软件工程课大作业

——对《成绩管理系统》的软件工程实践

零、今年大作业的总体要求

今年仍以小组（每组 4 人，自行组队）为单位提交一份报告。写明每人的分工和贡献，并选一位组长（得分要看每人的具体贡献，组长相当于 Scrum Master）。报告文档命名规则：“组长名-2024 软工大作业.docx”。

去年还规定如果增加了程序功能有加分。但后来发现针对代码加分不容易形成统一标准，不同功能的工作量差异很大，且评阅代码也很费劲。因此今年取消对程序扩展的考量。但发现和修改 Bug 还是有加分。测试部分可能需要一些代码。

大作业文档部分合在一个文件内提交，不要分成几个文档（最好是 word，我好统计字数）。提交截止时间 2025 年 1 月 5 日。提交链接还是坚果云(点我)。

一、大作业目标

之前的小作业让同学们掌握了软工一些基本概念，包括对软件开发各阶段和相应文档的认识，并训练了一些绘图技巧。大作业作为本门课程的一个独立组成部分，考察大家综合应用软件工程各阶段知识与能力，对一个具体软件项目进行需求分析、设计和测试的过程。

本作业采用“反向训练”的方式，模拟一个 Java 软件的开发过程，即先给出源代码，然后在此基础上进行分析、梳理和描述，形成开发文档。该项目为一个简单的单机版成绩管理系统，包括学生、教师、管理员三类角色。具体内容见《成绩管理系统说明文档》。

请同学们针对该系统，按照软件工程规范，提供较为完备的过程文档。文档的结构和格式可参考《软件项目报告模板（2024 版）》（不必完全按照该模板），内容上可参考《软件项目报告样例文档-成电微记》和《健康码管理系统案例》。

二、依托的资料及代码

- 代码部分：成绩管理系统源代码采用 Java 语言编写，并附有《成绩管理系

统说明文档》。该代码和附带的说明文档主要由许毅老师提供（我略作了修改）。我在 win10 下测试过，源代码可以运行，基本能用（软件有 Bug）。但要注意该源码中把有些目录或路径写死了，如果拷到不同的目录下，有可能出现无法识别的情况。之前有同学报告，在登录环节输入说明文档中提供的账户密码，无法登录，有可能是该原因。

- 《软件项目报告模板》最早是由邱元杰老师提供，该文档是一个通用的报告模板，结构比较完整（还包含硬件部分的描述）。但从前几年的大作业实践来看，同学们普遍反映该模板比较死板，采用了较多的结构化分析和设计思想，对面向对象开发思想的体现不是很足。另外该模板有些提法和课上讲的对不上，还误导了一些同学。因此我对模板做了一些精简和修改。原模板附录 A 之前本带有“总论部分”，我把“总论部分”删掉了，直接从附录 A 开始。我还修改和删除了一些过时的提法，不恰当的条目等，并在容易理解有误的地方加了一些注释。总之请大家用我发的新模板，不要使用老模板或别的班上发的模板。

  同学们也可根据自己对“成绩管理系统”的认识，自行裁剪和调整该模板的内容（有什么就写什么，没用到的模板中的章节可以删掉；对模板中的说明性的文字要删掉；还可以调整条目的顺序，修改条目的标题，或增加某些条目）。
- 《软件项目报告样例文档-成电微记》由课题组提供，供大家撰写报告时参考。不过“成电微记”是一个手机端 app，而“成绩管理系统”是一个 PC 端的软件，两者的类型和风格不同，大家也不要硬套“成电微记”文档的结构和写法。
- 《健康码管理系统案例》，该文档来源于钱伟中老师去西安开教学研讨会时带回来的课件。它的优点是以一个具备实用性的“健康码管理系统”为贯穿式案例。该案例从需求获取与分析开始，一直到软件部署与维护，对该系统的全生命周期都有涉及（本次大作业只到测试环节，不涉及部署和维护）。虽然中间省略了一些实现细节，但各环节基本要素都体现了，从概念上和操作层面都可以给同学们一些直观的感受或理解。同学们可以学习和模仿该案例，如何一步步展开，每个环节从哪些方面入手、有哪些重点，如何用

