Fall 2014 Semester Project

CP Website: http://lyle.smu.edu/~jmmurphy/cproj/

A semester project combining the brainpower of the students from CSE 3330 (Database Concepts) and CSE 3345 (Graphical User Interface Design)

Executive Summary

The Project Overview

You and your team are a recently established software startup firm. Your team is adequately equipped with the brightest minds from SMU in both Database systems and Graphical User Interfaces, and now you are ready to change the world with your latest "big idea." Unfortunately as college students, there isn't much money to jumpstart the project. Using your SMU connections, your team was able to set up a business meeting with Professors Fontenot and Murphy to discuss seed funding. Your goal is to create a Project Proposal to sell your "big idea" to both Professors and obtain the required startup funding[1].

Major Project Due Dates

Date	Activity/Deliverables
September 4, 2014 @ 9:30am	Project Kickoff
September 14, 2014 @ 5:00 – 8:00pm	Product Proposal Presentation with associated documentation
October 2 – October 17, 2014 @ 11:59pm	Mini Collaborative Project Assignment
October 10, 2014 @ 11:59pm	Functional Specification and Design Document
October 17, 2014 @ 11:00pm	Iteration 1 Feature Set Plan
October 20 – November 2, 2014	Iteration 1
November 3 – 6, 2014	Testing for Iteration 1
November 7, 2014 @ 11:00pm	Iteration 2 Feature Set Plan
November 10 – 23, 2014	Iteration 2
November 24 – 25, 2014	Testing for Iteration 2
November 30 – December 5, 2014	Iteration 3 (Feature Freeze - no new features are to be implemented; therefore, no iteration feature set plan is required.)
December 12, 2014 @ 8:00 – 11:00am	Final Project Presentation and Report Due with Completed Prototype.

1 Please note that the professors will not be supplying any money to anyone at any point throughout the semester. This is a hypothetical pitch.

The Details

What you will be doing

The goal of this project is to develop an idea into an actual software product. We are deeply interested in your creativity and innovation with respect to coming up with a "cool" product. However, we are primarily concerned with your ability to implement the idea technically (as these are technical courses in which you are enrolled). You can reinvent the wheel, but your product pitch should be clean, succinct, and you should be passionate about the project.

Below, we provide more detail on the various deliverables for which each team will be responsible.

The Product Proposal

The Product Proposal will serve as a "product pitch" or "elevator speech" for the product your team decides to pitch to us. The Product Proposal will effectively explain the product in sufficient detail for the reviewers to understand your idea and vision, and what features you intend to provide to the users of your system. The Product Proposal should contain the following:

1. **Team Introduction**: This will include your team name, product name, members of your group, and each member's role.
2. **Product Idea**: For this project, you and your team will incubate a concept/product/application that will:

    a. Implement a three-tier software architecture
    b. Provide a web site/web application and native Android UI client
    c. Equally draw upon the disciplines taught in both CSE 3330 and 3345
    d. Be deemed appropriate for Academia
    e. Other than the previous requirements, you and your group are free to create anything.
3. **Market Analysis**: This section explores which products already exist in your project area. Who are your competitors? What features do they provide? What user groups do they appeal to (target audience)? What are they lacking, etc.? Use your analysis to assert the validity of your project, how your project differentiates itself, provides better experiences/services/features, and ultimately merits its creation and success.
4. **Product Features**: Using your project idea and Market Analysis generate a high level feature list that your project will offer. This sections needs to exhibit adequate thoughtful consideration. Project leveling will primarily be done using that which is communicated in this section.
5. **Intended User Group(s)**: What user groups is your project targeting? These should closely relate to "User Profiles" that will be used during the Design Phase for your project.

Mini Collaborative Project Assignment

An assignment for both CSE 3330 and CSE 3345 students. This homework assignment will be a minified 3 Tier web application and gives students the experience of the Collaborative Project but on a much smaller level. ****This assignment will be due roughly at the same time as the Iteration 1 Feature Set Plan and occurs during the weeks surrounding Fall Break (which is when some Midterms are scheduled). The professors strongly recommend that each team work diligently, **start early**, and plan accordingly to accommodate this assignment,and school work such as Midterms.****

The Functional Specification and Design Document

The Functional Specification and Design Document will provide the reviewers a detailed understanding of the state of your thinking with respect to the design and implementation of your forthcoming product prototype. By the submission deadline, your team may have already made some technical decisions and done some implementation, but you may also be evaluating options for some other aspects of the product.

