CSCI-4448/5448: Project Description
Spring 2016

Objectives

- CSCI-4448 Work in a group of size 4-5 people
- CSCI-5448 Work in a group of size 2-3 people
- Develop an object-oriented software product

Semester Project Overview

All students in OOA&D are required to complete a group project. Time will be provided during class to discuss project ideas and form teams.

The project is divided up into 5 parts.
Notice that the focus is on *Object-Oriented Analysis and Design,* as per the title of the course!

Part	Points
1: Topic and Team	5
2: Requirements & Analysis	
Group	40
Individual	20
3: Presentation	10
4: Finished Project, Report	20
5: Peer Evaluation	[*incorporated*]
5b: Peer Eval questions	5

Project Part 1

This is the initial part to your projects.
The goal for this part is to find your group, figure out your topic, and scope out what you intend to do. The instructor will then help ensure the scope of your project is achievable, or guide you towards an achievable project.

Create a .pdf document with the following details (should be about 1 page or so, in this order, with these headings):

a. **Team:** Full name (FirstName LastName) of each member in the group
b. **Title:** The title for your project (A very brief description)
c. **Description:** A longer description
d. Intended **platform/environment:** (Ruby on Rails, Mobile, etc.)
e. **Programming languages** to be used (and how familiar your team is with each)
Note: You must use object-oriented language(s)!

f. **Functionality:** A bullet-list of functionality you intend to complete. For example, if you want to implement Google, you just don't have time or resources to do so. But you could do parts of Google, such as indexing web pages and rudimentary search algorithms or the such.

Example (Use this template):

Team: Liz Boese
Cookie Monster

Title: Delivery Service

Description: A delivery service website that clients can lease access to which keeps track of their customers, what gets shipped and how often. We will set up for one client only.

Platform/Environment: PHP Yii Framework integrated with MySQL

Programming Languages:

Language	**Liz Boese**	**Cookie Monster**	**Superman**
PHP	Expert	Beginner	Beginner
JavaScript	Expert	Intermediate	Guru
HTML	Guru	Expert	Guru
CSS	Expert	Intermediate	Guru

Functionality:

- Customers can sign up
- Customers can log in
- Admin can log in
- Admin can set up what products get delivered each week
- Admin can add products
- Admin can add customers

[OPTIONAL] **Stretch Functionality:**

- End world hunger
- Learn how to fly

Submission

Submit a single .PDF document named **TitleOfYourProject_Part1.pdf** to you group GitHub account containing all of your work (do not put in a subdirectory).
Add a link to your group's GitHub project on the Google sites course website under the tab"Your Projects".

Project Part 2

Part 2 asks you to engage in analysis and design activities for your semester project. You will generate a detailed set of tasks that can be accomplished with your proposed system and a comprehensive design of that system. The goal of these analysis and design activities is to generate information that will allow you to start implementing the system with confidence.

Your deliverable for Part 2 is a single .PDF that contains the information listed below:

- **Team:** Liz Boese,
  Cookie Monster
- **Title:** Delivery Service

- **Project Summary:** What is the high-level overview of your semester project. What are you trying to accomplish? What will your system do when you are done? *This can be mostly copied from Part 1 that you submitted.*

- **Project Requirements:** Based on the project summary, what are the requirements and responsibilities for your system? List the requirements and their associated responsibilities in this section in a table. Be sure to break them down into small manageable separate requirements and label each one. These requirements need to be in tables. Each item must be numbered for reference.
  You can write the requirements in short sentences or in the Agile format.
  - 4 Separate tables for the requirements:
    - Business Requirements
    - User Requirements
    - Functional Requirements
    - Non-Functional Requirements

  If there are no business requirements then state so instead of a table.
  - [optional] you can add priority (Critical, High, Med, Low, Nice-to-have)
  - [optional] you can add topic/area (e.g., Login, Profile, DB, etc.)
  - [optional] you can add user(s)/actor(s) involved in each requirement

