****

**Ain Shams University**

**Faculty of Engineering**

**Computer and Systems Engineering Department**

**CSE 321: Software Engineering – 3rd Year CSE – 1st Semester 2016/2017**

**PROJECT REQUIREMENTS**

The following are required to be submitted by the end of your project:

* The developed code, both the source files and an executable version (or jar file if Java is adopted) of the project are needed. Also, any additional files, libraries, or resources that are needed by the project to properly run must be submitted.
* A Readme.txt file that shows how to compile the developed code and how it can be run as well as the required running environment by the project.
* testCases.txt, a file that contains all test cases used during testing phase.
* Project document, one document is needed for each group. The document must be delivered by e-mail; no hard-copies will be accepted. Project document shall contain the following sections:

**Cover Page**

This page contains university name, faculty name, and department name at the top of the page. In addition, class code and name, project name, your names, and your student IDs should be provided on the cover page.

**Abstract**: A brief description of the project.

**Table of Contents**

The table of contents of the whole document; include page number for each section/sub-section.

1. Introduction

|  |  |
| --- | --- |
| **1.1 Purpose** | The purpose of this document and its intended readership. |
| **1.2 List of Definitions** | The definitions of all used terms, acronyms, and abbreviations. |
| **1.2 Scope** | Identify the product by name and explain what the software does. |
| **1.3 Overview** | Short description of the rest of the document and how it is organized. |

1. **General Description**

|  |  |
| --- | --- |
| **2.1 Product Perspective** | The relation to other systems that may be used by the client. |
| **2.2 General Capabilities** | The main capabilities of the system. |
| **2.3 General Constraints** | Constraints, background information, and justification. |
| **2.4 User Characteristics** | The characteristics of the different user roles. |
| **2.5 Environment Description** | A description of the operational environment of the system. |
| **2.6 Assumptions and Dependencies** | The assumptions upon which the specific requirements are based. |
| **2.7 Other resources needed** | Describe all other resources that might be needed by your project such as database engines, additional libraries that you might rely on during the project …. etc. |

1. **System Requirements**

|  |  |
| --- | --- |
| **3.1 Functional Requirements** | A list of all functional requirements of the project. |
| **3.2 Non-functional Requirements** | A list of non-functional requirements of the project. |

1. Use-Case Diagram

You need to identify in this section the different roles and use cases of your project. Then, use case diagram should be developed. UML notations must be adopted for use case diagram. Use case relationships should be shown in the diagram. Roles’ relationships may also be shown in use case diagram.

1. Narrative Description/Swimlane Diagram of Use Cases

Narrative description or swimlane diagram of each identified use case should be provided in this section.

1. Data Model

If your project relies on a database, you should provide this section. This section mainly represents the data model in your system, which should be represented in EER diagram. You need to identify the different entities as well as their attributes that has to be built as a database in your system so that the rest of the system can interact with it.

1. Requirements Validation

The requirement and source traceability matrices should be at least provided in this section. All requirements should be checked for any conflicts among them or redundancies.

1. Class Model

In this section the class model for your project should be provided. Noun extraction and/or CRC cards can be used. Also, both methods can be used as noun extraction can be used as an initial step then CRC cards can be used as a refinement step. Classes, their relationships, and attributes should be identified in this section. A class diagram based on UML notations should be provided in this section.

1. State Diagram

In this section, state diagram of your system should be provided.

1. Interaction Diagram

This section should contain the interaction diagrams (either sequence diagrams or collaboration diagrams could be adopted) for each use case you have provided. Proper descriptions and/or comments on the provided diagrams should also be shown in this section.

1. Detailed Class Diagram

Updated class diagram that contains classes, their relationships, attributes, and operations should be provided in this section. Attributes’ types, operations’ arguments, arguments’ types, and return types of each operation must be provided. If you use user-defined data types, you need to provide the detailed description of these types in terms of the built-in types of the language that you use for implementing the project. You need to provide proper description of the class diagram in which you can show all assumptions used, any modifications done on the class diagram and the justifications of the modifications, and any other details that you cannot provide on the diagram itself.

1. Data Model Design

If you rely on a database on your project, you need to provide the detailed schema used by your data model. That mainly shows the tables in your database, their attributes, and the keys in each table.

1. User Interface Design

In this section, you need to describe the UI styles adopted in your project, the user actions that cause the interface state to change, and how the user may interrupt the state of the system. In addition, any other decisions related to user interface (e.g., accessibilities for who has special needs, internationalization … etc.) should be shown clearly in this section.

1. Client-Object Relation Diagram

This section contains the client-object relation diagram for your project and its proper description. All objects that must be initiated by the main application object (or main function if C++ is adopted) should be clearly stated in this section.

1. Detailed Design

Use Program Description Language (PDL) to detail major operations defined in the classes of your project. You need to provide reasonable details in the PDL of each operation so that the mapping from PDL to implementation will be straight forward.

1. Testing

Detailed testing process that you adopt in the project should be described in this section. It must cover class-level testing, integration testing, and system testing. It should detail all test cases used during testing process associated with test results that should be properly described and commented.

1. Estimated Project Cost

An analysis of the estimated cost of the project based on COCOMO-II model should be provided in this section. All assumption used in the estimation process should be clearly described.

1. User Guide

In this section, you should provide the details of how to use your software. Screenshots and sample output of your software must be provided and commented. Step-by-step description of how to use your software should also be provided in this section.

Appendices

This section might be needed if you have extra information that helps clarifying certain issues in your document that you don’t need to distract the reader by adding them inside the text.

Notes:

* Use UML notations (when applicable) for the diagrams you provide in your document.
* Use a professional drawing tool (e.g., MS-VISIO) to draw the diagrams in your document.
* Use consistent document format (font sizes, titles, subtitles, captions, paragraph formatting … etc.). Recommended font sizes are: main title 14pt, subtitles 12pt, main text 12pt, and captions 10 pt. Recommended font type is “Arial Black” for titles and subtitles, and “Arial” for all other texts. Recommended spaces before and after paragraphs are 12pt before and 6pt after each paragraph, and 1.5 spacing is highly recommended. Justified paragraphs from both sides are also recommended.
* Figures and tables must be centered in the pages, and they should be numbered separately. Each figure/table must have a caption that appears below the figure/table.
* Pages must be numbered consistently except the cover page.
* All reports must be written in English, always avoid typos and grammatical errors.