文字和图形来表现该项目的全过程等。不过该案例也不完美，比如需求分析阶段没有用分析型类图和顺序图来建模。

- 辅助文档：除此之外，还有两个指导格式规范的文档（已上传到 QQ 群文件中的“大作业”目录中）

  1) 《Word 排版艺术》：直到现在，我发现很多同学对 Word 中的常用功能都不熟悉，对格式的概念以及 Word 对格式的管理仍然没有理解到位，仅把 Word 当成普通的字处理软件来用，而没有当成排版软件来看待。这本书虽然对应的 Word 版本较老，但把 Word 的精髓都讲到位了。

  2) 《带你读“格式规范”（导出自 Onenote 笔记）》：这篇文章最初来自我给自己指导的大四毕业设计的同学写的内部指导的帖子。但我后来发现，无论是大二同学大作业的写作，还是大四同学的毕设写作，都存在一些共性问题，一直没有得到系统性解决和指导。虽然你们的大作业的具体模板结构和毕设不同，但帖子中指出的典型问题都是相通的。希望同学们结合上面的《Word 排版艺术》一起看会效果较好。

这次大作业没有安排代码开发，由于是纯文档写作，因此我对文档的写作质量会要求更高。希望借助这次机会，让大家系统性地训练文档写作，以后无论是大三企业实习报告，还是大四毕设论文撰写，都打好基础。

三、过程要求和输出内容

请同学们分析成绩管理系统的源代码及《成绩管理系统说明文档》，撰写系统的需求分析、概要设计、详细设计、测试四大阶段的文档。

基本要求如下：

阶段	文档	对图的要求	其他要求
需求分析	软件需求分析报告（或需求规格说明书）	必画图：用例图、活动图、分析类图、分析顺序图 可选图或表：数据字典、状态图、数据流图等。	用例部分除了画图、文字，还要辅之以表格形式的用例规约来说明。
概要设计	软件概要设计报告（或概要设计规格	必画图：功能模块图（可以选用层次图、IPO、HIPO 图等）。 可选图或表：体系结构图、部署	对数据库或数据层的设计不做强制要求。

8.2类图

8.3顺序图

	说明书)	图、包图、ER 图等。	
详细设计	软件详细设计报告(或详细设计规格说明书)	只需对代码中的 controller 包和 model 包中的类进行详细设计，可以不包括 view 包（即界面模块)。 必画图：设计类图、设计顺序图。 可选图或表：状态图、流程图、活动图、构件图、判定表/判定树等。	a 对用户界面的设计不做强制要求。当然有更好。 b 详细设计中不必对所有方法的算法进行描述，只需选重要的、过程复杂的、或对系统性能影响大的描述即可，既有点也有面！
测试	软件测试报告(或测试规格说明书)	要求体现单元测试、集成测试、验收测试三个环节： 1 单元测试只需从 controller 包中选一个类进行测试即可，采用白盒测试中的基本路径测试法设计测试用例。 2 集成测试只要求对 model 包里的几个类进行集成测试。假设 controller 和 view 包里所有类已完成集成，且 model 包里的类已做了单元测试。 3 验收测试不用分α、β等多阶段，只需测试“教师”这个角色（不是 Teacher 类）涉及的用例。采用黑盒测试中的等价类划分和边界值分析法相结合，设计测试用例。	a 对于非功能性测试(健壮性、性能、压力等) 不做要求。 b 测试部分不仅要有测试用例的设计，还要有测试结果的记录和分析，并附以系统运行截图。 c 对于自动测试框架（如 JUnit）的使用不做要求，当然有更好。 d 修改 Bug 可少量加分(仅 1~2 分)。修改 Bug 不是本次的考察重点，重在发现。另外，若有修改，还需进行回归测试。 e 无需增加软件功能，扩充功能不加分。

上述图中若属于 UML 标准内的图，要求用 starUML 软件来画（防止用 GPT 直接生成图，但可以用 GPT 来参考)；UML 标准外的图不限制绘图软件，但是不能用手绘。

以上各环节除了图、表以外，文字说明也是必须的。不能只靠图展示一切，要图文配合。

考虑到大家还没学过数据库课程，因此概要设计部分的数据库设计(包括 ER 图、各种表等）不做要求。且本程序的数据量较小，也不需要用到数据库，用文件存储即可。

四、评分标准

1. 有效字数：≥25000 字，但页数不能超过 100 页。不贴 Java 源代码（除了测试阶段自己写的测试脚本）。模板中的注释性文字要删掉，不能算字数。其中详细设计部分不能低于 7000 字。
2. 需求分析 30%，概要设计 20%，详细设计 30%，测试 20%。这只是报告总分中的各部分占比，不是负责各部分的同学的得分比例。上一届就有某位负责概要设计的同学在全组得分最高。
3. 文档格式的整洁规范，画图的质量也是评分重要参考。组长统稿的时候要对全篇的格式进行规范。
4. 我会先打报告的总体分。在总体分的基础上，每人按贡献得分 = min(总体分+个人贡献波动分,100)。总体分范围为 60~100，个人贡献波动分范围为±0~10。因此需要组员齐心协力，保障报告的整体质量。不要只关心自己负责的那部分的水平，若报告整体质量差，即使局部环节优秀也会被拖累。