  Example

Business Requirements				
ID	**Requirement**	**Topic Area**	**User**	**Priority**
BR-001	All login userids must be the user's corporate email address.	Authentication	All	Critical
BR-002	Employee payments are distributed on the first of each month.	Payroll	Employees	High
BR-003	Monthly reports are generated between the hours of 2am-3am	Reporting	Admin	Medium

- **Use Cases:** Document how the system will support each task via a *use case*. Provide both use case diagrams and use case documents.
    - **Actors:** List all the actors who will make use of your system.
      Example:
      Actors: Admin, employees, customers
    - **Use Case Overview:** Create a single overview use case diagram depicting the main use cases the actors interact with. *Note that these should map back to your user requirements.*
    - **Sub-diagrams:** Create any necessary sub-diagrams to show more in depth the details of the use cases including any <<includes>> and/or <<extends>>. Be sure to label which use case you are describing in more detail.
    - **Use Case Documents:** List all the use case documents, one per use case. You should have at least 2 per 4448 team member or 4 per 5448 member in your group at a minimum.

- **Activity Diagram**: Each individual in your group picks a complex use case and creates an activity diagram that documents all of the possible paths through it. Typically, one activity diagram corresponds to a single use case, but if you can think of a way to combine multiple use cases into a single activity diagram that's fine too (as long as each team member draws their own activity diagram). *Note you must use UML 2.0 version with "swim lanes".*
    - Label the diagram with the Requirement ID #, Use Case ID #, Use Case name (short description), and the full name of the person who implemented this diagram.
    - Make sure you use swim lanes.
    - Make sure there is a system swim lane. When the system does something, make sure that is designated in the system swim lane!

- **Data Storage:** Discuss how you will persist data in your application. What storage technology will you use? Text files? XML? SQLite? Where will the data be stored? Describe the classes that you will use to access this data at run-time. *These classes should show up in your class diagram as well.*
    - Example:
      Data Storage: Flat text files.
      Classes:
        - PersonModel class to store information about customers, employees, and admin.
        - ReportModel class to store reports.

- **UI Mockups:** Create screen mockups for the user interface of various parts of your application.
    - What will a user see as they work through the tasks identified in your use cases?
    - What is the overall organization of your user interface?
    - How will data be displayed?

  - How will the user navigate from screen to screen?

    Note: it is okay to work on paper for this task and then scan in your work to include in your document. There are also free wire-framing tools online.

- **User Interactions:** Use the use case and activity diagram that you worked on individually and UI mockups to show a *sequence diagram* of the objects that will participate in the use case interaction. Recall that sequence diagrams *do not contain conditional constructs*, so be sure to clearly describe the interaction that is being displayed in the sequence diagram. If you find yourself needing to show an if-else situation, simply create two sequence diagrams, one that shows the true branch and one that shows the false branch.
  - Label each interaction with the Requirement ID #, Use Case ID #, and Use Case short description, and the individual person responsible for this one (each person must create a sequence diagram).
  - Make sure all classes are also represented in your class diagram.
  - Make sure all messages (method calls) are also represented in your class diagram.
  - Make sure you show the actor and the action that triggers the start of this use case!

- **Class Diagram:** Create a class diagram containing: what relationships the classes have, what are their attributes and (public) methods, what design patterns you may already know about are present in your design, etc. Be sure to show the visibility modifiers and relationships between the classes.
  - If you are making use of a framework such as Android then focus your class diagram on the classes you need to create. You can specify an inheritance or composition relationship if it helps but do not put all the attributes/behaviors of the framework classes into the diagram.
    For example,
    If I was using Java Swing I could simply specify “submitBtn : JButton” as an attribute in one of my classes without drawing the JButton class.
    If I want to make use of Java’s Observer interface, I may draw the Observer interface in the class diagram to show a class implements it.
  - Show all classes you are writing, including attributes and methods.
  - Show relationships between all classes.
  - Make sure all classes appear in your class diagram from all sequence diagrams.
  - Make sure all method calls from your sequence diagrams appear in your class diagram.

