Requirements Engineering
Homework Assignment #3
Due: April 4, 2018

Assignment Topics:

Graphical-based Requirements Specifications

Assignment Description:

The following description outlines current procedures for registration and payment within a continuing education department at a local university. The information provided captures some of the department personnel's current procedures to be automated within a planned computer-based registration and accounts receivable system to support administrative personnel in the department.

Continuing education for professionals is a must to keep abreast of the current but ever changing technology. At the local university, professional students register each August for the Fall semester, each December for the Spring semester and each May for the Summer sessions. To take a class, students must live in Texas, work in Texas, have a high school diploma, and pass an entrance exam with a score greater than 75. When the given registration period begins at the beginning of the month (August, December, or May), students are given 2 weeks to register and must pay for their tuition during the registration period; they may pay with cash, credit card or check. No financial aid is provided. To register for a class, the student must determine the course to take, make sure the course schedules do not conflict with work, receive approval from work, and receive acceptance from the continuing education admissions office. Most people take continuing education courses to acquire or update their knowledge in a given area. A degree plan is not relevant.

Once registration is approved by an authorized department supervisor, payment is requested. Courses cost $1000 per course. If payment is not received by the end of the registration period, the student's registration is cancelled. A complete payment record will indicate the student name, payment amount, type of payment (check or cash), payment receipt confirmation number, and confirmation date. The student's registration record will specify the student's course schedule and semester enrolled. A course schedule includes the name, date, time and room number of each course the student is taking. Students are not allowed to take more than 3 courses in a given semester.

On the last day of enrollment, department personnel access the current enrollment for each class to determine the amount of income for the upcoming semester. If income is greater than cost, all courses will be offered. If income is less than costs, some courses will be cancelled. Once the registration period is over, the primary duties for the registration and accounts receivable system are complete.

Since the registration time is brief and many students are seeking to take courses in the department, the system will be used by a number of department personnel during this peak usage time. During *this peak time, users expect no more than 2 seconds per transaction for all requests. Federal regulations require payment data to be encrypted when transmitted or stored. At their computer screens, they would like to be able to readily access either the registration or payment service by clicking on a respective icon. Additional departmental personnel in student counseling offices and alumni relations may wish to access the data base of student records in the future. The department does not have a great deal of money at this time and intends to implement a minimal core set of functionality with the intention to extend the system with many additional services in the future. New computers and networking system are investments they plan to make before the automated system arrives. They are planning to buy servers with 2 GHz processors, 2 GB memory, and support for gigabit networks. All the employees have asked for 21" LCD monitors. They have selected an Oracle as the DB vendor.*

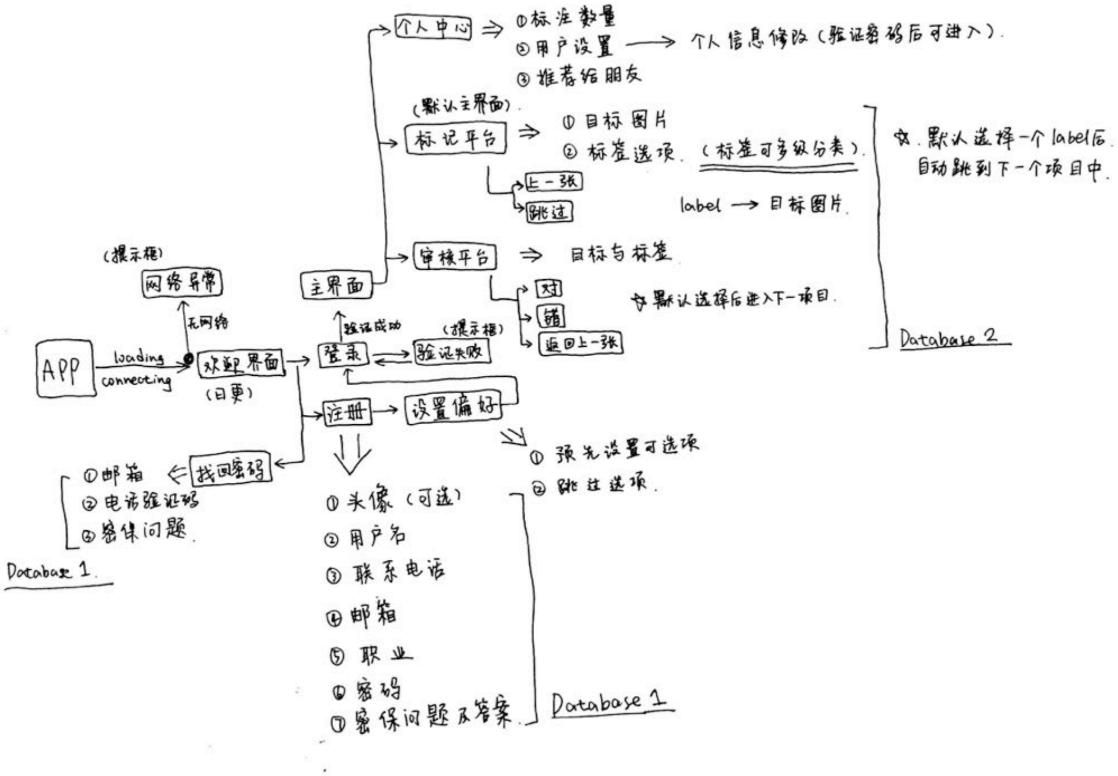
For the following questions, consider yourself the requirements engineer playing the "modeler" role (i.e., another requirement engineer conducted the acquisition and you have been tasked with modeling the results).