This document will be completed in the form of a Spreadsheet via Google Drive. You will be provided read-level access to the spreadsheet. One person from each team should:

1. Make a copy of the spreadsheet in your local Google Drive.
2. Share with your team.
3. Complete the spreadsheet by the due date.
4. Share the spreadsheet with Prof. Fontenot and Prof. Murphy.

The Functional Specification and Design Document will contain the following:

- A List of System Actors
- User Story Summaries
- Detailed User Stories
- Data Model
- Data Model Verification
- User Profiles
- Object/Action Analysis and Matrix
- Paper Prototypes
- A Usability Test outline that describes how you'll be testing your software. This should include the number of users you will be testing, which usability factors you will be focusing on, which type of test you will be using, how you are measuring usability, etc. **Note, Professor Murphy will be one of your usability testers. Also, you need to use real people that are outside your team as testers!**
- A Usability Report that describes the outcome of your usability tests.
- Software Lexicon

Source Control Management and Software Process

Each team will be using Git for source control management. In addition, each team will set up the following environments for software development (See Figure 1 as well)::

- Local Development - Developer machine
- Working / Internal Testing - GitHub
- Production - Cloud Server

Developer Machine

Each developer will

- Have a full WAMP, LAMP, or MAMP setup allowing them to host a local database, server, and web client on their machine.
- Pull source code from the GitHub repo to their machine.
- Push source code to GitHub.

GitHub

GitHub will serve as a working repository for developers. After a code snippet, feature, or bug fix has

been implemented and tested on a local machine, a developer will push their code to GitHub to share with other team members.

In addition, each team will use GitHub for the following:

- **Bug Tracking** - When a team member or test team member finds a bug with software, they will use GitHub to create a bug report which documents:
    - What error occurred
    - How to reproduce the bug
    - Any other helpful information.

  The software team will be responsible for assigning the bug to the proper developer and fixing the bug in a timely manner.
- **Project Document Hosting** - Each team will create a folder at the top level of their repository called **Project_Documents**. In this directory, teams will post their project document deliverables on the day they are due. This is where the professors will look for deliverables to be graded for the project.

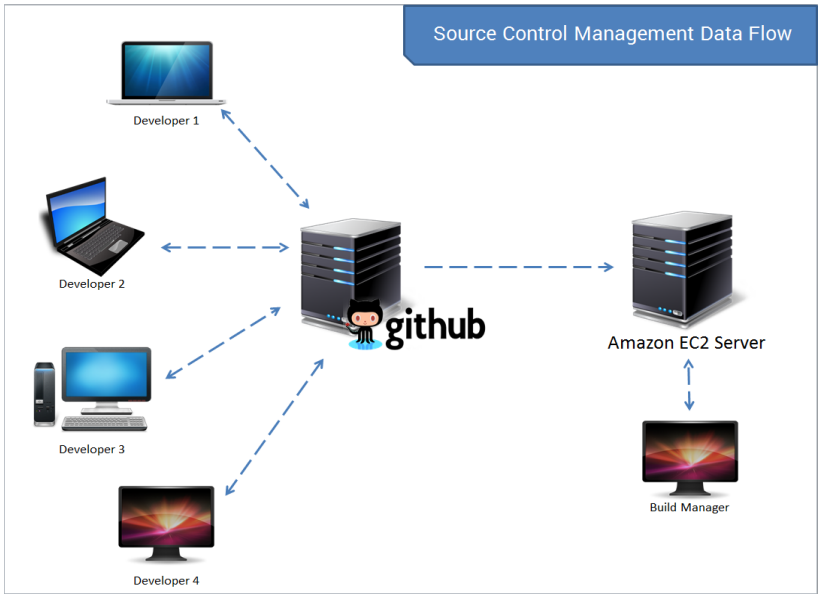

Figure 1 - Management of Source Code Data Flow

Cloud Server

Each team will use a cloud-based server which will serve as the team's production environment, and host code that has been tested and "blessed" by the team's build manager. Each team needs to have a build manager which is responsible for approving code revisions from the GitHub repo and pulling it into the production environment. **Developers are not to use the Cloud Server for development.**

