

Software requirements master rubric

The rubric below collects all the grading criteria from every CSSE371 project milestone and groups them approximately as they were originally presented to CSSE371 students. None of these items are new to students; they've all seen these very criteria and know exactly what is expected of them by each checkbox.

In 371, individual grade items on the list were generally graded using a **pass/fail basis**, with the bar being about **80% correctness for each checkbox**. Grade items were arranged in the rubric in increasing order of sophistication akin to the style of "mastery" assessment.

To earn a given letter grade, students should demonstrate proficiency in the work presented under that letter grade and all lower letter grades.

D

- ☐ Planned Questions section listing the specific questions you intend to ask
- ☐ Transcript that accurately captures actual interview
- ☐ Needs section
 - ☐ Needs are linked to the Transcript Entries supporting them.
 - ☐ Needs capture major needs of stakeholder.
 - ☐ Each Need is appropriately worded for its section. (per Leffingwell and Widrig's Managing Software Requirements: A Use Case Approach)
- ☐ Features section
 - ☐ Features are linked to the Needs they support.
 - ☐ Each Feature is appropriately worded for its section.
 - ☐ Features logically follow from major needs of stakeholder.

C

- ☐ Use Case Diagram section
 - ☐ Formatted appropriately
 - ☐ Every stakeholder discovered in the system is represented in the diagram.
 - ☐ Each stakeholder has named use cases stemming from described Features.
 - ☐ Shared use cases are linked to their shared stakeholders.
 - ☐ You do not have to write full use cases for this section; only the diagram.
- ☐ List of emphasized Design Principles for each critical Feature
 - ☐ See Preece et al's Interaction Design Ch 1 and 3
 - ☐ Two design principles and include why each is important for the current Feature
 - ☐ Two cognitive paradigms and include why each is important for the current Feature
- ☐ Transcript of emails discussing Use Cases and Prototypes.
- ☐ Nonfunctional Requirements section
 - ☐ NFRs stated in shall format.
 - ☐ NFRs do not violate the following attributes:
 - Feasibility
 - Validity
 - Unambiguous

- Complete
 - Verifiable (Testable)
 - Modifiable
 - Consistent
 - Traceable
- ☐ NFR section may contain some, but not all, relevant NFRs.
- ☐ Prioritized Features section
 - ☐ Use techniques described in the course to help the client prioritize the features.
 - ☐ Put the highest priority items at the top of the list
 - ☐ Include a description of each feature
 - ☐ All features prioritized

B

- ☐ Use Cases meet the following criteria:
 - ☐ Follow the use case format prescribed by Leffingwell, Jacobson, etc.
 - ☐ Don't make these mistakes: <http://www.drdoobs.com/top-ten-use-case-mistakes/184414701>
 - ☐ Every use case exists in the Use Case Diagram.
 - ☐ Every use case is explicitly hyperlinked back to a feature.
 - ☐ Every actor involved in the use case is identified.
 - ☐ An appropriate description for each use case is provided.
 - ☐ Use case has an appropriate Name.
 - ☐ Use case provides appropriate basic flows
 - ☐ Use case provides appropriate alternate flows.
 - ☐ Use case provides a pre and post condition as necessary.
 - ☐ Basic and Alternate flows are appropriately worded.
 - ☐ Basic and Alternate flows are linked by step numbers.
- ☐ Balsamiq Prototypes for every Use Case:
 - ☐ Prototypes show every screen/system response
 - ☐ Prototype buttons are functional - linked using Balsamiq's linking tools.
 - ☐ For each screen, include the corresponding use case step as a Note in the screen.
- ☐ Release Schedule
 - ☐ The number of releases is up to you
 - ☐ The goal is to build your client's confidence in your company while also providing a functional product for a smaller investment. The eventual goal is that your client will ask you to complete each future release as you build their confidence.
 - ☐ All features are scheduled.
- ☐ Justification explaining:
 - ☐ Why/how you split your releases
 - ☐ Why/How each release provides benefit to your client
 - ☐ Why/How each release provides enough functionality to be a viable product
 - ☐ Justification is consistent with the client's needs.
- ☐ Usability Test Plan includes all elements:
 - ☐ Goals
 - ☐ Consent form
 - ☐ Script
 - ☐ Pre/post-test questionnaire
- ☐ Business Process Diagram capturing the stakeholders' perspectives on the business process.
 - ☐ Diagram shows the strategic view of the business process.
 - ☐ Diagram distinguishes Human, Message, and Service tasks.
 - ☐ Notation and labels follow Camunda's guidelines.
 - ☐ Diagram syntax is complete and correct overall. The diagram does not need to compile at this stage, but successful compilation and execution strongly indicates that the syntax is correct.
- ☐ All non-functional requirements that are applicable to the system have been captured and documented.

A

- ☐ Project documentation demonstrates that it will satisfy the needs of all stakeholders to their satisfaction