COMP 3111: Software Engineering

Activity 2: System Requirements Specification

Assigned: March 22, 2016 — Value: 40% of project grade
Due: April 19, 2016, 5:00 p.m. (**REVISED** on 05/04/2016) — Activity type: Team

To build a software system that meets the client's needs and that can easily be modified as the requirements change, it is necessary to both understand clearly what the client's needs are and to document them in a clear and understandable way. The System Requirements Specification documents, in a formal way, the data and functional requirements for a software system.

> **NOTE:** ***The System Requirements Specification must include all the requirements in the problem statement as well as the proposed additional feature (see below) and any additional requirements resulting from clarification or amendment of the original requirements as specified in the System Requirements: Project Specific Q&A course web page.***

ADDITIONAL FEATURE

Propose one additional feature that will be included in the implementation of the final system. *The proposed feature will be evaluated based on the value that it adds for the users of the system.* Briefly describe the feature in at most one page in its own section in the System Requirements Specification explaining how the feature will add value to the system. Clearly describe any additional data requirements that the feature needs and include these data requirements in the domain model. The functional requirements for the feature also should be included in the use-case model. ***At most one additional feature can be proposed.***

SYSTEM REQUIREMENTS SPECIFICATION FORMAT

The System Requirements Specification should address *at least* the items listed below. **Note that there is no unique System Requirements Specification for the system!** (Welcome to the real world!)

1. Data Requirements—Domain Model

(a) Class Diagram

Construct a class diagram showing the domain model classes, their attributes and the associations or association classes that relate the classes. Use generalization, aggregation and composition relationships, as necessary. Where necessary, name all associations, show their multiplicity and include any needed role names.

Note: ***A glossary/data dictionary is not required. However, you should explain in accompanying text any class, attribute or association whose meaning is not obvious from its name.***

(b) Relational Tables

Reduce the domain model class diagram to the relational tables used in your system.

2. Functional Requirements—Use-case Model

Construct a *context diagram* showing the actors and use cases and how they are related.

For each actor provide a brief description.

For each use case provide a description that includes:

- the ***purpose of the use case***;
- the ***basic (normal) flow of events*** of the use case;
- any ***optional or variant flow of events*** of the use case ***(However, DO NOT include any exceptional flows)***;
- any ***nonfunctional requirements*** of the use case.

Your use-case descriptions should adhere to the following writing guidelines[1].

- Each use case should be *described only once*.
- Each use-case description should *start on a new page*.
- The flow of events should always start with the actor selecting to do the use case.
- The flow of events should describe both what the actor does and what the system does.
- The flow of events should describe all actions to completion without exceptions (e.g., no cancelling of the basic flow).
- The flow of events should not describe a program/procedure or how to use the user interface.
- The flow of events should not refer to user interface elements in the use-case description.
- The flow of events should not check for incorrect input. (This is an exceptional flow.)

Note: ***The use-case model should include only user-level use cases (i.e., those use cases that provide functionality of value to the end user of the system). Do not include any administration use cases or use cases that support nonfunctional requirements. In particular, there should not be a login use case!***

Note: ***You must use a modeling tool that supports domain and use-case model specification using the UML, such as Rational Software Modeler or Microsoft Visio, to construct your Domain Model class diagram and your Use-case Model use-case diagram. Hand drawn diagrams will not be graded.***

WHAT AND WHEN TO SUBMIT

Submit the following two items ***by 5:00 p.m. on Tuesday, April 19***.

1. **The System Requirements Specification** for your system, which must be
   - single-spaced,
   - minimum 10 pt. font and
   - printed *one-up* and *double-sided*.
2. The completed **Activity 2: System Requirements Specification—Individual Contribution** form, which can be downloaded from the Project Resources Module of the course web page.

WHERE TO SUBMIT

Submit the System Requirements Specification into the box on the counter in the CSE Admin office (Room 3528) by 5:00 p.m. on Tuesday, April 19.

ACTIVITY 2 GRADING

Item	Marks	% of Project Mark
Additional Feature	10	4%
Data Requirements—Domain Model	30	12%
Functional Requirements—Use-case Model	60	24%

CLARIFCATION AND AMENDMENT OF PROJECT/ACTIVITY REQUIREMENTS

Further clarification of general issues related to the course project is provided in the *Project General Q&A* module of the course's web page. Please read this information.

Furthermore, each project team can ask clarification questions of the client representative regarding the requirements stated in the problem statement or the System Requirements Specification activity. ***All questions should be submitted to the client representative either by email at 3111rep@cse.ust.hk or by Twitter at @3111rep.*** The submitted questions and their replies will be posted on the *System Requirements Q&A* web page, which can be accessed from the *Project Specific Q&A* module of the course web page. ***You should check this web page on a regular basis for clarification and amendment of project requirements.*** Any requirement added or amended on a web page in the *Project Specific Q&A* module of the course web page becomes part of the project requirements.

[1] More detailed use case description writing guidelines are provided in the course lecture notes.