

Overview

软件工程与计算II

- 刘钦、房春荣
- 大二下
- 232（南京）+154（苏州）人选修
- 平时作业Lab（15%）+ 大作业Homework（35%）+ 期末考试（50%）

课程定位

- 定位
 - 专业核心课
 - 软件工程基础入门课程
- 主要内容
 - 概念基础 *
 - 需求分析 ** 面向对象需求分析方法，需求文档化
 - 体系结构 ** 体系结构风格，接口设计，包的原则
 - 详细设计 *** 面向对象详细设计过程，模块化信息隐藏，类的原则
 - 构造测试 ** 表驱动，黑盒测试，白盒测试
 - 交付演化 *
- 教材
 - 《软件工程与计算（卷二）— 软件开发的技术基础》
 - 《软件工程—经典、现代和前沿》



计划安排

	12		3	4	5	6		
	软件工程基础	需求基础	需求分析方法-I	需求文档化与验证	软件体系结构基础	软件体系结构设计与构建-I		
	软件的概念	需求工程	需求分析概念	为什么文档化需求	软件体系结构的发展	体系结构设计		
	软件工程的	需求的	面向对象分析	用例文档	理解软件体系结构三要素	体系结构构建过程		
	软件工程知识点	需求的层次	结构化分析	软件需求规格说明文档	体系结构风格	包的原则		
二月	如果学习软件工程	需求的分类	使用需求分析方法纸化和明确需求	文档化需求的注意事项		体系结构逻辑设计		
三月	项目和项目管理	需求获取的常见困难		验证需求				
	软对组织与管理			度量需求功能点				
	软件质量保障			需求文档错误分析				
	软件配置管理			软件设计基础				
	管理实践			软件设计概念				
	项目实践			软件设计的分层				
				软件设计过程、方法和模型				
	78		9	10	11			
	软件体系结构设计与构建- II	人机交互	详细设计-I	模块化与信息隐藏	面向对象的模块化			
	体系结构物理设计	易用性	详细设计概念	动机	面向对象中的模块与耦合			
	体系结构构建	人类因素	结构化的详细设计过程	发展	访问耦合			概念基础
	体系结构文档化	计算机因素	面向对象的详细设计过程	概念	继承耦合			需求分析
四月	体系结构验证	交互性		结构化的模块化	内聚			体系结构
		交互设计过程	GRASP原则-创建者	结构化的信息隐藏	耦合和内聚的度量			详细设计
			GRASP原则-控制器		类的原则			构造测试
			为类间协作开发集成测试用例					
			详细设计文档和评审					
			详细设计文档和评审					
	1213		14	15	16			
	面向对象信息隐藏、设计模式	软件构造	代码设计-II	软件测试	软件开发过程模型			
	类的职责	构造概念	决策表	软件测试概念	软件开发各典型阶段			
	类的封装	构造的活动	伪代码	软件测试的分类	软件开发声明周期模型			
五月	OCP	重构	程序流程图	软件测试技术	瀑布			
六月	多态	测试驱动开发	为代码开发单元测试用例	黑盒测试	增量迭代			
	DIP	结对编程	代码复杂度度量	白盒测试	演化模型			
		代码设计-I	代码大全	面向对象测试	原型模型			
	可修改性	设计易读的代码	问题代码	测试的度量				
	设计模式	设计易维护的代码			软件交付			
	策略模式	表驱动			安装			
	抽象工厂模式	设计可靠的代码			部署			
	单件模式	防御式编程			培训			
	迭代器模式				文档支持			

课程变化

- 2（理论） + 2（实践）
- 理论
 - 精简知识点
- 实践
 - 讲解（开发技术、Lab和Homework说明）
 - Lab（环境配置、平台操作、客观题Lab、文档Lab、代码Lab）
 - Homework（大作业：实践项目）
 - PaperReading（经典和现在论文对比）

