**Final Group Term Project**

Create a Project Plan for your Group Term Project. The Project Plan should include the following sections. The Project Plan should be submitted as a Microsoft Word document (and all necessary additions if applicable). The text should be left-justified and in font Calibri Body 11. Line spacing should be ‘0’. Specify page numbers in form **Page X of Y** at the bottom of each page. Use the MS-Word Styles -> Headings for the title of each section such that enabling ‘View’ -> ‘Navigation Pane’ will result in a navigable Navigation Pane. Please run spell-checker on the entire document before submitting it, after selecting in MS-Word ‘Review’ -> ‘Language’ -> ‘Set Proofing Language…’ -> ‘English (United States)’.

## 1. Title, Names of Group Members, Table of Contents generated via MS-Word TOC generation tool.

## 2. Requirements Definition Section

The Requirements Definition Section should include at this stage a tentative Group of related Requirements that will cover 4-5 (four to five) meaningful Functional segments (Use Cases) of your project. The requirements should follow the format as the sample requirements we saw in class. Each requirement must have a Unique Identifier that is unique across the entire project and is traceable to and entry from a Use Case. Each Requirements Group should contain at least 5 steps, or actions.

- Copy each Requirements Group from the Requirements Section to within the Use Case, preceding the tabular/textual Use Case table.

## 3. Use Cases

a) The Use Cases section should include 4-5 (four to five) meaningful Use Cases in tabular/text format **and their equivalents in UML** (for Cameo/Magic-Draw see: https://docs.nomagic.com/display/MD2022xR1/Creating+diagrams). The Use Case IDs and the internal steps numbers should create unique identifiers across the entire project (e.g. “UC-1”, “UC-2”, “UC-3”, should be unique across the entire project). The Use Cases should be traceable to a specific Requirement or group of Requirements from the Requirements Definition Section.

b) Please number the Steps within each Use Case uniquely across the entire document, i.e. if all the Steps in Use Case UC-1 are numbered 1.1, 1.2, …, 1.n, then number the Steps in Use Case UC-2 to be 2.1, 2.2, …, 2.n, etc.

c) The “Use Case Name” field in the tabular/textual should contain exactly the same text (with the same spelling and capitalization) as the Label in the Oval of the UML representation of the Use Case.

d) The “Use Case Name” field in the tabular/textual should contain exactly the same text (with the same spelling and capitalization) as the Title at the top of the Use Case, and should also match to-the-letter the section title of its corresponding Requirement Group.

e) **The tabular/textual Use Case should abide by all the syntactic and semantic rules shown in class.**

f) Keep each Use Case simple with one Use Case Oval and two Actors.

g) Among the Use Cases please provide at least one (or one additional) Use Case that is a back-end Use Case, i.e. that doesn’t involve only interaction with the end-User, but rather models internal interactions within the system.

## 4. Process Model

The Process Model section should include:

a) A **Data Flow Diagram** (DFD) for each **Use Case. Each element in the DFD must be traceable and match semantically and syntactically the entries in the corresponding columns of its Use Case**.

b) Please arrange the DFDs to appear next to the Use Cases they represent, rather than having a list of all Use Cases and only then an entire list of DFDs. So you should have in one section Use Case UC-1 followed by DFD-1, then in the second section Use Case UC-2 followed by DFD-2, and so on.

c) Within each DFD please attempt to break the Processes into at least 2 separate Processes per DFD.

d) Within your overall DFDs (i.e. throughout the entire project) please attempt to come up with more than one Data Store, i.e. more than one default Data Store which is the generic system Data Store. Actually, you can come up either with an additional Data Store or with an additional External Entity.

**e)** Below the DFD, please provide a '**DFD Description Section**' that will include your choice of **5 consecutive steps** from the DFD, i.e. 5 consecutive Data Flows from your DFD. The Data Flows can span one or more Processes if necessary.

**Each Data Flow in the DFD has to be described in its own full sentence in the 'DFD Description Section' (Subject, Verb, Object, Source, Destination), using the correct qualifiers, numbering, and designators (EE** <label>**, IDF** <number> <label>**, ODF** <number> <label>**, P** <number>**, DS Dx** <label>)

## 5. Class Diagram

a) The Static Model section should include an OO UML Class Diagram. The semantic and syntactic rules should be enforced as shown in class and as shown in class and in sample case "ATM User".

b) Each diagram (or partial diagram) should be accompanied by a textual Description Section of the main elements in the diagram. This textual narrative description should explain the Classes, Associations, Hierarchies (if any), main Attributes and main Methods/Operations (as well as Traceability, and Multiplicity sections).

c) The Class Diagrams should match and be consistent with all the DFDs in the project:

   i. Class names should be consistent with DFD External Entities, Data Stores, and possibly to some of the Data Flows, and should be traceable to the corresponding columns in the Use Case.

   ii. Attributes should be consistent with DFD Data Flows (as much as possible), and should be traceable to the corresponding columns in the Use Case.

   iii. Operations/Methods should be consistent with DFD Processes and Data Flows (as much as possible), and should be traceable to the corresponding columns in the Use Case.

   iv. Regarding operations/methods: As a first step you should use your existing DFD Processes as the basis of the Operations/Methods that you'll add to your Class Diagram. You should have one method for each Process in your DFD with the appropriate name that will match the Process name in the DFD but changed such that it will conform with our UML standard naming for methods. In the end, you should have at least 2 meaningful methods for each Class (excluding setter and getter methods which should be omitted). If you have enough Processes in your DFD to satisfy the requirement of 2 meaningful methods for each Class just go ahead and use them. Otherwise, add additional methods that would be invoked in addition to the methods corresponding to your DFD Processes.

The Class Diagram may contain additional Classes and Attributes that are not reflected in the DFDs since a Class Diagram usually adds details that are not always available in the DFDs, so as much as you are encouraged to match elements in the Class Diagram to the DFDs, you can have some elements in the Class Diagram that do not have corresponding elements in the DFDs.

## 6. Dynamic Interaction Model – Sequence Diagram

The Dynamic Interaction Modelling section should include a Sequence Diagram corresponding to each of the Use Cases (for each Use Case provide one Sequence Diagram). Elements in the Sequence Diagram should match and be consistent with Class Diagrams, and Data Flow Diagrams. The main methods in the Class Diagram should be shown in the Sequence Diagram.

Use all the syntactic and consistency rules, as discussed during the semester.

## 4. Contribution Table

**A tabular description** of how each member of the group has contributed to this iteration with columns Date, Student Name, Use Case Name/Number, and Contribution Description.

– and updated for each increment, i.e. separate entries for 'Requirement' works, 'Use Cases' work, 'DFDs' work, etc.

The UML version of the Use Case and other models should be created with Cameo Enterprise Architecture – MagicDraw, or any other graphical drawing software or drawing tool – but should be **consistent** across the Project.