# Functional Requirement

- 用户注册  
 - 描述：新用户可以通过输入基本信息来创建账户。  
 - 输入：用户的姓名、电子邮件地址、密码。  
 - 输出：新创建的用户账户，用户可登录到系统。  
   
- 用户登录  
 - 描述：已注册的用户通过输入姓名和密码验证身份后登录系统。  
 - 输入：有效的用户名及其关联密码。  
 - 输出：登录到系统。  
  
-创建起始点  
 描述：用户可创建新的起始点坐标和地址信息。  
 输入：起始点的详细信息（如地址、名称）。  
 输出：数据库存储用户创建的起始信息。  
  
-管理起始点  
 描述：管理员可更新或删除系统中的起始点。  
 输入：起始点的标识符、需要更新的信息或删除的操作。  
 输出：系统中的起始点信息更新或被删除。  
  
-创建目的地  
 描述：用户可创建新的目的地坐标和地址信息。  
 输入：目的地的详细信息（如地址、名称）。  
 输出：数据库存储用户创建的目的地信息。  
  
-管理目的地  
 描述：管理员可更新或删除系统中的目的地。  
 输入：目的地的标识符、需要更新的信息或删除的操作。  
 输出：系统中的目的地信息更新或被删除。  
  
-规划路线  
 描述：用户可以根据起点和目的地信息获取由City Mapper应用推荐的路线列表。  
 输入：起点和目的地的详细信息。  
 输出：系统提供的起点到目的地的推荐路径。  
  
-选择交通方式  
 描述：用户可以选择特定的交通方式参与路线规划。  
 输入：用户的选择（如公共交通、步行、骑行等）。  
 输出：系统存储用户的选择，并作为进一步分析的基础。  
  
-距离查询  
 描述：用户可以查询起始点和目的地之间的直线或道路距离。  
 输入：起始点和目的地的详细信息。  
 输出：两地之间的距离信息，以千米或英里表示。  
  
-计算旅行时间  
 描述：用户可以计算从起始点到目的地的预计旅行时间，基于选择的交通方式。  
 输入：起始点、目的地、和交通方式的详细信息。  
 输出：预计的旅行时间。  
  
-设置用户偏好  
 描述：用户可以设置其在交通方式和路线上的偏好。  
 输入：用户偏好信息（如偏好使用公共交通而非单车）。  
 输出：存储于数据库中的用户偏好设置。  
  
-实时更新交通状况  
 描述：系统将实时更新交通状况，并反映在应用中。  
 输入：来自外部的实时交通数据源。  
 输出：数据库中最新的交通状况信息。  
  
-查询时刻表  
 描述：用户可以查询相关交通方式的时刻表信息。  
 输入：查询条件，例如特定交通方式、站点名称和日期等。  
 输出：符合查询条件的时刻表信息。  
  
-探索兴趣点  
 描述：用户可以浏览或搜索各种兴趣点，如景点、餐馆等。  
 输入：兴趣点的查询条件。  
 输出：系统提供的搜索结果，包括兴趣点的名称、地址和类别等信息。  
  
-创建和管理管理员账户  
 描述：为系统管理员创建账户并进行账户管理（更新或删除操作）。  
 输入：管理员账户的详细信息，更新或删除操作。  
 输出：系统中管理员账户信息的更新或删除。  
  
-用户偏好记录  
 描述：系统记录并持续更新用户偏好设置。  
 输入：用户的偏好选择。  
 输出：存储在数据库中的用户偏好记录。  
  
-实时交通状况记录  
 描述：系统存储和更新实时交通状况数据。  
 输入：交通状况的数据源信息。  
 输出：数据库中最新的实时交通状况记录。

# External Description

### 第四章：External Interface  
  
本章节详细描述了City Mapper软件与外部系统（包括硬件、其他软件和服务）之间的接口需求。这些对接包括各种通信协议、数据格式和接口类型，旨在确保系统的功能完整性和数据交换的准确性。  
  