How to Scope

While I have asked for a non-trivial amount of information above, you should only generate the information you need to achieve your desired functionality given the size of your team. Each member of your group should expect to work on one to two use cases (possibly with other team mates). A 5-person team is looking at 10-20 use

cases (2-person 5448 team 8 use cases), although I would be surprised if any team identifies 20 tasks that their system can perform. Use these guidelines to scope the work of this project part and really focus on generating information that will guide your implementation efforts.
The point breakdown of this project part is as follows:

Section	Points
Project Summary	2
Requirements	5
Use Cases	10
Activity Diagram *(individual score)*	10
Data Storage	3
UI Mockups	5
User Interactions *(individual score)*	10
Class Diagram	15
Total:	60

Assessment
In general, you will be graded on the thoughtfulness and completeness of answering the questions above, along with the overall quality of the work.

Submission
Submit a single .PDF document named **TitleOfYourProject_Part2.pdf** to your GitHub containing all of your work (do not put in a subdirectory).

Project Part 3: Presentation
Present and demo your final system.

A **demo** of the final system during the last week of class – to the rest of the class. You will have a limited number of minutes for your group, including set-up *(as soon as a group ends, your time begins).* Your presentation should consist of two parts:

- A brief demo walk-through demo of the system showing one or two use-cases.
- Then focus on a particular design pattern or two, or the diagrams that you did to define the project. Describe what you applied and why, or what you could apply and how.
  - Consider what is interesting about your project that the rest of the class can learn from.
  - Do not try to show everything, just focus on some part of the whole to share with the group.

Presentation tips:

- Use 20-point font minimum
- *Pictures say a thousand words* much easier to convey information than lots of words on each slide

- Face the audience when you speak
- Make sure we can read your content. Don't show the whole class diagram; zoom in to some classes showing a particular design pattern to discuss.
- Include title and group member names!
- Everyone must speak during presentation.
- You have only 6 minutes to present including setup time. It is up to you and your group to ensure you can finish in the time allocated.

You must also record a demo of your software and post it to your GitHub (do not put in a subdirectory). You can use this video during your presentation. Label your presentation **TitleOfYourProject_Video**.

Submission
Sign up for a time to demo.
Submit a single .PDF document named **TitleOfYourProject_Part3.pdf** with your slides and a link to a demo of your software to your group's GitHub (do not put in a subdirectory).
Distance students: Either record your presentation together as a group (use Google Hangouts/Skype/etc.) or you can each do the full presentation yourself and submit one per person in your group. Use your cell phone camera, Jing or other screen video capture, etc.

Project Part 4: Final Report
Project Part 4 asks you to complete the work on your semester project.

The required deliverable is a **report** that documents the final state of your system. List these questions in your document with your answers.

1. What features were implemented?
2. Which features were not implemented from Part 2?
3. Show your Part 2 class diagram and your final class diagram. What changed? Why? If it did not change much, then discuss how doing the design up front helped in the development.
4. Did you make use of any design patterns in the implementation of your final prototype? If so, how? If not, where could you make use of design patterns in your system?
5. What have you learned about the process of analysis and design now that you have stepped through the process to create, design and implement a system?

Submission
Submit a single .PDF document named **TitleOfYourProject_Part4.pdf** to your group's GitHub containing all of your work (do not put in a subdirectory).

Project Part 5: Peer Evaluation and Team Dynamics

Each person individually fills out the peer evaluation form (see Moodle) and submits into Moodle as an Excel spreadsheet (do not convert to .PDF). This will be incorporated into the individual project grade for groups that have someone not helping much at all or someone who put considerable extra effort helping others and/or developing the project as a good team player. *However, taking over the project and not being a good/helpful team player is bad!*

Please note: Giving everyone the same number does not count as filling out this form! You will need to fill in the comments to support your evaluation, as well as the Team Dynamics question. For Team Dynamics, enter the letter a, b, or c for "Which one" to specify which question you are answering.

Submit to Moodle as: **TitleOfYourProject_YourLastName_PeerEval.xls**