Eval 、Coder、Developer

- Eval
 - 客观题（每章）
- Coder
 - 体系结构
 - 设计模式
- Developer
 - 大作业

大作业

- 完成方式
 - 分组完成（4人1组）
 - 先自由组队，再老师分配
- 作业内容
 - 代码作业（案例学习、完型填空、自由扩展）
 - 文档作业（项目启动文档、用例文档、需求规格文档、架构设计文档、详细设计文档、人机交互文档、[测试报告](#)）
- 大作业技术栈
 - Web项目（框架不是主要学习的目的，只是借用框架简化开发）
 - 前端：html, css, js, Typescript, Vue3
 - 后端：Java（重点在后端逻辑部分的设计实现），SpringBoot, JPA

文档lab作业和代码Lab作业

周	课程	文档	代码
1	软件工程概念	Lab1：项目启动文档	
2-3	需求	Lab2：用例文档	Lab1：熟悉项目（用户模块）
4-5	体系结构设计	Lab3：需求规格文档	Lab2：商店模块、商品模块
6-7	人机交互	Lab4：体系结构文档	Lab3：订单模块、评价模块
8-9	详细设计	Lab5：人机交互文档	Lab4：优惠券模块
10-11	详细设计	Lab6：详细设计文档	Lab5：额外功能：商品查询、支付API模块、报表导出
12-14	构造	Lab7：测试文档	Lab6：自主拓展
15	测试部署		
16	过程		答辩

大作业检查

- 代码作业自由扩展部分
- 四个检查点
 - 功能
 - 代码
 - 文档
 - 过程

答疑群

苏州



答疑群

南京



软件工程与计算II答疑...

群号: 284955371



扫一扫二维码，加入群聊

为啥要还要学瀑布模型？
为啥还要学些那么多文档？

会跑之前，要先学会走

AI for SE

workflow

- 需求
- 原形
- 代码

需求

以下是根据敏捷软件开发中的User Story形式，给出的打车需求描述：

User Story 1: 乘客叫车

作为 乘客

我想要 发起叫车请求

以便于 快速获取附近的可用司机，前往目的地。

接受标准：

- 乘客能够在应用中输入起始地点和目的地。
- 系统根据乘客的位置，显示附近空闲的司机。
- 乘客可以选择并确认叫车请求。
- 系统返回接单的司机信息，包括司机姓名、车辆信息、预计到达时间等。