五、补充说明

1. 关于分工

大家的分工除了按照过程环节（需求，概要设计，详细设计，测试）来划分以外，也可以纵向划分。比如：

编程能力强的，可以负责读程序，剖析代码，编写测试脚本，查找 Bug 等工作。

写作能力强的，可以负责文字部分的写作、加工润色、格式调整，由其他组员为其提供素材。写作的质量表现对总分的影响有时超过内容！

画图能力强的负责绘制（别小看这部分，图的数量不少，工作量未必比文字部分小）。

搜索能力强的，可以负责资料检索和技术学习的工作。

网上摘的帖子：有什么事是你过了很多年突然明白的？ - 卜卜麻麻的回答

2. 关于 ChatGPT 的使用（或国内外类似的 LLM 应用）

ChatGPT 横空出世，给文档写作和作业评判都带来一定的挑战。但目前的 GPT 最擅长写那些缺乏问题针对性，不面向具体应用场景的“口水话”。

我并不反对在本次大作业的撰写中使用 ChatGPT 进行辅助，毕竟是一种生产力工具。但实质性观点要自己去写。可以使用的地方包括：

1) 知识和技术查证。但 ChatGPT 有时对一些常识也会一本正经的乱说，注意用别的搜索引擎查证。
2) 分析和解释代码，这个倒是大模型的强项。目前代码已经提供给大家了，可以用 ChatGPT 辅助分析。但要强调的是，ChatGPT 擅长战术性分析，但涉及到“战略”层面的，比如需求分析和概要设计，恐怕还是需要对整个软件通盘考虑后再写。
3) 文字润色。要先有自己写的内容，再用 ChatGPT 进行润色。

不能使用的地方有（以下几点过于明显会从总分中扣 5~20 分）：

1) 背景介绍。例如，“成绩管理系统在高校使用情况的介绍”，“Java 后端主流技术介绍及趋势”，“Spring 框架介绍”等假大空的内容。
2) 知识、名词、技术介绍或术语解释。例如，“什么是软件体系结构”，“什么是白盒/黑盒测试”，“怎么画类图”等言之无物的内容。
3) 本软件没有的功能和设计。比如有同学直接给 ChatGPT 一个 prompt：“Java 开发的带界面的成绩管理系统的需求规格说明书”。ChatGPT 生成的内容必然有一些本软件没有的东西，也要扣分。记住：给的 prompt 越具体越好，越大越泛则表现越差，不要偷懒。有时甚至喂给 prompt 的字数，要超过最终从 ChatGPT 生成中摘选的字数！
4) 详细设计部分描述重要函数的算法时，不能用伪代码来表现，因为很容易用 ChatGPT 生成。当然大家可以用 ChatGPT 生成伪代码或解释来帮助理解是可以的。
5) 同理，不能直接把大模型生成的图贴进文档。目前大模型渐渐具有文生图的能力，你可以使用大模型生成图做参考，然后自己照着再用 starUML 画并做一些修改，这样是可以的。

参考这个回答（作者“魔法师”可能是个在海外大学任教的华人）：学生使用 ChatGPT 写作业，学校应该坚决禁止这种行为吗？

总之，不能用 ChatGPT 凑字数！我要求大家不要做背景介绍、也不要做技术性解释，一上来就紧扣本软件，直接写干货！ChatGPT 生成的内容，一定要自己读过，要进行选择和组合；最好用自己的话重新说一遍。

我会发几份全年的大作业中滥用 GPT 的反面教材给大家看看，大家引以为戒！

要知道写这些文档的目的，不是让大家去讲课，也不是去推销售卖软件，而是模拟在软件开发组织里面的过程，写出的文档是要给设计和开发人员看的。最终追求的效果是：对于没有见过该软件的开发者，也可以依据这些文档，实际开发出和这个一模一样的软件出来。