

## Requirements Specification Project

**Issued:** Tuesday, October 2

**Due:** Tuesday, December 11

### Purpose

This assignment has several objectives. It is intended to:

- introduce you to the content of a typical Requirements Specification Document
- familiarize you with techniques for informally, but systematically, documenting the requirements of a software system
- give you an opportunity to practice and improve your verbal-communication skills
- allow you to practice working in a small team
- provide you with a subject for subsequent development of formal specifications

### Assignment

This is a team project. I will compose teams of five or six people. You will be notified by email.

Your team's assignment is to determine and specify the requirements of a software system. You are encouraged to develop a marketable system with attractive features. I expect a good system to be complex, but not complicated. Your team *must* discuss its choice with me. Here are some suggestions:

- Irrigation Control System
- Home Security and Control System
- Collaborative Work/Scheduling/Meeting System
- River-Permit Lottery System
- Airline Reservation System
- School Administration System
- Hotel Administration System

- Campground Reservation System
- Hunting/Fishing License Distribution System
- Department of Motor Vehicles Administration System
- Book-Lending Library Administration System
- E-Commerce Site-Building System

We do not have the luxury of a real customer. Therefore, you or I must pretend to be the customer. You can use your creativity to develop an attractive system, or you can ask questions of me. I don't think I'm much more qualified to play the customer than you are.

Your team's primary deliverable is a Requirements Specification Document. The content is discussed below.

Each member of your team is also responsible for sending me, via email, a short progress report each week. To clarify:

one progress report, per week, per member

A progress report should describe your project-related activities for the week. You can ask questions, and I'll try to answer them. You can (and should, if necessary) complain about your teammates, and I'll take their behavior into consideration. Do not send drafts of your document. Progress reports are confidential.

Your team's document must be typeset using a system such as L<sup>A</sup>T<sub>E</sub>X. If you choose to use L<sup>A</sup>T<sub>E</sub>X, see:

```
onyx:~buff/classes/471/pub/latex
http://cs.boisestate.edu/~buff/etc/html/latex2e/latex2e_toc.html
http://tug.ctan.org/tex-archive/documentation/beginlatex/html/index.html
```

When choosing a typesetting system, remember that:

- Your team's work must be integrated into a single document.
- Your team's document must have a table of contents.
- Your team's document must have page numbers.
- Your team's document should have diagrams.

## Document Structure

Your document will be graded according to the following rubric.

- General Description 20%
  - Executive Summary
  - Feature Description
  - User Characteristics
  - Environmental and Social Impact
  - Perspective, Scope, and Boundaries
  - Assumptions, Dependencies and Constraints
  - Standards Compliance
  - Glossary and Acronyms
  - References and Appendices
- Functional Requirements 30%
  - Feature Description
  - Use Cases
  - User Interfaces
  - Hardware Interfaces
  - Software Interfaces
- Nonfunctional Requirements 10%
  - Time/Space Performance
  - Reliability
  - Portability
  - Security
- System Tests 10%
  - Test Procedure
  - Pass/Fail Criteria
- Users' Manual 20%
  - Correctness
  - Completeness
- Overall 10%
  - Comprehensibility
  - Consistency
  - Maintainability
  - Portability
  - Reusability

Our textbook also discusses requirements, in Chapter 10.

You may also want to investigate *IEEE Standard 830: Software Requirements Specifications*. For example:

[http://en.wikipedia.org/wiki/IEEE\\_830](http://en.wikipedia.org/wiki/IEEE_830)

You need not use this format exactly, but your document should contain most of the relevant information.

## Sample Features

Assuming your team chose to develop requirements for an Irrigation Control System, the following ideas are intended to suggest what the system might do. They are in no particular order.

- Users include groundskeepers. Make your system easy to use.
- Your system's database needs to maintain times and periods. Give a great deal of thought to the syntax and semantics of these entities.
- Consider how your system should behave when confronted with invalid input. Also consider conflicting inputs.
- Your system's database also needs to maintain valves, sprinklers, and who knows what else. Try to avoid arbitrary limits. Allow these entities to be named and grouped in nonrestrictive ways.
- Consider how your system should behave in the presence of faults (e.g., a power failure).
- At least one of your system's inputs is the weather: past, present, and future.
- Your system should log its activities. This information, as well as the system's current status, should be available in the form of concise reports.
- Use a general-purpose computer, rather than an embedded controller. Consider different kinds of terminals, including hand-held cordless devices.
- Cordless communication invites unauthorized use. Make your system secure.
- Consider textual and graphical interfaces.