User Story 2: 司机接单

作为 司机

我想要 接受乘客的叫车请求

以便于 为乘客提供服务并赚取报酬。

接受标准：

- 司机可以查看乘客的起始地点和目的地。
- 司机能够接单或拒绝叫车请求。
- 系统通知乘客司机已接单，开始前往接客地点。

User Story 3: 司机到达接客地点

作为 司机

我想要 确认到达接客地点并通知乘客

以便于 让乘客知道我已到达并准备接送。

接受标准：

- 司机到达接客地点时，能够在系统中标记“已到达”。
- 系统通知乘客司机已经到达接客地点。
- 乘客能够查看司机的车辆信息和位置，准备上车。

User Story 4: 乘客上车

作为 乘客

我想要 上车并开始行程

以便于 乘车到达目的地。

接受标准：

- 乘客在确认司机到达后，可以选择上车。
- 系统记录开始行程的时间和路线。

User Story 5: 司机行驶

作为 司机

我想要 驾驶并跟随导航至目的地

以便于 准时送达乘客。

接受标准：

- 司机可以查看目的地并获得导航指引。
- 系统提供行程中的实时交通信息和路径建议。

User Story 6: 付款

作为 乘客

我想要 在到达目的地后支付车费

以便于 完成整个打车过程。

接受标准：

- 乘客能够查看费用明细。
- 系统支持多种支付方式（如信用卡、支付宝等）。
- 支付成功后，司机和乘客收到确认信息。

User Story 7: 行程结束

作为 乘客

我想要 在付款完成后结束行程

以便于 确认我已完成支付并结束打车过程。

接受标准：

- 系统在支付完成后，自动结束行程并提供电子发票或支付凭证。
- 乘客能够评价司机和行程。

总结

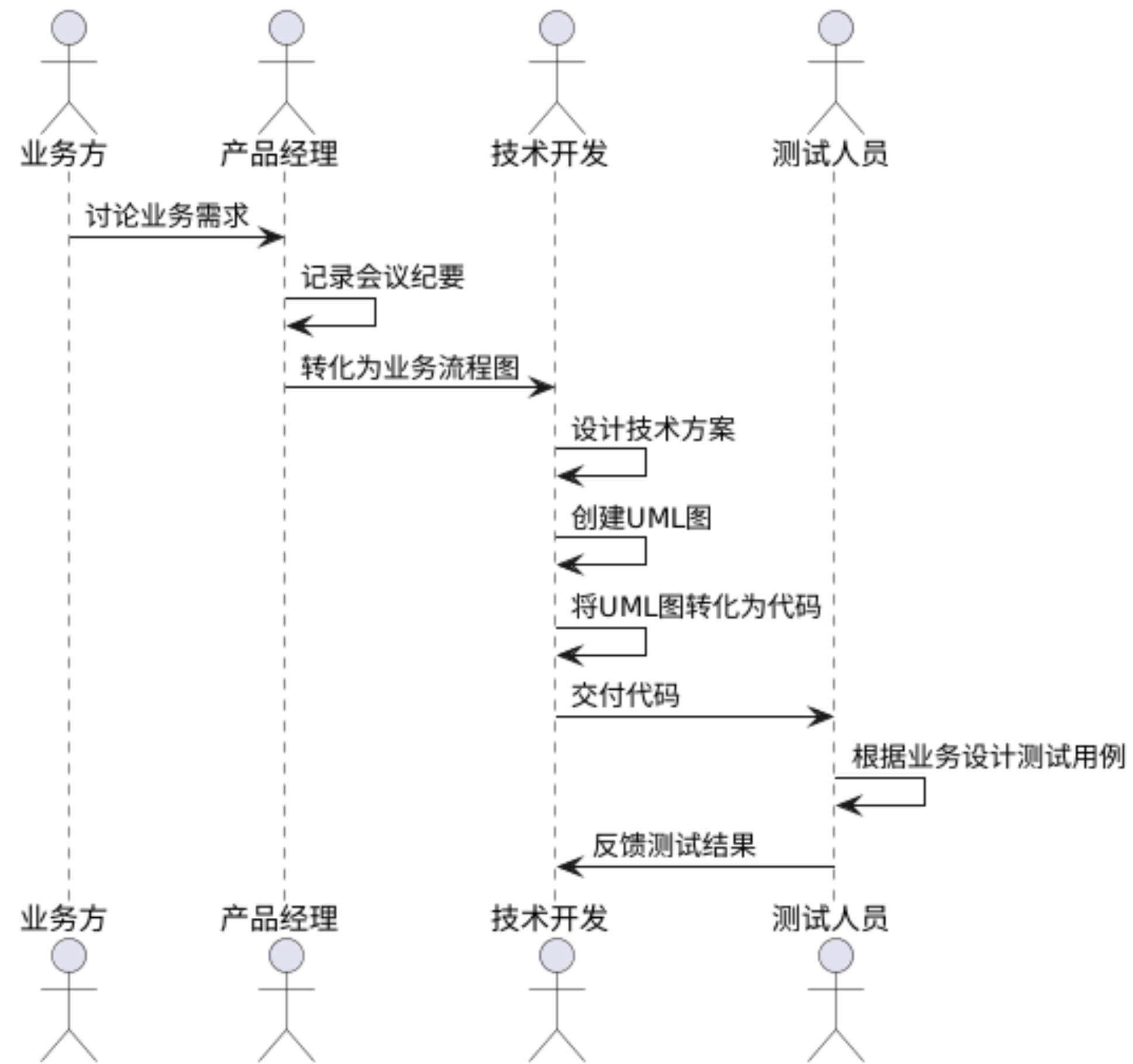
这些User Stories描述了打车过程中每个角色（乘客和司机）的需求，帮助开发团队明确每个功能模块的功能和交互，符合敏捷开发中“逐步迭代、交付功能”的原则。

