CSC 400 – Summer 2018

Software Specification and Design Document Template

(Some sections are adapted from Ofer Faigon and from UML Distilled by Martin Fowler)

Draft of Design Document due by 8am on Monday 06.25.

For the draft, focus on sections 1, 2, 3, 6 and 7.

1. Introduction

Background

A well-written background should not cover more than a single page. As it is the opening section of your design document, the background must refer to the following questions:

- What the new system is
- An overview of important features
- Its advantages over comparable systems
- Who the potential users are, and how they will benefit from it

2. Architecture

Should contain details such as:

- The type of the system (distributed, standalone, client-server, etc.)
- What platform(s) the system will run on (e.g. mobile, cloud).
- The major inputs and outputs of the system.
- What user interfaces the system will have and in what form (web, Windows GUI, etc.)

A block diagram of the modules/components and the relationships between them can be very useful here. Try to point out the dynamic aspects even though this view is mostly static: include arrows to indicate flow of data and/or control.

3. Use Cases

To think about what a use case is, we'll first define what a scenario is. A scenario is a **sequence of steps describing an interaction between a user and a system**. For instance, if we have a web-based store, we might have a Buy a Product Scenario that would say this:

> *The customer browses the catalog and adds desired items to the shopping basket. When the customer wishes to pay, the customer describes the shipping and credit card information and confirms the sale. The system checks the authorization on the credit card and confirms the sale both immediately and with a follow-up email.*

The scenario is one thing that can happen. But the credit card authorization might fail. That would be a separate scenario. **A use case is a set of scenarios tied together by a common user goal.**

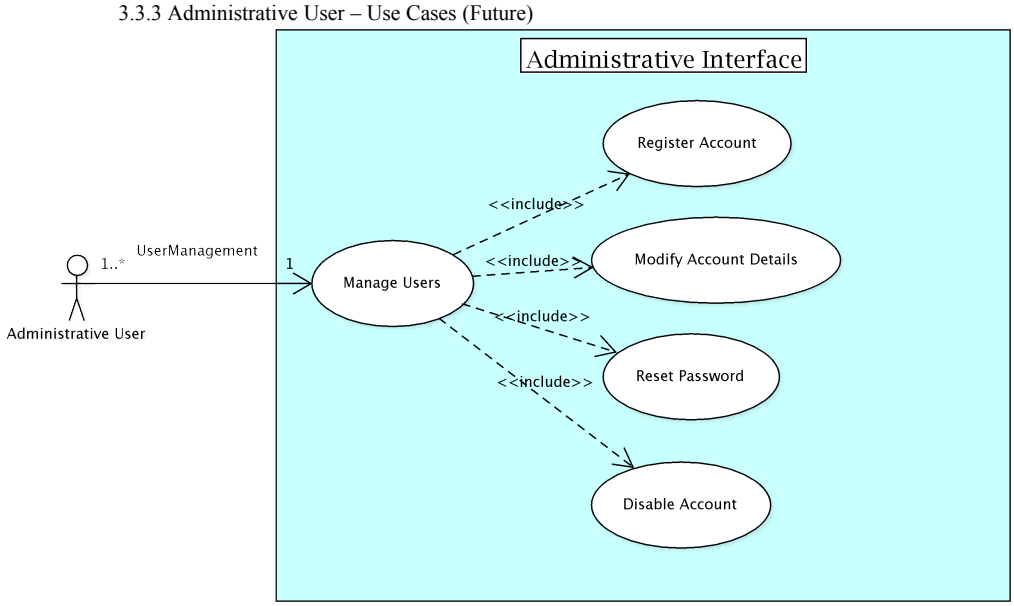
This section should contain the following:

- Use cases diagrams that define the interactions between the user roles and the system