#### 4.1 硬件接口需求  
  
- \*\*硬件兼容性\*\*: City Navigator旨在支持各个平台设备，包括iOS和Android操作系统下的智能手机和平板电脑，确保在不同品牌和型号的设备上均可正常运行。  
  
- \*\*传感器集成\*\*: 利用设备内置的传感器（如GPS），支持实时定位功能，提供地理位置服务。  
  
- \*\*屏幕分辨率及尺寸适应\*\*: 软件能够适应不同的屏幕分辨率及大小，保证在不同尺寸设备上的用户体验一致性。  
  
- \*\*硬件加速\*\*: 支持硬件加速技术（如GPU加速），以确保地图渲染和动画效果的流畅性。  
  
#### 4.2 软件接口需求  
  
本部分描述了City Mapper软件与第三方软件和服务的接口需求，这些接口包括但不限于API调用、数据交换等。  
  
- \*\*第三方地图服务API\*\*: 基于开源或商业地图服务供应商如Google Maps API、高德地图API等，用于获取地理数据如路线规划、地点搜索和实时地图渲染。  
  
 - \*\*接口类型\*\*: RESTful API, Web API  
 - \*\*通信协议\*\*: HTTP/HTTPS  
 - \*\*数据格式\*\*: JSON  
 - \*\*功能\*\*: 路线规划、地点搜索、地形信息及实时地图更新  
  
- \*\*实时交通状况API\*\*: 连接至实时交通数据源，为用户提供当前道路拥堵情况，路线及时调整。  
  
 - \*\*接口类型\*\*: RESTful API, Web socket API  
 - \*\*通信协议\*\*: HTTP/HTTPS, WebSocket  
 - \*\*数据格式\*\*: JSON  
 - \*\*功能\*\*: 实时交通信息查询、路线实时更新  
  
- \*\*交通时刻表更新API\*\*: 集成了公交、地铁及其他公共交通工具的运营时刻表更新服务，确保提供最新公交、地铁线路以及时刻表信息。  
  
 - \*\*接口类型\*\*: RESTful API  
 - \*\*通信协议\*\*: HTTP/HTTPS  
 - \*\*数据格式\*\*: JSON  
 - \*\*功能\*\*: 获取公交、地铁线路以及时刻表信息，时刻表更新  
  
- \*\*兴趣点管理服务API\*\*: 与第三方地点数据库服务集成（如景区、饭店等兴趣点），为用户提供丰富的兴趣点信息和搜索功能。  
  
 - \*\*接口类型\*\*: RESTful API  
 - \*\*通信协议\*\*: HTTP/HTTPS  
 - \*\*数据格式\*\*: JSON  
 - \*\*功能\*\*: 搜索、添加、编辑兴趣点信息，获取附近兴趣点信息  
  
- \*\*用户系统及身份验证API\*\*: 用于支持注册、登录、用户信息管理等功能，可能对接至社交登录（如微信、Facebook等）或其他身份验证机制。  
  
 - \*\*接口类型\*\*: RESTful API  
 - \*\*通信协议\*\*: HTTP/HTTPS  
 - \*\*数据格式\*\*: JSON  
 - \*\*功能\*\*: 用户注册、登录、信息管理、社交登录接口  
  
#### 4.3 数据库接口需求  
  
- \*\*数据库类型\*\*: 使用现代关系型数据库（如PostgreSQL）或NoSQL数据库（如MongoDB）存储用户信息、路线偏好、兴趣点等数据。  
  
- \*\*数据交互协议\*\*: 基于SQL或NoSQL查询语言与数据库进行数据交互。  
  
 - \*\*功能\*\*: 用户信息存储与查询、对注册用户的登录验证、路线偏好数据管理、兴趣点数据同步等  
  
- \*\*界面设计\*\*: API接口的设计需考虑数据的灵活性和可扩展性，能够随着未来增加数据库表结构和字段的变化，同时实现与第三方数据库服务的无缝对接