提示词

- 我：业务方和产品经理开会谈业务需求，产品将会议纪要转成成业务流程图给到技术开发，技术开发根据流程图设计技术方案画出UML图并转化成代码语言，开发完成后测试人员根据业务设计各种测试用例分析来验证代码的正确性。 请将以上的流程转成PlantUML

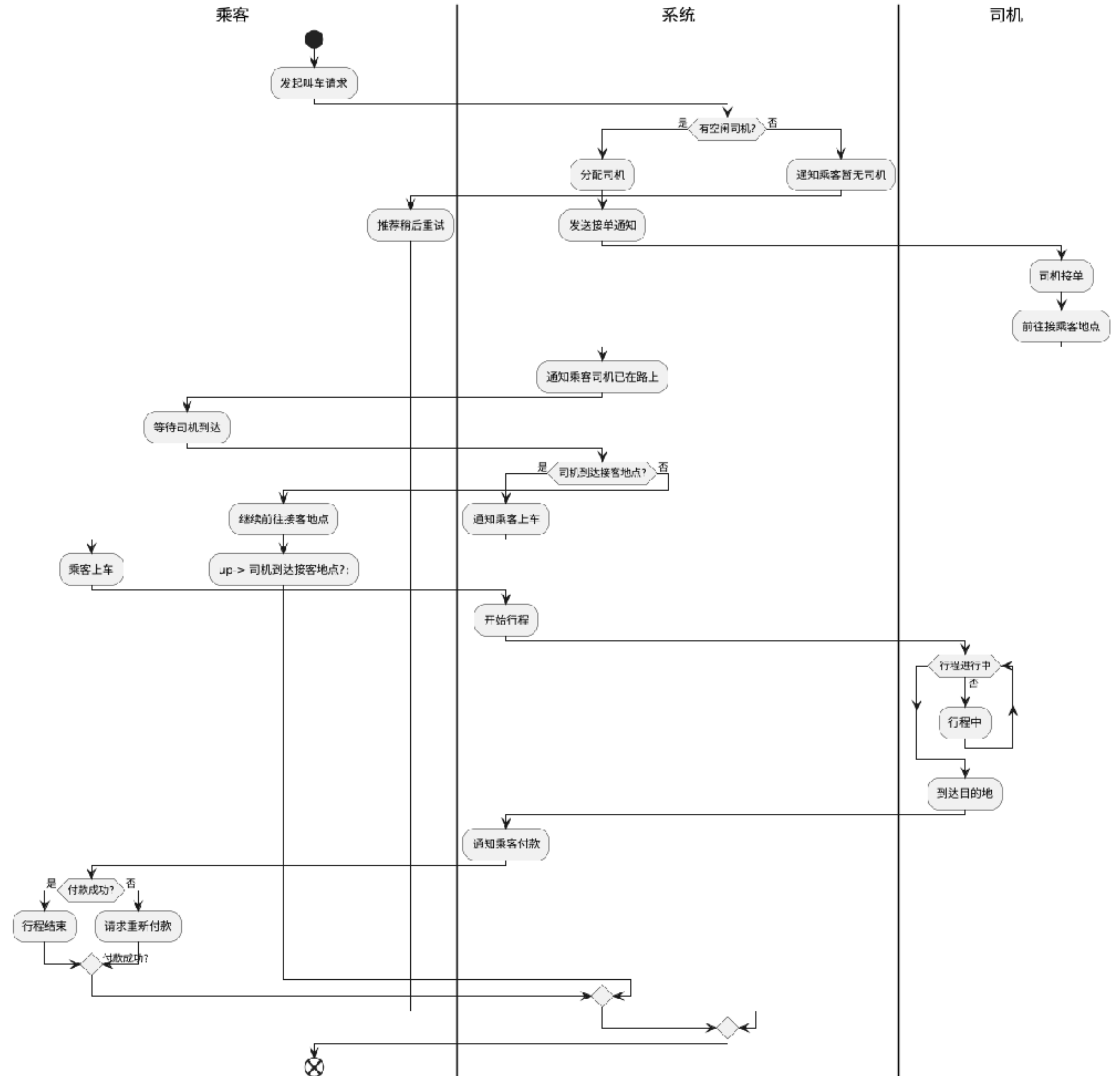
产出

- ChatGPT:
- @startuml
- actor 业务方
- actor 产品经理
- actor 技术开发
- actor 测试人员
- 业务方 -> 产品经理 : 讨论业务需求
- 产品经理 -> 产品经理 : 记录会议纪要
- 产品经理 -> 技术开发 : 转化为业务流程图
- 技术开发 -> 技术开发 : 设计技术方案
- 技术开发 -> 技术开发 : 创建UML图
- 技术开发 -> 技术开发 : 将UML图转化为代码
- 技术开发 -> 测试人员 : 交付代码
- 测试人员 -> 测试人员 : 根据业务设计测试用例