Iteration Feature Set Plan

The idea behind each iteration is for each team to decide on small manageable chunks of work that can be completed in two weeks. This is an attempt to prevent teams and individual developers from creating lofty goals with no specific deadlines because this often results in failure. To help define these small manageable goals, teams will create Iteration Feature Set Plans for each iteration. To create the plans each team needs to create a Milestone for each iteration in their GitHub repositories and create Issues which will document each user story that will be implemented as part of the

iteration. In addition, the feature set plan should help each team form a clear picture of the roadmap for their product, assign work to teammates, and prevent teammates from duplicating each other's work. Once a user story has been assigned to an iteration, it should also be indicated on the Story Summary sheet of the Functional Specification spreadsheet.

At the end of each iteration, a separate testing team will use the software development team's iteration feature set plan and create a test suite against it. Each Milestone and associated features requests / issues needs to be detailed enough to allow professors and the test team understand what stories are expected to be implemented and work.

When planning features, make sure they are *specific*, *measurable*, *attainable*, *relevant*, *and time-bound*. If there is a story that cannot be finished in two weeks, break it up into multiple levels and create a minimal pass through for the first iteration and complete it on the second iteration. See the Tips for Creating Successful Iteration Feature Set Plans document for help in creating your Iteration Feature Set Plan. See the GitHub Help Section on the CP website for videos on how to create Milestones and Issues with GitHub.

Iterations

This project will be composed of 3 Iterations. Before the beginning of the first two iterations, each software team will create an iteration feature set plan detailing the work for the upcoming week. At the end of each iteration, the software team is responsible for the following:

1. Their build manager pulling all code pertinent to the feature set plan from the GitHub server to the production server to allow for testing.
2. Posting an Android APK to the production server's website for testing.
3. Providing instructions, test accounts, etc. to the testing team and professors for how to test their product.

Testing

Each software team will be assigned a testing team. At the end of the first two iterations, the testing team will use the software team's iteration feature set plan and test against it. Any bug, usability issue, visual error, etc. that is found, whether it was a part of the features set plan or not, should be logged on the software team's GitHub page as an **Issue**. An **Issue** should be as detailed as possible and contain the following:

- A brief yet descriptive title of the problem/bug/issue.
- Tester's name and contact information. If the software team has a question with the issue they should be able to contact the tester to get additional details.
- Bug Frequency. How often can this bug be reproduced: once, every time, 3 out of 10 times, etc.
- Reproduction Steps. Detailed step by step instructions for how to reproduce the bug. If it only occurred once, try your best to remember the steps.
- The bug or error. Describe what the bug actually is.
- Picture. If possible attach a screenshot/image of the problem. Use a paint program to highlight the bug area if it isn't obvious.

See Figure 2 below for an example.

Prototype Submission and Presentation

You will submit the following:

- A final working prototype
- A final design report
- A presentation/pitch of your product/prototype to the reviewers

