

Senior Project 2017

November 6, 2017

Note: This slide is posted on Moodle and also Facebook group “ITS/CSS 400 Senior Project”.

1 Review of senior project stages

2 Writing specifications

- Project Concept
- **Requirements Specification**
- **Design Specification**
- Implement
- Test

why do it?

what will it do?

how will it do it?

do it

did it do it?

- 1 Deliverable 1: Project Concept (10%) Done
 - Faculty has evaluated this Deliverable
- 2 Deliverable 2: Requirements Spec (20%) **Deadline 6 Nov**
- 3 Deliverable 3: Design Spec (10%) **Deadline 3 Dec**
 - Faculty will evaluate Deliverable 2+3 after 4 Dec
- 4 Presentation 1: Project Concept (5%) Done
- 5 Presentation 2: Mockup (5%) **Deadline 13 Nov**
 - Prepare a 3-4 minutes video. Post the video file online (e.g. Youtube) and give its link in Section 3.5 of Design Spec. Also upload the file to SPM system.
- 6 Presentation 3: Prototype - 35% **Nov 30 and Dec 1 (1pm-4pm)**
 - You can select your preferred date
 - 8 minutes presentation per group including video/demonstration

Requirements Specification

- List the things that your system must/shall/may do
- Purpose: force you to think about features; (potential) clients clear on what you will do; you are clear on what the client expects, etc

Requirements Specification: (Recommended)

Sections

2.1 System Description:

2.1.1 Perspective

2.1.2 Functions

2.2 Requirements

2.2.1 Interface requirements

2.2.2 Functional Requirements

2.2.3 Performance Requirements

2.2.4

Requirements Specification: Sections

2.1 System Description: Brief description of your system

2.1.1 Perspective:

- How does your system relate/interface with other systems and users
- Block diagram showing your system as a block interacting with other blocks for Users/other Systems

2.1.2 Functions:

- List the main functions/features of your system
- Block diagram showing the functions

2.2 Requirements

2.2.1 Interface requirements

- How does your system interface with other systems/users?
- External interfaces only
- Optionally may divide into sub-sections, such as: User interfaces, Hardware interfaces, Software interfaces

Examples:

- IR1: The system must interface with the Authentication System.
- IR2: The following messages sent to the Authentication System must be supported: DoesUserExist, IsPasswordCorrect
- IR3: The following messages sent from the Authentication System must be supported: UserResponse, PasswordResponse

2.2 Requirements

2.2.2 Functional Requirements

- What will the functions/features do?
- Longest list of requirements, most important
- Split into sub-sections based on functions/features. For example, Authentication, Course Setup, Evaluation, ...

2.2 Requirements

2.2.2 Functional Requirements

Examples:

- Function: Evaluation
 - Eval 1. The system must automatically fill in the students name in the evaluation form.
 - Eval 2. The system must allow the staff to define the names and number of scores for the evaluation.
- Function: Reporting
 - Report 1. The system must be able to group data into the following data sets grouped by: Semester, School, Program
 - Report 2. The system must calculate the following statistics for any data set: Mean, stdev, min, max, Q1, Q2, Q3

2.2 Requirements

2.2.3 Performance Requirements

- Not everyone will have them
- Measurable, verifiable: computation complexity, speed, size, reliability measures, etc

2.2.4 Design Constraints

- What constraints are imposed by standards and other external factors?
- Not everyone will have them

Design Specification

- Describes how the features will work in your system.
- The design specification should include both “what” and “why”. That is, what is the design and why did you choose to design it that way.

Design Specification: (Recommended) Sections

- 3.1 System Architecture: A high-level view of your system design
Similar to the “System Description” section of the Requirements Specification, but in more detail and focusing on how the system works, not what it will do. You may include a block diagram showing the system components/modules and describe each briefly.
- 3.2 Detailed Design: A low-level view of your system design
For each of the components identified in the System Architecture, give details of how they will be built. You may use different notation for the design, e.g. UML, block diagrams, pseudocode, message sequence charts, flow charts, hardware designs.

Design Specification: Sections

3.3 Feature List: In your Requirements Specification you identified what your system should do. Considering those requirements and summarise key features that you actually deliver in your mockup system and give the status of each feature

Example:

- 1 Combat System: Player can combat with either players or monsters.. (with physicals attack or magic attack)
Feature status: Complete Partial Initial
- 2 Feature: short description of the feature
Feature status: Complete Partial Initial
- 3

Design Specification: Sections

3.4 Test Results: A description of how you tested your system, especially the completed features, and what results were obtained.

Tests may be described informally (e.g. using text like We asked 5 students to complete evaluations of 3 courses over a 1 week period. The feedback from the students was used to improve the website, in particular reducing the number of clicks needed to complete the evaluation from 5 down to 2.) or formally (e.g. using tables of tests and results).

Design Specification: Sections

- 3.5 Link to video demonstration of Mockup/Prototype system: Groups will be able to choose how they demonstrate their projects, with the main limitation being the time available (3-4 minutes). Demonstrations may make use of computer screen capturing of your current system, talking and slides. Post the video file online (Youtube) and give its link in this subsection. Also upload the video file to the SMP system.