- 测试人员 -> 技术开发 : 反馈测试结果
- @enduml



<https://plantuml.com/zh/>

泳道图



状态图

```
@startuml
[*] --> 叫车中

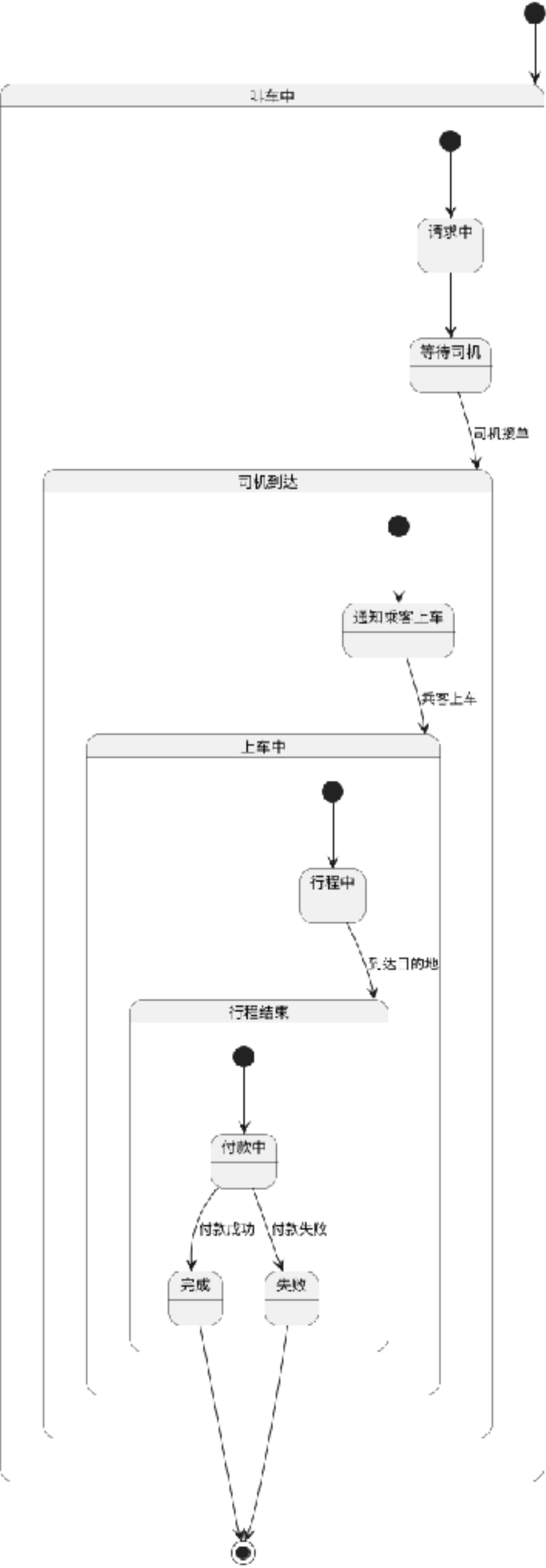
state 叫车中 {
    [*] --> 请求中
    请求中 --> 等待司机
    等待司机 --> 司机到达 : 司机接单
}

state 司机到达 {
    [*] --> 通知乘客上车
    通知乘客上车 --> 上车中 : 乘客上车
}

state 上车中 {
    [*] --> 行程中
    行程中 --> 行程结束 : 到达目的地
}

state 行程结束 {
    [*] --> 付款中
    付款中 --> 完成 : 付款成功
    付款中 --> 失败 : 付款失败
}

完成 --> [*]
失败 --> [*]
@enduml
```