1. **Identify three *Installation Requirements* and for each describe how stakeholder expectations have changed over time.**
    - A mobile-phone base on Android Operating System.
    - At least Android 6.0.
    - At least 2G CPU and 1G RAM.
    - With a camera better than 8 millions pixels.

2. **Identify three *Non-Functional Requirements* (NFRs) and for each suggest scope and evaluation criteria.**
    - A flat interface.
    - Convenient operation process.
    - A database server.

3. **Draw one *Operational Reference Model* (ORM) for the process occurring during the registration period.**
    - This part is the same picture in next step. I will show the whole picture later.

4. **Draw a *Task Decomposition* (TD) diagram for each task in your ORM and any additional tasks that can be decomposed that are not represented in your ORM.**



- *Database1* is about users' information and *Database2* is about the information of the picture which we need to collect and annotate the data.

5. **Create at least one *Task Hierarchy* (TH) to describe how a task can be performed more than one way.**
   - A picture in database2 is showed to user in the main interface and get a label from this user, we will get this data in the database. And then this picture will show to another user and get a label, also save this data to the same database. The most important column is the picture and we need to update different labels to this picture, which is aimed to get the more credible labels about this picture.

6. **Select two leaf tasks in the Task Decomposition models to specify in a *Task Template* (TT).**
   - As for the same picture, we will show it to more than one user to give a label to it. And we will choose the more credible one according to the users' history and their favorite parts.
   - We have a part to remark it, we will send the picture and its label to another user to confirm it, just 2 chooses-right or wrong, if one label have confirmed by another user, we need to believe this and if this label been denied by others, we will remark it.

7. **Create one *Temporal Sequence* (TS) reflecting the process to *register for a class* with no less than three swimlanes.**
   - For users, they will mark different pictures with time flow.
   - For different pictures, they will be sent to different users to give label to it.

- For users' credits, if his/her labels be denied many times by others, we will reduce the weight on him/her.

8. **Identify additional information that you would like to acquire based on missing content in two different types of models (i.e., evaluate for completeness).**
    - From TT and TS, I would like to make sure which user is "Good" and which is "bad", I need to analysis their data by their histories and give them scores which influence his/her weight when I calculate the best label about the picture from many answers.

General Comments:

- I fully expect that everyone's models will look slightly different – there is no single correct set of models, even though each set may capture roughly the same information. Differences will be based somewhat on interpretation but even more on modeling choices.
- For this assignment, I'm telling you what models to create, so there may be content in the description that does not get represented. For example, there may be many possible ERDs, but I'm only requiring two. This is in contrast to the second assignment where I expected you to capture everything in the description.
- You may use whatever diagramming application(s) you wish as long as you are relatively faithful to the notation (syntax) for each model. Example applications include UML tools (e.g., Enterprise Architect), Visio, and PowerPoint.
- The task templates can be created in Word or Excel. However, be sure to include every field, whether populated or not.
- Task names (e.g., in your TD and ORM) begin with verbs (i.e., action) and should not include the performer. For example, "Add Course" is an acceptable task name, but "Student Adds Course" and "Course Added" are not. Do not want to overload model with more than we wish to convey, and we may realize that a more general resource (parent) is capable of performing a given task.
- Note that the order in a task decomposition doesn't matter *and* the decomposition does not imply *all* subtasks are necessary to accomplish the parent task (determined by pre/post conditions).
- Events should be specified in past tense (e.g., "Tuition paid").
- All models should have a traceability annotation block. To assist with your model annotations, you can refer to the following information:

  For the "RM Number" field in the annotation, you can use the following id scheme:

  Perspective_ModelTypeAbbreviation_#

  …where ModelTypeAbbreviations include the following:
  OR – Operational Reference Model
  TD – Task Decomposition
  TH – Task Hierarchy
  TT – Task Template
  ER – Entity-Relationship Diagram
  TS – Temporal Sequence

  For the "RA Source" field in the annotation, you can use "Admin_Overview_001" since the information is provided from the Admin perspective.

Grading Guidelines:

- *Model Syntax/Notation:*
  - Models correctly follow respective syntax/notation. For example, relations (lines) in entity-relationship diagrams are properly labeled and include cardinality.

- *Traceability:*
  - Traceability annotation block is present and correctly filled out.

- *Model Semantics / Purpose:*
  - A given model is used only for its intended purpose. For example, a task decomposition describes tasks and their constituent subtasks. It is not meant to describe anything with regard to task execution order, task hierarchy (Is-A), task performer, etc.

- *Layout:*
  - A model on a given page is not too cluttered. For example, if a task decomposition has many levels, consider giving one or more sub-trees their own separate models. (Note: If you opt to do this, give the sub-tree model a new unique "RM Number.")

- *Accuracy and Completeness:*
  - The complete set of models accurately and completely captures the information in the requirements description *to the best of their combined ability* (i.e., none of the models are intended to describe non-functional requirements, so this information will not be present in a model).

- *General Presentation Quality*