Capstone Projects
SE 491

Mid-Spring Guidelines

Dr. Simon Fan

Professor of Computer Science and Software Engineering

California State University ● San Marcos

sfan@csusm.edu

Time table

6	Student	Students start to schedule Phase III presentation	Monday,	February 27,	2023
7	Student	Students start to submit Report Version 3.0	Monday,	February 27,	2023
8	Student	Submission deadline for Report Version 3.0	Monday,	March 06,	2023
9	Student	Students start to give Phase III presentation	Monday,	March 06,	2023
10	Advisor	Faculty evaluation deadline for Phase III presentation	Friday,	March 10,	2023
11	Student	End of Sprint-006	Sunday,	March 12,	2023
12	Student	Students start Agile Sprint-007	Sunday,	March 12,	2023
13	Advisor	Faculty evaluation deadline for Report Version 3.0	Monday,	March 13,	2023
14	Student	Students start to submit Report Version 3.5	Wednesday,	March 15,	2023

What's in Report V 3.0

CapStone→Student→Feedbacks→Project Technical Reports

Technical Quality

Criteria 4.1 — Should explain the potential local or global impact of their capstone project on three aspects: individuals, organizations, and society — *c.f. Section 5.3*

0	1	2	3	4
No provided	Provided potential impact on one aspect only	Missing potential impact on one aspect and no explanation is provided	Provided potential impact on two aspects, and explained why when an aspect is not applicable to the project	Provided potential impact on all three aspects

Criteria 4.2 — Quality of system modeling: system models should conform to standard practices such as ER diagrams and UML diagrams (UML diagrams should follow the UML notations and practices given in the CapStone system) — *c.f. Section 6*

0	1	2	3	4
No system model is provided	All the system models are not documented in appropriate diagrams	Most of the system models are not documented in appropriate diagrams	A few system models are not documented in appropriate diagrams	All the system models are documented in appropriate diagrams

Criteria 4.3 — Change management: report should be thoroughly revised to reflect new changes made in user requirements, system design, or implementation — *c.f. Section 4, 6, 7, 8*

0	1	2	3	4
None of the newly added user requirements are engineered into system requirements	Missing use cases for newly added requirements	System designs do not cover newly added/changed requirements	System implementation is inconsistent with system designs	Requirements, system designs and system implementation are consistently documented

Criteria 4.4 — System Models [Architecture Design]: the system architecture design should be appropriate and sound — *c.f. Section 6.1*

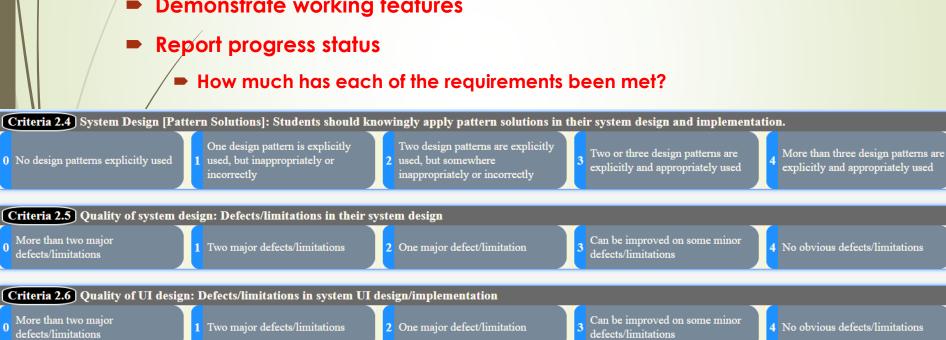| 0	1	2	3	4
No architecture design provided	The architecture design is trivial	Important components are missing in the architecture design	Key components are included in the architecture design, but their relations are inappropriate or incorrect	Key components and their relations are excellently illustrated

What's in Report V 3.0

- Some reminders:
  - Major changes summarized in the section Report Revision History
  - Has report been thoroughly revised to reflect the changes of user requirements, if any?
  - Has report been thoroughly revised to reflect any changes made in system development?
  - System designs are documented in Sections 6.
  - **Most test case designs and execution reports** are documented in Section 8.
  - Sections 1--5, 7, 9, and all appendices are revised to reflect new changes along the agile process
  - In case user requirements have been changed (or newly added), did the team refine them into system requirements, and revise the system design to incorporate the changes?

What's in Presentation

- Some reminders:
  - Review needs
  - Review system design
  - **Focus on implementation, testing**
  - **Demonstrate working features**
  - **Report progress status**
    - **How much has each of the requirements been met?**

Criteria 2.4	System Design [Pattern Solutions]: Students should knowingly apply pattern solutions in their system design and implementation.			
0 No design patterns explicitly used	1 One design pattern is explicitly used, but inappropriately or incorrectly	2 Two design patterns are explicitly used, but somewhere inappropriately or incorrectly	3 Two or three design patterns are explicitly and appropriately used	4 More than three design patterns are explicitly and appropriately used

Criteria 2.5	Quality of system design: Defects/limitations in their system design			
0 More than two major defects/limitations	1 Two major defects/limitations	2 One major defect/limitation	3 Can be improved on some minor defects/limitations	4 No obvious defects/limitations

Criteria 2.6	Quality of UI design: Defects/limitations in system UI design/implementation			
0 More than two major defects/limitations	1 Two major defects/limitations	2 One major defect/limitation	3 Can be improved on some minor defects/limitations	4 No obvious defects/limitations

Criteria 2.7	System Implementation: Were they able to demonstrate a partially functioning system?			
0 20% functional	1 30% functional	2 40% functional	3 50% functional	4 60% functional

Presentation Scheduling