<https://www.pm-ai.cn/>

历史记录

原型图生成PRD(Beta)

阅读《使用说明》，快速上手>

单一功能

整个系统

原型图截图

示例1

示例2

直接拖入/点击上传截图

OrWrmA4WgLiFAQGeem7F.jpeg

取消查看大图

功能名

2/50

打车

需求描述

0/2000

通过登录注册模块，建立账户体系，沉淀用户

重新生成

取消

打车功能的PRD

全部导出

PRD 正文

修改

导出

复制

重新生成

显示脑图

一、概述

本PRD主要描述了打车功能的需求，通过页面信息进行分析和描述。

二、用户使用旅程

1. 用户打开滴滴出行APP，进入主页。

2. 用户选择出发时间为12:00。

3. 用户选择出发地点为北京市。

4. 用户选择目的地和车型。

5. 用户确认订单并等待司机接驾。

三、页面描述

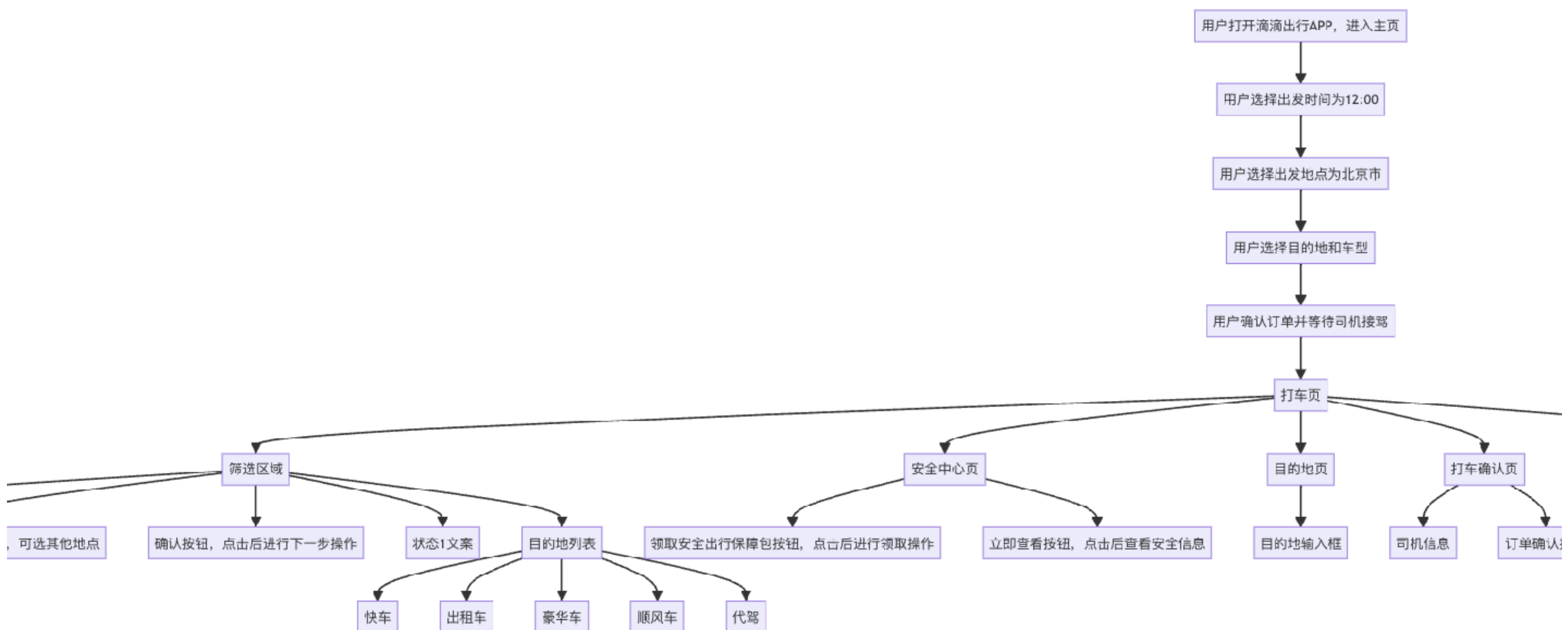
主页-状态1

- 本页面主要功能是展示滴滴出行的图标和状态信息。
- 页面包含以下元素：
 - 滴滴出行icon
 - 状态1文案

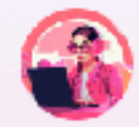
打车页

- 本页面主要功能是提供打车服务。主要包含选区域、车型列表区域、目的地列表区域。
- 选区域主要用途是设置出发时间和出发地点，包含以下元素：

PRD



流程图



ChatPRD

Product ▾

Reviews

Pricing

Resources ▾

Company ▾

Become a **10x product manager**
with an on-demand AI copilot.

ChatPRD is **the #1 AI tool for product managers & their teams.**
Write great product docs—fast, and use AI to become an elite PM.