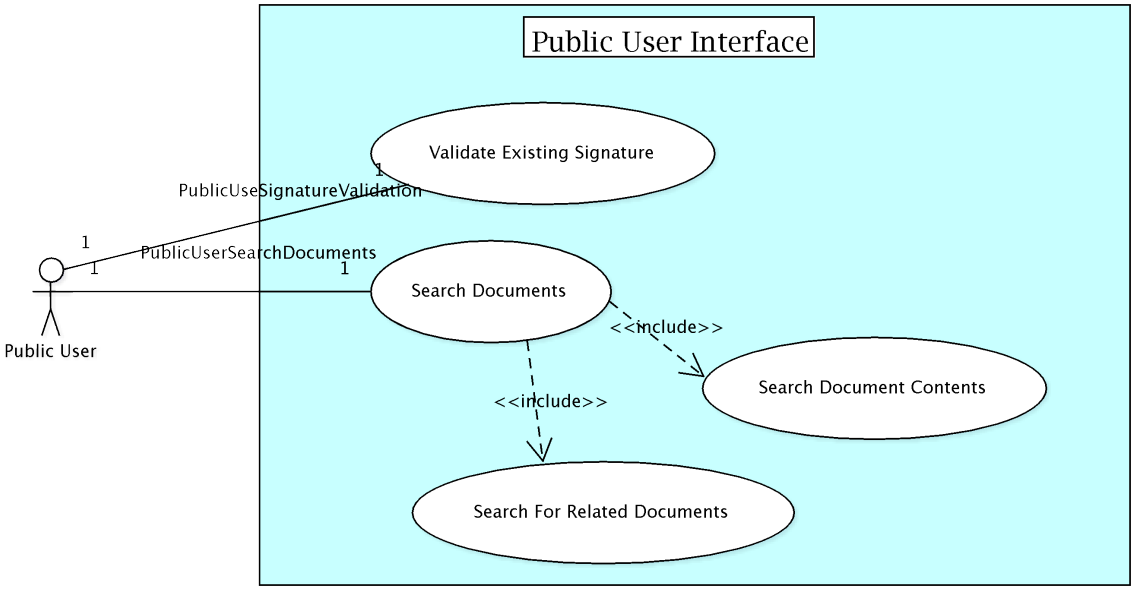
- Each use case diagram **must be accompanied by one or more scenarios expressed as a numbered sequence and alternatives.**

Below are examples of use case diagrams:

3.3.3 Administrative User – Use Cases (Future)



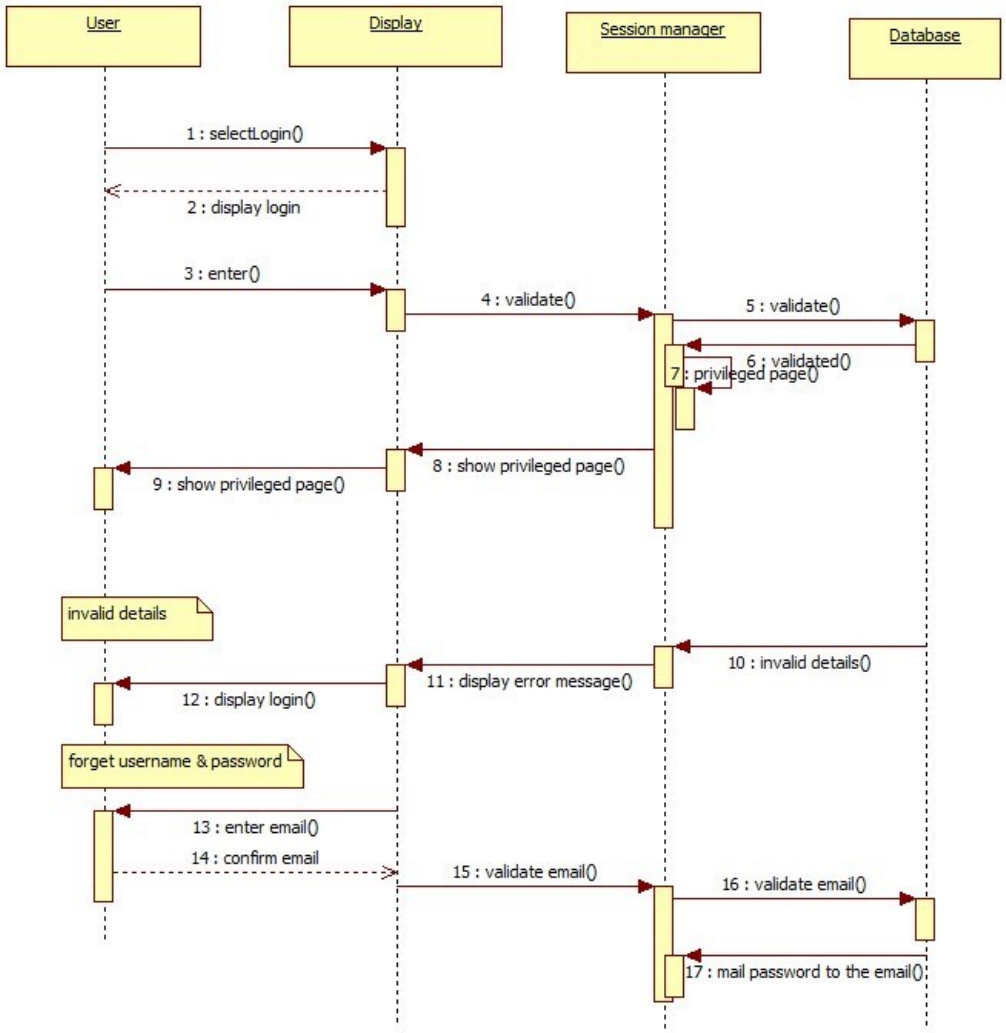
3.3.4 Public User – Use Cases (Future)



