Joseph Vitale
Fall 2019 CS4320 Software Engineering
Assignment 2: Requirements Analysis

Step 0: Make sure you understand the problem. You can ask questions in the GitHub channel #requirements in the SWES17 Team.

Step 1: Identify the different types of Users of the software system.

The users for the system will be students, TA's, and instructors.

Step 2: For each identified User, identify the Activities they will perform with the software. (These are User Requirements.)

Students will:

1. Log in to the system.
2. Select one of their currently enrolled courses.
3. Select assignment.
4. Submit assignment.

TA's will:

1. Log in.
2. Select a course which they are currently assisting with.
3. Select the assignment they'd like to collect.
4. View all student submissions.
5. Download submissions.

Instructors will:

1. Log in.
2. View their list of courses for this semester.
3. Select course to manage.
4. Manage (view, edit, add, remove) a course's TA's
5. Manage the course's sections and students.
6. Manage assignments.

Step 3: For each identified Activity, identify…

- **relevant data within the system. Data entities and attributes may be simply listed or you may construct a data model if it helps.**
- **constraints (non-functional) on the activity or the resultant state of the system**

Student:

1. Log in: There should be a table for **users** with attributes id (integer), user type i.e. student, TA, or instructor (char), name (string), and password (string).
   Logging in should be quick and secure.
2. Select course: There should be a table for **courses** with attributes course id (integer) name (string), and instructor id (foreign key, integer) and an **enrollment** table to keep track of student's courses with attributes student id (foreign key, integer) and course id (foreign key, integer) and section (integer).
   Course selection page should be user-friendly, organized, and visually appealing.
3. Select assignment: There should be a table for **assignments** with attributes assignment id (integer), course id (foreign key, integer), name (string), and due date (date time).
   Assignments and their details should be listed clearly and organized in an intuitive way.
4. Submit assignment: There should be a table for **submissions** with attributes submission student id (foreign key, integer), assignment id (foreign key, integer), course id (foreign key, integer), filename (string), and server file path (string).
   Submitting assignments should be easy and fast. File submission should have sufficient exception handling.

TA:

1. Log in: TA information will be entered in the user table described previously.
   Logging in should be quick and secure.
2. View courses: Along with the courses table previously describes, there should be an **course-assisted-by** table with a TA id (foreign key, integer) and course id (foreign key, integer).
   Course selection page should be user-friendly, organized, and visually appealing.
3. Select assignment: This action will use the assignment table previously described.
   Assignments and their details should be listed clearly and organized in an intuitive way.
4. View student submissions: This action will use the submissions table previously described.
   Viewing submissions should be made non-tedious, such as featuring a preview feature. Submissions should be presented in an organized, logical manner.
5. Download submissions: This action will use the submissions table previously described.
   Downloading submissions should be quick and easy.

Instructor:

1. Log in: Instructor information will be entered in the user table.
   Logging in should be quick and secure.
2. View courses: This action will use the course table, which has the instructor attribute included.
   Course selection page should be user-friendly, organized, and visually appealing.
3. Select course: This action will use the course table.
   Course selection page should be user-friendly, organized, and visually appealing.
4. Manage course's TA's: This action will use the course table.
   Course management page should user-friendly, organized, intuitive, and visually appealing.
5. Manage course's sections and students: This action will use the enrollment table.
   Course management page should user-friendly, organized, intuitive, and visually appealing.
6. Manage assignments: This action will use the assignment table.
   Managing assignments should be user-friendly and intuitive.

Step 4: Identify System constraints and requirements
i.e. hardware and necessary components

This application will need a web interface with pages for login, course selection, course and assignment management, assignment submission and assignment download.
It will need to be hosted on a server.
There will need to be adequate storage on the server to temporarily store student submissions until they are all graded.
There will need to be a database and adequate storage for entries in the tables for users, courses, enrollment, course-assisted-by, assignments, and submissions.
The web application should support multiple users using the system simultaneously.
Users will need internet access to use application.