合条件的兴趣点”。

异常2：用户未授予权限访问位置信息。

系统提示“请授予位置权限以使用此功能”。

用例4：用户登录与数据同步

用例名称：用户登录并同步云端数据

参与者：普通用户、用户账户管理API

前置条件：

用户已注册City Mapper账户。

用户设备已连接至互联网。

后置条件：

用户成功登录并同步云端数据（如常用地址、偏好设置）。

主事件流：

用户点击登录按钮，输入邮箱和密码。

系统调用用户账户管理API验证登录信息。

验证通过后，系统从云端下载用户的同步数据。

系统将数据存储至本地设备，并更新界面显示。

用户完成登录并正常使用应用。

异常事件流：

异常1：用户输入错误的邮箱或密码。

系统提示“邮箱或密码错误，请重试”。

异常2：网络连接中断。

系统提示“无法连接至服务器，请检查网络设置”。

用例5：分享路线至社交媒体

用例名称：用户分享路线至社交媒体

参与者：普通用户、社交媒体分享API

前置条件：

用户已规划一条路线。

用户已授权社交媒体账户权限。

后置条件：

路线信息成功发布至用户指定的社交媒体平台。

主事件流：

用户点击分享按钮，选择目标社交媒体平台（如微信、Twitter）。

系统调用社交媒体分享API，传递路线信息（如起点、终点、预计耗时）。

社交媒体分享API将内容发布至用户账户。

系统提示“分享成功”。

异常事件流：

异常1：用户未授权社交媒体权限。

系统提示“请授权社交媒体权限以使用此功能”。

异常2：社交媒体分享API返回错误。

系统提示“分享失败，请稍后再试”。