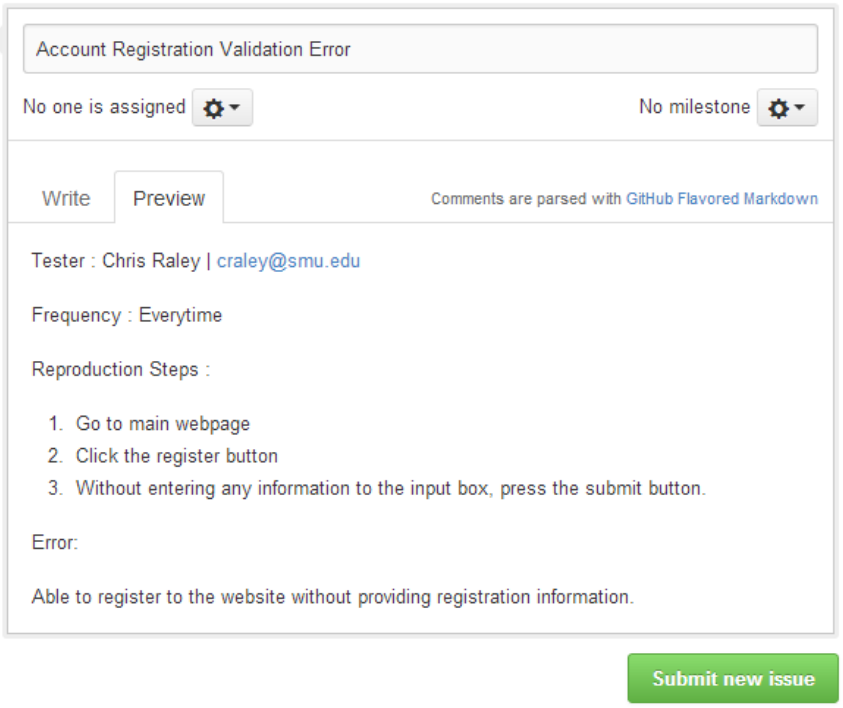

Figure 2 - Bug Tracking in GitHub

Additional Requirements

- The Database students are in charge of creating a RESTful API. See the following links for information on REST.
  - RESTful Web services : The basics
  - Designing a Beautiful REST + JSON API (video and ppt)
  - Best Practices for Designing a Pragmatic RESTful API
  - SLIM PHP Framework - Framework for writing RESTful APIs.
- GUI students are in charge of creating a professional UI aesthetic and appeal. Both the Web and Android clients need to have a cohesive look and feel. The color palette, font choice, iconography, layout, etc. all need to be taken into consideration and should not be left to chance or defaults. We want something that is attractive and easy to understand and use. This is something that should **NOT** be tacked on at the very end, but instead should be considered from day 1 of the project.
- You can't use Bootstrap, Foundation, Ink, or similar projects for UI.
- The mobile application only needs to be designed for ONE screen size (phone or tablet) and ONE orientation (portrait or landscape).

Tips for Success

- **Be Bold**: Begin the creation of a brand. Choose a team/company name that is appropriate. Create a logo for your team/company (the logo doesn't have to be fancy). This brand should be used throughout the project.
- **Be Professional**: Your Product Proposal and technical documentation should be professional in nature. This does not only refer to content/command of the English language. The document itself should look professional.
- **Be Brief**: Craft your proposal and other documentation to succinctly communicate its intent while still being clear and covering all required components. We prefer "short, sweet, and to the point."
- **Be Engaged**: Each team member must participate in the creation of the Project Proposal and other documentation as well as the implementation. It is not acceptable for one or more team members to only participate in documentation and not in technical implementation.

Project Evaluation

Overall, the project will be 30% of your semester grade in both CSE 3330 and CSE 3345.

- Project Proposal: 5%
- Functional Specification and Design Document: 15%
- Iteration 1 Feature Set Plan: 2.5%
- Iteration 1 Test Results: 2.5%
- Iteration 2 Feature Set Plan: 2.5%
- Iteration 2 Test Results: 2.5%
- Final Project Presentation and Report: 20%
- Individual Contributions to team: 20%
- Individual Git commit activity and productivity: 20%
- RESTful API - 10% (DB Responsibility)
- User Experience and UI Aesthetic - 10% (GUI Responsibility)

Caveat Lector: There is a world of difference between a good product and a great product. Unfortunately, it is nearly impossible to objectively define what separates good from great; we all just know it when we see it. Therefore, a good project will receive a maximum grade of B overall for each team member. Members of teams producing a great project will be eligible for an A.

Team Dynamics

We recognize that situations occur in which a particular team member does not "pull their weight". A non-productive team member can cause other team members to become disgruntled. In order to address this problem, we have put into place several review mechanisms:

1. **The Review Team**: The professors will be monitoring team/individual Git commits throughout the semester.
2. **End-of-project peer and project reviews**: Each team member will have the opportunity to critically evaluate the performance and contribution of other team members.
3. **Team member grievance process**: In the event that a team member becomes completely unproductive and fails to communicate, the remainder of the team (or a majority of its members) may initiate a grievance process with the professors.
    - **Step 0**: The team will email the non-producing team member and inform said team

member that the grievance process is about to be initiated if work habits do not change. The professors should be copied on this email.

- **Step 1**: If behavior of non-producing team member does not begin to become a contributing member within 3 calendar days from the sending of email from Step 0, the team may contact the professors via email and request assistance. The professors (one or both) will set up a meeting with the non-producing team member to address the issue.
- **Step 2**: If the non-producing team member does not begin to become a contributing member within 3 calendar days from the meeting with the professors, the team may dismiss the noncontributing student from the team. At this point, the non-contributing team member must complete the project alone. Note: A student may not be dismissed from a team during the last two weeks of the project.