4. Sequence Diagrams

Sequence Diagrams are UML diagrams that show how system components/objects interact with each other and in what order. For each scenario create a sequence diagram. Below is an example of a sequence diagram.

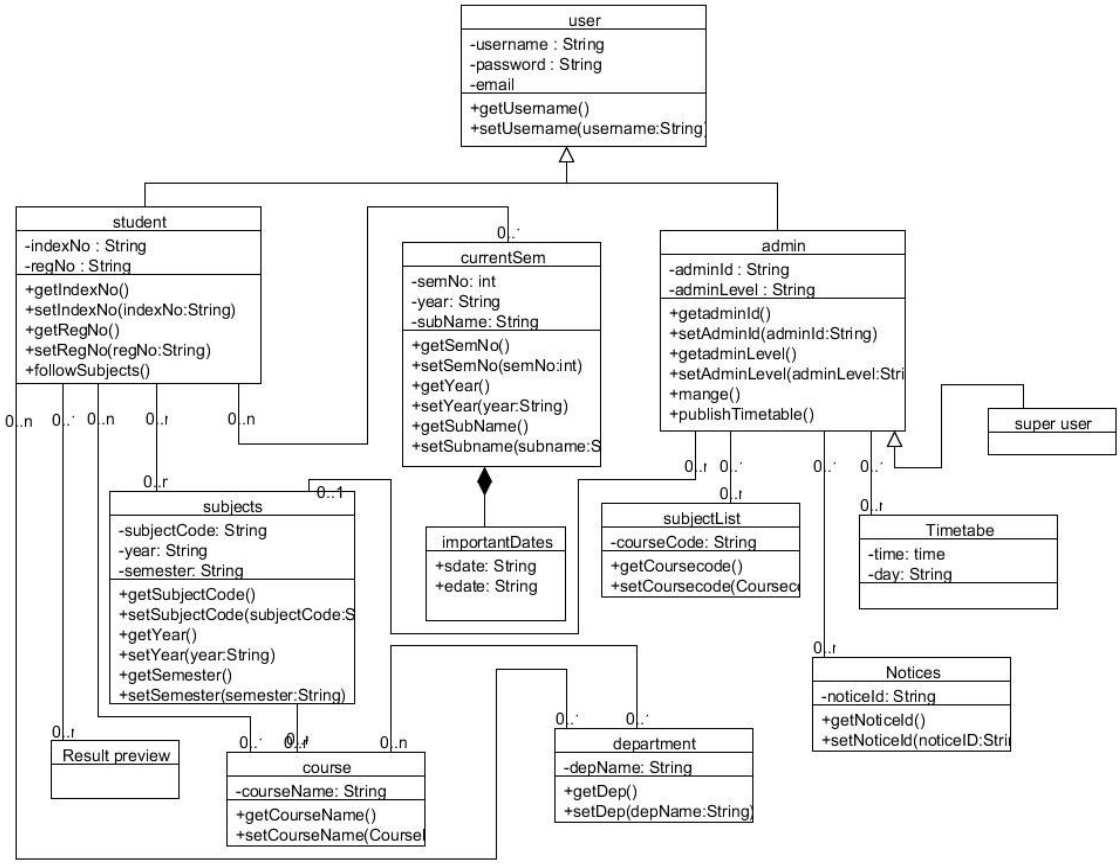
3.3.1 Sequence diagram 1: Login into the system



5. Structural Design

The following UML diagram should be included:

- **Class Diagram**: A UML diagram that shows the classes, attributes and the relationships between objects.



6. Data

This section describes the persistent data, and/or any other data important enough to justify a separate section. Such a section will not be needed when there isn't a database.

Schema

Should give a list of tables and columns in each table, a description of each of the following - data type, size, number of records, what data it keeps, which parts of the software read it and why, which parts of the software write it and when.

7. User Interface Design

Should include mockups of all user interfaces in your system. For each use case, there should be an interface. There are numerous tools for creating mockups of interfaces.