<https://www.chatprd.ai/>



<https://xinghuo.xfyun.cn/desk?botId=1660>

滴滴出行

**[https://www.woshipm.com/pd/
3525723.html](https://www.woshipm.com/pd/3525723.html)**

Cursor

下载与安装

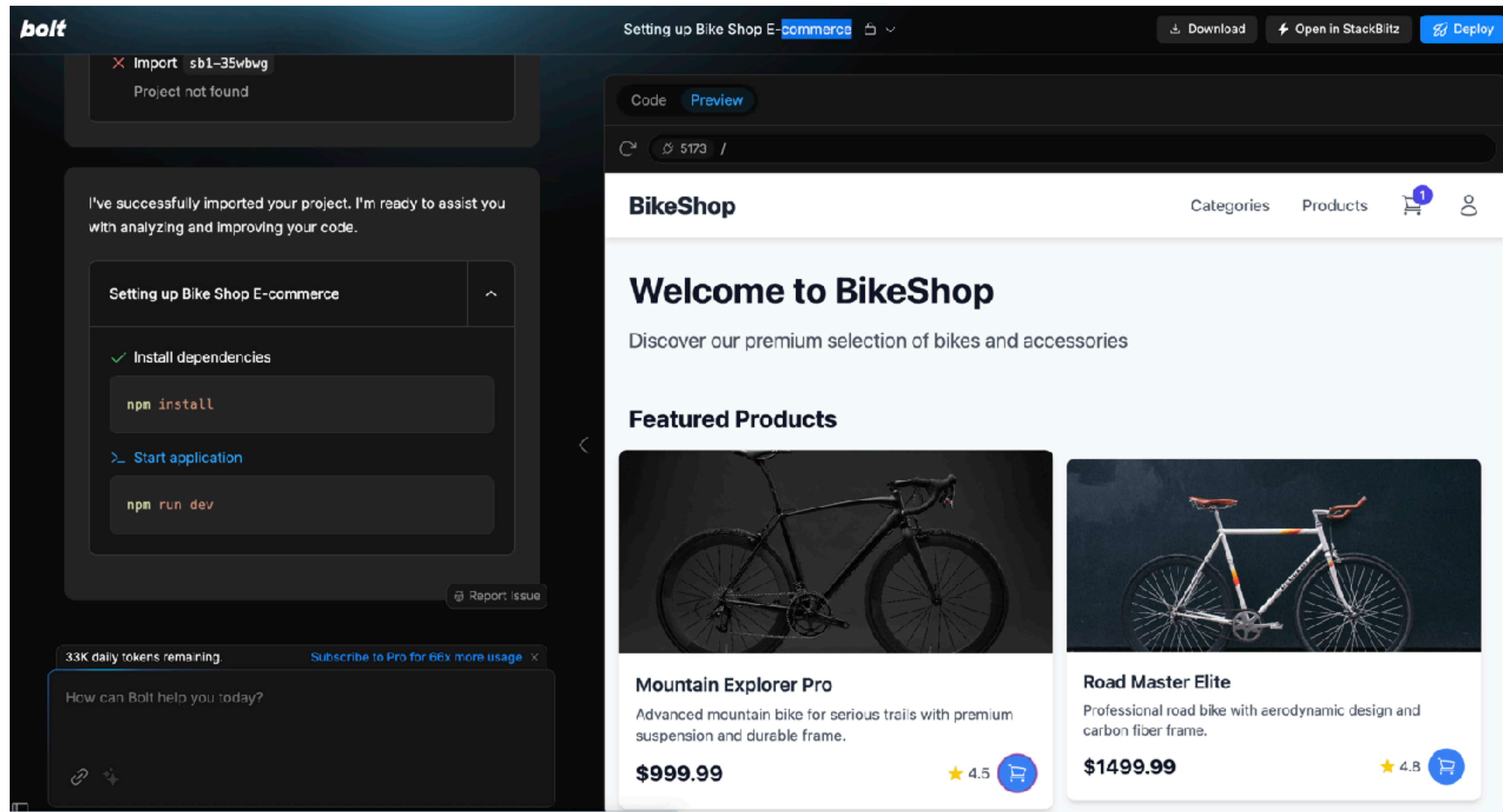
- 下载地址: <https://www.cursor.com/>
- 点击 Download for Free 下载并安装
- 点击 Sign up 使用邮箱注册一个账号

介绍

- 官方文档: <https://docs.cursor.com/>
- “Cursor is a fork of VS Code.” —— Cursor的界面与VS Code完全相同, 相信大家一定都很熟悉
- Cursor Settings > General > Account > VS Code Import 可以一键导入VS Code的配置和插件

简单使用

- 与VS Code相同，打开一个文件夹作为项目的工作空间
- 按Tab可以自动补全代码
- 选中代码后，按Ctrl+K可以给AI提要求修改代码
- 选中代码后，按Ctrl+L可以与AI对话。选中代码后，按Ctrl+Shift+L可以将选中代码片段加入到AI对话上下文中
- 在对话时，按下Ctrl+Enter扫描整个代码项目，然后可以根据该基础向AI提出问题



<https://bolt.new/>

Vercel

- 官网地址: <https://vercel.com/>
- 官方文档: <https://vercel.com/docs>
- 点击 Start Deploying, 从Github/Gitlab/其他Git仓库导入项目 (登录相应的账号并授权)
- 选择需要导入的项目, 点击Import
- 在配置页面输入项目名称; 选择项目的预设框架, 如Vue.js; 填写环境变量 (如需要); 最后点击Deploy进行部署
- 进入项目的Dashboard即可看到项目状态
- 项目 > Settings > Domains可以为项目设置域名, 具体可参考: <https://vercel.com/docs/getting-started-with-vercel/use-existing>