

Product Name  
Team Member Names

**[Product Name]**

[Product Team Members]

# **System Design Specification and Planning Document**

Draft X  
Date

CSE 403 - DEMS

Product Name  
Team Member Names

## Revisions

<b>Version</b>	<b>Primary Author(s)</b>	<b>Description of Version</b>	<b>Date Completed</b>
Number	Full Name	Information about the revision.	00/00/00

Product Name  
Team Member Names

## System Architecture

*This section provides a detailed definition of the system software components. It identifies the major modules and their functionality, and the interfaces between modules, required to implement the system. It should incorporate the design principles we discussed in class. It should address both the design of the system from the customer's viewpoint, as well as that of the developer or administrators.*

***[Note: the use of UML was required in a previous course. You are not required to use UML, but you should have the same basic kinds of information as the UML would convey.]***

*Include:*

- *Two views of the system architecture, with one view being a picture diagram one being a UML class (object) diagram of the major classes. The two views should describe different aspects of the system.*
- *Two UML sequence diagrams outlining two of the use cases from your SRS.*
- *The high level database schema you will be using. If you are not using a standard database, describe how you are storing the data and its organization.*
- *If there is a reasonable design alternative, outline why you made the selection you did. If there are particular assumptions in your design, identify them.*

### **1. Introduction**

### **2. <System architecture view 1, your choice>**

### **3. Design view – UML class diagram**

### **4. Process view – UML sequence diagrams**

### **5. Database Schema**

### **6. Design Alternatives and/or Assumptions**

Product Name  
Team Member Names

## Development Plan

### 1. Team Structure

*Describe your team structure (how you have organized the team, what are members' roles and responsibilities). You may find yourself revisiting your organization during the lifecycle of the project.*

*Describe the way your team will communicate, for example: date/time of weekly meeting, location of weekly status reports, wiki, email alias.*

### 2. Project Schedule

*Identify milestones (external and internal), define tasks, and specify the team member responsible for each task where this is known. It's ok to have some flexibility in the schedule, especially towards the later weeks. This should reflect your actual plan of work, possibly including items your team has already completed. Consult your SRS Feature List for input.*

*You will have three required releases (dates yet to be set)*  
Zero feature release  
Beta release  
Final release

*To build a schedule, start with your major milestones (tend to be noun-like) and fill in the tasks (tend to start with a verb) that will allow you to achieve them. A simple table or spreadsheet is fine for this size of a project (see example below). Schedule tools, like MS project, are also good as they track dependencies, but may have some ramp up to learn.*

<b>Task/Milestone</b>	<b>[Estimated effort]</b>	<b>Date due</b>	<b>Resource(s)</b>
Set up bugzilla	1 day	April 20	Betty
Set up initial db	...	April 25	Veronica and Reggie
...			
<b>GUI talks to DB</b>		<b>April 28</b>	
...			
<b>Beta Release</b>		<b>May 12</b>	
....			

### 3. Risk Assessment

*Identify the top five software development high risk areas particular to your project along with an analysis showing why, at this point, you believe they will not become "show*

Product Name

Team Member Names

*stoppers” for the project, and what your risk mitigation paths would be if they arose. Use the table provided.*

<b>Risk</b>	<b>Chance of occurring (High, Med, Low)</b>	<b>Impact if it occurs (H,M,L)</b>	<b>Steps taking to increase chance it won't occur</b>	<b>Mitigation plan should it occur</b>

## Test and Documentation Plan

### 1. Test Plan

*Describe what aspects you plan to test and why they are sufficient, as well as how specifically you plan to test those aspects in a disciplined way. Discuss unit test, system (integration) test, and usability test strategy, along with any specific test suites identified to capture the requirements (SRS). Identify the mechanism that you will use to track bugs that occur during use and testing.*

*We will not have covered this material in class by the time the SDS is due. Do your best and plan to revisit this section later. Below is a sample structure for the section.*

#### *1.a Unit test strategy*

- What this will test (coverage/purpose)*
- How the tests will be developed, by all or by particular team members*
- Frequency with which they will run*

#### *b. System test strategy*

- What this will test*
- How the tests will be developed*
- Frequency with which they will run*

#### *c. Usability test strategy*

- What this will test*
- How the tests will be developed*
- Frequency with which they will run*

#### *d. Adequacy of test strategy*

#### *e. Bug tracking mechanism and plan of use*

### 2. Documentation Plan

*Define the documentation that you plan to deliver with the system, e.g., user guides, admin guides, man pages, help menus.*