Objectives: Project

- Discuss project requirements for course
- Discuss client interactions

Time for the Computer Science...

- Software engineering
  - Design, development, implementation in cycles
  - Feedback from client
  - Collaborative software tools

Planning Stage

- Talk to client, gather requirements of application
  - What is the application’s functionality
  - What must it do and what would they like it to do?
  - Where are their priorities?
  - Clarify as much as possible!
- Analyze requirements
  - Is it possible? Within the time frame?
  - If there are multiple ways to implement something, which should you do?
  - Anticipate difficulties (technology, implementation, …)

Planning Stage

- Write out requirements
  - Get rid of any ambiguities as soon as possible
  - Know all functionality, behavior
    - Required input/output
    - Clarify as much as possible
  - Otherwise, disputes with client
- Develop Work Plan
  - Steps to complete task
  - High-level, on Course Web Page
  - Drill down: divide up responsibilities

Project Overview

- Think about what needs to be clarified
  - What can the application do?
  - What do you need to know to make the UI, the backend?
  - Any hidden assumptions?
- Discussion in a bit...

Approaches to Software Design

- Inside-out
  - Develop a system
  - Add an interface
- Outside-in
  - Develop the interface
  - Then build the system to support it

When design decisions are made, either the developer must conform to the user or the user must conform to the developer.
Approaches to Software Design

• **Inside-out**
  - Develop a system
  - Add an interface

• **Outside-in**
  - Develop the interface
  - Then build the system to support it

When design decisions are made, either the developer must conform to the user or the user must conform to the developer.

April 26, 2016
Sprenkle – CSCI335

Traditional CS Courses are almost entirely inside-out.
Modern systems need to be designed outside-in to be effective.
Web sites especially need to be usable.

Our approach to the project

Project Architecture

User Interface
[HTML, CSS, JavaScript]

Web Application Server

Database

Spring Term Festival:
Last Friday of Term

April 26, 2016
Sprenkle – CSCI335

Requirements, Design, and Work Plan

• Requirements
• Design: steps to complete project
  - Includes what will be implemented and the technologies used to implement each piece
• Work Plan: a tentative plan for what parts of the work each member is charged with doing
  - Prioritization of features

April 26, 2016
Sprenkle – CSCI335

Deliverables

• Deliverable 0: Project Requirements, Design, Work Plan
• Deliverable 1: Static HTML Mockup
  - Clarify flow, appropriate results
  - Feedback on presentation, usability
• Deliverable 2: Web application Implementation, I
  - High-priority functionality implemented
• Deliverable 3: Web application Implementation, Final
• Deliverable 4: Documentation
  - For users and for system administrator
• Deliverable 5: Demonstration

April 26, 2016
Sprenkle – CSCI335

Deliverable 0:
Project Requirements,
Design, Work Plan

Deliverable 1:
Static HTML Mockup

Deliverable 2:
Web application Implementation, I

Deliverable 3:
Web application Implementation, Final

Deliverable 4:
Documentation

Deliverable 5:
Demonstration

April 26, 2016
Sprenkle – CSCI335
Requirements Gathering

- Clarification of requirements
- Involves asking lots of questions
- Talk through the application
  - Flow chart of what happens

Requirements Gathering: Questions

- What does the user want to do?
  - Go through a variety of use cases
    - Common case, error case
    - Part of your job is organizing these use cases
- What is needed to do that task?
  - User input? Saved data? Other sources?
- What does the user see?
  - Draw on whiteboard, use paper
  - What is interface?

Requirements on Wiki

- For each feature, you need to describe
  - The feature
  - The prerequisites – what needs to be true or to have happened before a user can use the feature
  - What a user sees (Be specific; e.g., are results in a particular order?)
  - What a user does on the page (Be specific)
    - Any input that needs to be validated? Any constraints?
  - Example use cases - describe some typical situations of what a user can/will do
  - Relative priority of feature

Practice Application

- Students in interdisciplinary majors have difficulty finding courses to take
  - Example: Neuroscience major
    - http://www.wlu.edu/neuroscience-program/about-the-program/neuroscience-major-requirements
    - Not designated as NEUR courses
    - Change each semester
  - Goal: create an application that helps students find courses that will help them towards their major

Plans

- Read overview of project
  - http://www.cs.wlu.edu/~sprenkle/cs335/project.php