

CS108 Group Project

Learning Objective

In this project, you will learn how to interact with a team of software developers to propose, design, implement, test, and demonstrate a software application.

Problem Statement

Typical application development is conducted as a team effort. Work roles include requirements analysis, detailed design, programming, testing, and deployment (presentation in our case).

Your goal is to develop a **prototype** for a real-worlds business application. A prototype is a small-scale implementation (to limit complexity and time required) that will serve as a proof-of-concept for a larger software project. The goal of developing a prototype is deeper learning about the problem, and developing insights to assist in a further development, should that decision be undertaken.

As a team, you will select the **subject** of your application. You have a lot of latitude in what you choose to do, and I will assist you as best I can.

Your application should include the following elements of application programming we've used during the semester:

- data abstraction, modeling business problems using objects.
- a useful process/algorithm (e.g. search/sort, calculation, etc.)
- a multiple-window graphical user interface for the application.

In addition, you will use the Java API and your knowledge of programming with objects to teach yourselves and **use some Java library components that we have not covered in class** (e.g. using files, advanced data structures, database, networking, etc.)

Once you select the subject, you will act out a software development life-cycle project, which will include several deliverables outlined on the next page. We will discuss this process, and each deliverable, in class as we go.

Development Team Rules

- Teams of 4-5 students will be selected at random during classroom hours.
- Each team must select a **team lead** who will coordinate meetings, submit all deliverables, and schedule any required group meetings with the instructor.
- Collaboration and teamwork are required *within* the teams. Collaboration across teams is *prohibited*.
- Each team member will evaluate each other team member on a form to be provided around the last week of class. Responses will be held in strict confidence, and will be taken into consideration for grading purposes.

Details, details, details

| Deliverable | What's expected/details | Due Date |
|---------------------------|--|--------------------------------|
| Project proposal | The project proposal will answer the question: what problem/application will you study? Explain why this is an interesting problem/application. (1/2 page at most). | 5 April 2005 |
| Requirement analysis | The requirements analysis will answer the question: what should the software application do? Here, you will be specific about user interactions, calculations/algorithms, and process/work flow. A UML use-case diagram is strongly recommended to support your analysis. (2 pages at most) | 12 April 2005 |
| Design and prototype plan | The design will answer the question: How should the program work? Here, you should go into detail about the objects involved, and provide UML class diagram(s) to show the interactions among the objects. Think about the behaviors/responsibilities required of each object, and show the method names in the UML class diagram. Provide a brief narrative text to accompany these diagrams (about 1-2 pages of narrative should suffice). The prototype plan will be a brief statement on which <i>subset of functionality will be built</i> for the prototype. (keep it brief!) | 19 April 2005 |
| Project presentation | You will have the opportunity to present your project in class on Tuesday April 26 th or Thursday April 28 th , 2005. Your presentation should be a fairly relaxed talk about your prototype application. The presentation must include a brief summary of the requirements, the design, and a demonstration of how the program works. You may use a <i>small</i> number of PowerPoint slides or handouts, but keep it simple. | 26 April 2005 28 April 2005 |
| Complete project | You will submit a complete project portfolio including all previous deliverables (revised as appropriate), all source code, and an executive summary. The executive summary will introduce the project deliverables, discuss lessons learned from this prototype exercise, as well as a go/no-go recommendation – should this application proceed to a full-scale development. (2-3 pages maximum) | 28 April 2005 |
| Group evaluations | | 3 May 2005 |

Grading Criteria

- Good design, separation of interface from implementation, good user interface.
- Attention to detail in code, style, comments in the code, etc.
- Well-written and revised project documents.

Questions?

Bring them to class, or email me, azs@bu.edu.

NOTE: I will take all questions and meet with groups up until Monday, April 26th at 6pm. After that time, you're on your own. Plan your time wisely and start early.