PREFACE {LESS THAN 1/2 PAGE}
This should let the reader know that this is the second version (a revision of) the Contract, the previous version having been delivered on Feb 6.

DOCUMENT CHANGE HISTORY
Date and note any changes to the document here. For example, from Contract I to Contract II, I bumped the previous section 5 to section 6 this time, and added a new section 5 (validation and verification). This can be very brief, but will save your readers a lot of time. Such sections of deliverable documents are routine/common.

GLOSSARY
Define anything you need to here. Omit if not appropriate/necessary.

1. Introduction {~1/2 page}
* Who is your client?
* What is the problem they are trying to solve?
* Other relevant history? (Interesting/relevant story?)
* Other?

2. User-Level Requirements Specification {~1-2 pages including diagrams}
* Include Context Model Diagram(s) w/ hierarchical numbering of blocks (see example below)
* Include Process Model Diagram(s) w/ matching hierarchical numbering
* Include User-level description of each block (w/ matched numbering—see examples below)
The following is an example Context Diagram for a toy cooking system.

The following sections 2-A. through 2-D. are examples of numbering matching Figure 1.

2-A. User Interface
So here is an example, the level 1 discussion of the user interface. This might be (for example) the level of the Context Model Diagram in the User Requirements Specification. Here you would describe the basic role/function of this major area of functionality.

2-B. Database
An example of a second major area at the top level.

2-C. Processing
An example of a third major area at the top level.

2-D. Oven Control
An example of a fourth major area at the top level.

3. System-Level Requirements Specification {~2-4 pages including diagrams}

3.1 Functional Requirements
* System-Level Context Model Diagram (see example below)
The following is an example of a system-level context model diagram.

![Diagram](image)

**Figure 2**: System-Level Diagram.

The following sections 3.1-A. through 3.1-D. are examples of numbering matching both Figure 1 and Figure 2.

### 3.1-A. User Interface

#### 3.1-A.1 Input
Here is the subsection where (continuing with this toy example) one might describe the *input* items of the user interface.

#### 3.1-A.1.1 Serving Size
Yet at a deeper level, the Serving Size input of the user interface. Note I did not show this in the diagram. That would be too detailed of a diagram (for our purposes).

#### 3.1-A.1.2 Number of Servings
At the same level, the Number of Servings input of the user interface.

#### 3.1-A.2 Output
Here is the subsection where (continuing with this toy example) one might describe the *output* items of the user interface.
3.1-A.2.1 Next Step Indicator
3.1-A.2.2 Oven Setting Instructions
3.1-B. Database
3.1-B.1 Indices
The database might have a child object that contains the indices.
3.1-B.2 Recipes
Then (in our toy example) the database “has a” recipe (a bunch of them)
3.1-C. Processing
etc., etc.
3.1-D. Oven Control
...and so forth and so on.

3.2 Non-Functional Requirements
Match to hierarchical numbering where appropriate. For example...
3.2-A. User Interface
n/a
3.2-B. Database
n/a
3.2-C. Processing
Ingredient substitutions should not take longer than 0.5 seconds to compute.
3.2-D. Oven Control
The temperature of the oven should be checked every 1 second.

3.3 Domain Requirements
* Include here information about co-existence with external products, conformance to external specifications, etc.

3.4 Goals
* Include here general client intentions, spirit, etc.

4. Hardware and Software Resource Requirements {~1/2 page or less}
* Include here plans to procure, estimated schedule of procurement/delivery, alternatives (risk management), etc.

5. Validation and Verification {~1 page}
5.1 Strategy
Approximately 1/2 page on your general plans, specific concerns or difficulties, timing (when will you test—throughout? one time?), etc.
5.2 Test Cases
Here list approximately 5-10 specific things you really need to test to meet your requirements. Try and make them specific and concrete (like milestones—see the class slides) not general. For example “Test A: It works” would be a bad test, while “Test A: invalid input” would be reasonable (with one sentence of specifics on what that means). Include a sentence or so with each of the 5-10 tests, explaining what it is/does.

6. Schedule (Task Budgeting) (~4 pages including diagrams)
Note that the goal here is to take a stab at budgeting time for the major phases of the project, identifying any known major tasks and milestones. The schedule will be refined as you go ahead, in particular as you begin the design specification.

6.1 Development Items (~1 page)
Major phases
Tasks and dependencies
Major milestones
Deliverables (documents, prototypes, etc.)

6.2 Schedule Diagrams
PERT chart (~1 page)
Identify the critical path
Gantt chart (~1 page)
Clearly identify the milestones

6.3 Risk Analysis and Management (~1 page or less)
6.3.1. Identified Risks
* What specific tasks/items are you concerned about?

6.3.2. Plans
* How are you addressing that concern?
Avoidance or assignment
Minimization or acceptance (w/ relevant contingency plans)

APPENDIX A
* Include appendices if necessary