

## Assignment 2

In4Mtx 113

The objective of this assignment is demonstration of your understanding of building a comprehensive requirements document, accompanied by supporting analysis.

### Process

1. Pick a team and email Leyna your team roster no later than Feb 24. Team sizes must be 4 (please email Leyna as soon as possible if you need a team). You can stick with the same team you had for Assignment 1, or you can form an entirely new team. There is a lot to do for this assignment and working in a four-person team will make it easier.
2. Select your desired systems (at least 3) from the “System Choices” list at the end of this document. Email Leyna your rank-ordered preferences no later than Feb 26—one email per team. You are not guaranteed your first system choice — it’s on a first come first basis, and only 2 teams will be assigned to a given system. Leyna will let you know your team’s system by 8:00 pm on Feb 26.
3. Create a RD for your system, following the instructions below for Sections 1, 2, and, optionally, 3 (bonus).
4. Submit the assignment according to the submission instructions below.

**Section I: Create a RD with the following structure and outline. (80 points total.)** You will be graded for the RD’s organization, clarity, consistency, and (lack of) ambiguity. The document should clearly reflect that you have attempted to correctly apply the techniques found in chapters 1-8. This does not mean that you have to apply all the techniques, or use all the notations. On the contrary, your document should demonstrate that you can identify appropriate techniques to use for the task at hand (subject to the specific requirements below).

I. Scope and System Overview: clearly describe what your system is. Include a context diagram, annotated with appropriate textual content. Include other diagrams as appropriate (such as a dataflow diagram). Note that the purpose of this section is *introduction*: detailed requirements and additional diagrams come below.

II. Goals and Goal Refinement Diagrams: describe the goals for the system and refine the top-level goals at least three levels down or until a specific software *requirement* (“shall statement”) or *expectation* (“goal for the environment”) is identified. Make sure the agent(s) responsible for each goal are identified. Assign a unique ID to each requirement or expectation.

Annotate your goals to indicate whether they are behavioral or soft, and whether the behavioral goals are achieve, maintain, or avoid. If you do not identify any “avoid goals”, explain why your system does not contain any bad conditions that must be avoided.

III. Specific Requirements

(a) Using “shall statements” describe additional specific requirements of your system not already captured in step II above. Assign a unique ID to each requirement, as before. (You’ll use the id’s in part (c) and also in Part 2 below.)

(b) Augment the requirements statements from Step II and (a) above with appropriate diagrams:

State Transition Diagrams  
Event Sequence Diagrams (and Use Cases as necessary)  
Dataflow Diagrams

(c) Correlate the requirements from part (a) to the diagrams, or portions thereof, as necessary. Use the requirements IDs to help with this.

IV. Fault Trees: Identify two risks associated with the system and provide a fault tree analysis for each ("risk analysis" trees as described in Section 3.2 of the textbook). The risk trees should terminate either at primitive nodes (nodes for which further refinement makes no sense) or at level 5 or 6.

V. Glossary of Terms: include a fully defined glossary for the key terms used in your RD. Italicize throughout the RD each term that is included in the glossary (so if the term "user" is defined in the glossary and used 16 times throughout your document, all 16 uses must be italicized).

## **Section 2: Verification (20 points)**

Explain clearly how satisfaction of **each** requirement can be determined. You might consider putting a statement after each requirement explaining how this requirement is expected to be verified. This could involve specific test cases, or demonstration, or analysis, or inspection, or simulations. *Or* you might want to consider including a matrix at the end that cross references each requirement (by id) with specific verification means.

Think of it this way: suppose you are the poor tester stuck with testing this system: what information would you need to make verifying these requirements easier?

## **Part 3: Bonus Points (20 points maximum. 15 points for Bonus # 1, 5 points for Bonus #2)**

1. If you use formal notation for system behaviors and system properties (see Section 4.4 of Chapter 4 and Chapters 16-18 for advanced reading topics).
2. If you turn in your assignment on Monday March 8 instead of Thursday March 11.

## **Submission instructions:**

Due on March 11 at 9:00 am via dropbox on EEE. Only one document per team, do not turn in multiple documents. Do not have more than one team member turn in the assignment.

No late submissions, timestamps will be checked. Late submissions will result in a 0.

Assignment 2 is worth 30% of your grade.

## System Choices

1. Banking System: a system that allows retail users to perform their basic banking activities: make payments, transfer funds, see account balances, see printed checks, summaries of spending. Users should not be able to overdraw on their balances, or make payments if they do not have enough funds in their accounts.
2. University Registration System: a system that allows university students to register for courses, waitlist for a class, see their class schedule, pay their tuition, allow external payments (i.e. from a student's parent). A student should not be able to register for a class that is full, and students should be able to see where they stand on a waiting list. Students should also be able to see their payment history.
3. Airline Reservation System: a system that allows a user to book flights, alert users to alternative airports, perform cost comparisons for flights, and allow the user to select one-way reservations, non-stop flights
4. HR System: allow employees to register for benefits, request paystub information, sign up for 401K, do direct deposits, see their vacation bank, update personal info, and allocate withholdings. Employees should not be able to see other employees' information.
5. Auctioning System (much like eBay): users can create accounts, take payments, make payments, auction for items, sell items, and be able to browse items and their descriptions. One expectation is that a standard will be placed on how an item appears for a user to browse its specs and make a decision.
6. Train Reservation System: a system that allows a user to book train tickets, alerts user to alternative train stops, looks for the shortest way to get from Point A to Point B and recommends this route to users, layout of a map to help a user make an informed decision about the best route for them
7. Pick your own and let Leyna know no later than Friday Feb 26. *You must obtain her approval before proceeding.*