实验九 UML，逻辑， 软件体系结构设计（一）

实验目的：

1. 深入理解UML

2. 了解计算机学科中的逻辑

3. 学习对比软件体系结构设计GB和IEEE最新SAD (Software Architecture Document)的标准

4. 研究经典软件体系结构案例

3. 完成自己项目的SRS

实验内容：

1. 阅读“The Unified Modeling Language Reference Manual”，进一步学习UML知识，理解如何应用UML对系统进行建模

根据需求分析需要使用的模块，分别对模块建模，使用UML语言连接并表示对应关系。在交流讨论过程中，我们提出不同的想法并交流。

UML是一种非专利的第三代建模和规约语言，是面向对象设计的建模工具，独立于任何具体程序设计语言。UML是在消化、吸收、提炼至今存在的所有软件建模语言的基础上提出的，集百家之所长，是软件建模语言的集大成者。UML不仅可以用于软件建模，还可以用于其他领域的建模工作。UML的主要特点包括：面向对象、可视化、表示能力强、独立于过程和独立于程序设计语言，且易于掌握使用。

UML包含多个图形化工具，这些工具被称为UML图，包括但不限于：

用例图（Use Case Diagram）：描述系统与外部参与者之间的交互。

类图（Class Diagram）：显示对象、类、接口、继承和关联等。

时序图（Sequence Diagram）：描述对象之间的交互顺序。

活动图（Activity Diagram）：描述业务流程或工作流。

部署图（Deployment Diagram）：描述系统中各个组件如何被部署到硬件设备上。

软件开发：UML类图是一种帮助软件开发人员设计和构建软件系统的工具，可以描述系统中的对象、类、接口、关联等元素，并用它们来实现系统的功能。

系统分析：UML图可用于描述系统中的不同组件和它们之间的关系，支持多种分析技术，如数据建模、用例建模等。

数据库设计：UML类图可用于描述数据库中的实体、属性和他们之间的关系，帮助开发人员更好地理解数据库中的结构和关系。

代码生成：开发人员可以使用UML类图来描述系统中的各种元素，从而生成具有标准形式库的代码。

测试：UML图还可用于软件测试领域，描述被测试的软件系统的各个组件和它们之间的关系，从而确定测试的范围和测试用例的选择。

UML的主要作用包括：为软件系统建立可视化模型，使系统结构直观、易于理解，有利于系统开发人员和系统用户的交流，还有利于系统维护。为软件系统建立构件，UML的模型可以直接对应到各种各样的编程语言。为软件系统建立文档，不同的UML模型图可以作为项目不同阶段的软件开发文档。

2. 浏览“LOGIC IN COMPUTER SCIENCE--Modelling and Reasoning about Systems”，了解常用逻辑及其在计算机学科中的应用

通过浏览“LOGIC IN COMPUTER SCIENCE--Modelling and Reasoning about Systems”，深入了解常用逻辑及其在计算机学科中的应用，掌握其在项目规划过程中的使用，思考如何将其成功运用到我们小组的项目中。

3. 分工协作，参考国标“13 - 软件(结构)设计说明(SDD)”等资料，对比参考SAD最新标准IEEE-42010.pdf，针对自己的项目设计SAD初稿。

通过参考国标“13 - 软件(结构)设计说明(SDD)”等资料，对比参考SAD最新标准IEEE-42010.pdf，结合我们小组的网上交易系统，我们编写了SAD初稿。

4. 分工协作，学习、检索研究经典软件体系结构案例。

On-the-Criteria-To-Be-Used-in-Decomposing-Systems-into-Modules.pdf

http://www.cs.cmu.edu/~ModProb/index.html

我们分组协作，鲍俊杰、陈一苇负责On-the-Criteria-To-Be-Used-in-Decomposing-Systems-into-Modules.pdf的学习和检索，夏楠翔、霍泉如、石子超负责<http://www.cs.cmu.edu/~ModProb/index.html>的学习检索，然后我们再交换工作，保证充分学习相关案例。

5. 完成软件需求规格说明SRS

**下周五（含）前将软件需求规格说明提交给相应的助教**

项目跟踪，建立能反映项目及小组每个人工作的进度、里程碑、工作量的跟踪图或表，将其保存到每个小组选定的协作开发平台上，每周更新。