CS619 Project 2

Instructor: Karen JIN

October 12, 2012

Project Overview

The purpose of the project is to have the students apply the principles they learn in class on a software development project that is large enough to exercise the project management techniques being discussed while small enough to be feasible within a one term project.

This is a group project in which you will learn about team dynamics, work distribution, etc. The project is designed so that it is of a manageable coding size for groups of 4-5 members. Team membership remains the same as in project 1. If you would like to change your current grouping, you need to request by email before **Friday, September 28, 2012**. However, the reassignment is not guaranteed due to the SVN client settings.

The project must be executed following a simplified UP process with three 3-week iterations. At end of each iteration, a partially but working system must be submitted. Project deliverables (described below) should also be produced using proper standards and formats during each iteration.

Project Description

The project description is posted below and all teams will work on this project. Your instructor, Karen JIN is playing the role of the client.

During the rest of this semester you will work together in groups to progress through the software development lifecycle to develop a working application using OOAD. The instructor, Karen JIN, will act as the client for the application you will develop, and will be able to clarify the following basic requirements for the application:

Develop a textbook exchange that enables students to sell textbooks to other students. Students will browse the available textbooks and have access to the following capabilities:

- Register for the system by providing their name, Wildcats ID and email address and a password
- Log in to the system by entering their Wildcats ID and password
- Search for textbooks (by title or ISBN number or author), and scroll through the results of their search. The results of their search should include the price and condition of the book.

Note that more than one of the same textbook may be returned by the search due to multiple students listing the same book. In this case the user should be presented with a list of results showing the different prices and conditions.

- Purchase textbooks (by sending an email to the textbook buyer and owner indicating an intent to purchase)
- Sell textbooks. The user will need to enter the title and ISBN number, price and condition of the book
- Remove user's posted textbooks from the system (once the student has completed a sale, or if the user decides not to sell the textbook)
- Display additional information about textbook search results, provided directly from Amazon, including an image of the textbook cover, price, authors, and the book description

You will note that the above requirements are quite brief. This is a common scenario in the real world of software development. Most clients do not think like software engineers, they do not envision all of the scenarios that can occur in a multi-user environment, or with an application that functions over the Internet. For each issue you encounter, you can decide whether to consult with the client (instructor) or use your best judgment as to how the system should behave. For example, you should decide on whether you are building a windows-based or a web application, and whether to use database for managing the books or just use plain files.

An example of a web-based system which implements the above features can be found here: http://ptx.tigerapps.org/

To retrieve information about textbooks from Amazon you will need to enroll in the Amazon affiliate program, and download and implement the Amazon Product Advertising API which allows you to submit queries to Amazon for information about their products: You can set up a group account for this purpose.

https://affiliate-program.amazon.com/gp/advertising/api/detail/main.html

This project will require more planning, design, and development time than projects you have been assigned in other courses. For this reason you will need to begin working on the project as soon as possible. It is therefore suggested that once the groups have been formed you meet with your group members as soon as possible to assign roles and responsibilities and begin the software engineering development process.

Also it is important to know that the clients in real world mostly like will make changes to the initial requirements. In our project, it is very likely that I (as the client) will change my above stated requirements and request additional features in your program. You should take this into consideration when designing your system.

Project Deliverables

Project deliverables **for each iteration** are listed below in priority order.

1. The working software, with documentation to be deployed into production.
2. Source code in Eclipse IDE. Must follow common coding guidelines.
3. Regression Test Suite. A collection of test cases, and the code to run them in the appropriate order. Should include a wide range of tests, including acceptance test, unit test, system tests.
4. System Documentation and release note summarizing the current release of system.
5. Requirement model, Describes the requirements which your system should fulfill. This consists of use case model (including fully dressed use cases, use case diagrams, System Sequence Diagrams) and domain model (CRC cards)
6. Design Model Describes the design of your system. This includes
   a. the object model, which includes the static structure and the dynamic aspects of the schema, shown in UML class diagrams and UML Sequence and/or Communication diagrams.
   b. a User Interface Model is also need to describes the user interface of your system.
   c. a Physical Data Model describes the physical schema of a data store, such as a relational database, XML files, or plain files.

The above items 4 - 6 should be incorporated in the Project Report that you submit at the end of each iteration.

Each completed work product must be maintained on your group's content management site and the project report (for each iteration) and weekly report must be posted on your group website. Each document must also have a revisions section in which changes to the document are tracked and the author of the changes identified.

Final Project Report

The Final Project Report should be finished in three iterations with increments, improvement and modification from the report of each iterative. The report should include all the following products covering the whole software development process.

1. Cover page (1 page) document name, project title, team member, department and university, date, etc.
2. Table of Content (2-3 pages) (including list of figures, list of tables.)
3. Introduction (3-4 pages)
   (a) Problem definition
   (b) Project Overview and release note summarizing the current release of system.
4. Software requirements specification document(about 8-10 pages). This document should include:
   (a) Consistent and unambiguous functional requirements. Give the complete use

case diagram with all the use cases and actors. Define in detail five use cases (the most complex ones) in fully dressed format.
(b) Software qualities (correctness, efficiency, robustness and user friendliness, etc).
(c) System constraints (programming languages, operating system, hardware, databases, Web services, software tools, etc).

5. Design specification document (about 8-10 pages), including:
   (a) Software architecture description with diagrams.
   (b) Five sequence diagrams (for the five defined use cases).
   (c) Complete class diagram with all the classes and their relationships.
   (d) Component diagram including the algorithm(s) behind your implementation.
6. Technical documentation, (about 6-8 pages) including:
   (a) Programming languages.
   (b) Reused algorithms and programs.
   (c) Tools and environments.
   (d) If using database system, list all your tables.
   (e) If using files, indicate the formats of the text files used in your code.
7. User documentation including several screen shots of the GUI (2-3 pages).
8. Software test plan including test approach, feature tested, tools and environment: (about 6-8 pages)
   (a) Correctness testing with some data tests.
   (b) Performance testing with some benchmarks.
   (c) User interface testing with some screen shots.
   (d) Robustness testing with some incorrect data.
9. A document that comments in detail on the UML tool used for your project (1 page).
10. Reference (citations) in APA citation format.

The report must be formatted with 1.5 line spacing and font 10 or font 11 text. All figures and tables must be sized appropriately and include title and description.

You must not just show the UML diagrams, but provide narrative discussion for all figures, tables, and diagrams. This is because diagrams, particularly technical diagrams, are rarely if ever self-explanatory. You should also document the alternative solutions that you considered as well as the arguments for the final choice. Diagrams only represent your final solution, but do not explain why you decided on this solutions and what alternatives were considered. Hence, all diagrams must be accompanied with explanation and discussion of alternatives and tradeoffs. Explanations should be written in prose and key arguments highlighted in bullet points.

The report should be prepared electronically. It is mandatory to use a software tool for producing UML diagrams for this report. Hand-drawn UML diagrams are not acceptable. Any tool that supports the UML symbols is acceptable. Hand-drawn figures are acceptable for illustrating the user interface requirements. It is your responsibility to ensure readability of your diagrams.

Here are some online resources to help with your report:
http://www2.dis.ulpgc.es/~jsanchez/MDS/EffectiveUseCases.pdf
http://www.cs.bham.ac.uk/~pxc/proj/ProjectReport.pdf

Effort Breakdown

Your report for each iteration must include a table showing effort breakdown. This activities help team members distribute the project responsibilities equitably and the instructor's grading process.

This report should contain information about your requirements analysis, user interface specification, and domain modeling, and system implementation. The effort breakdown must reflect all activities that contributed to the particular component of the report and the implementation of requirements during each iteration.

The sample effort-breakdown table format for this report is as follows. Assume that the team has four members and their individual responsibilities are shown in this responsibility matrix (with tentative weight for each responsibilities):

	Team Member Name			
	Member 1	Member 2	Member 3	Member 4
Project Management[1] (5%)	70%		30%	
Report: Requirements Specification (15%)	20%	20%	30%	30%
Report: Object Model (20%)	30%	25%	20%	25%
Report: User Interface Model (10%)				
Report: Physical Data Model (10%)			80%	20%
Report Preparation (5%)	50%	50%		
Implementation: Coding (15%)	25%	25%	25%	25%
Implementation: Testing (20%)	20%	30%	30%	20%

The bold activities in the above form must be evenly distributed (mostly) among all team members.

[1] The effort of project management includes: maintaining public websites for the project,

Project Evaluation

Iteration 1 deliverables due on **Oct 18** (10%)
Iteration 2 deliverables due on **Nov 8** (10%)
Final iteration deliverables due on **Nov 29** with project presentation on **December 4** and **6**. (25%)

It is mandatory for all students to attend their team presentation. Each presentation will lasting about 15 min.

The project deliverables at each iteration will be marked on the following basis:

- Completeness
- Consistency
- Level of specificity
- Clarity
- Organization
- Format, spelling, grammar
- Use of diagrams
- Software craft: All programs are expected to be written in a professional manner. The grading will consider at least the following:
    - Code quality and organization
    - Internal documentation

Note that the project deliverable for each team is graded *independently* of each team member based on the overall project mark, effort-breakdown chart, the weight of each responsibility and a (tentative) scale factor of 25%.

Important: Each team member must participant in the following activities and the total amount of work must be evenly distributed.

- Report: Requirements Specification
- Report: Object Model
- Implementation: Coding
- Implementation: Testing

Weekly Report

You need to submit a weekly report on each Thursday starting September 27. The reports must be posted on group's content management site as well as your team website.

Here is a sample template for the weekly report:

Team #:
Milestones this week : Status:
Deliverables this week : Status:
Meetings this week:
Planned Meetings:
Upcoming Milestones:
Upcoming Deliverables:
Resource usage (above, below, expected):
Problems encountered:
Tasks for this week for each member:
Accomplishments this week:
Tasks for next week: