**King Saud University**

**College of Computer and Information Sciences**

**Department of Software Engineering**

**SWE 496 – Graduation Project Evaluation Form**

**Grades Distribution (Assigned by Advisor)**

**Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Student Id: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Project Title: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Project Advisor: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Marks Obtained: /100**

|  |  |  |
| --- | --- | --- |
| **Item** | **Marks** | **Justification** |
| **A1**: **Written Report [65%]** |
| Overall Organization and Presentation | **/5** |  |
| English | **/5** |  |
| **Technical Content** |
| Abstract/ Introduction/ Problem Definition | **/5** |  |
| Domain Study/ Initial Project Plan | **/5** |  |
| QA Plan | **/5** |  |
| Realistic Constraints and Risk Assessment | **/5** |  |
| System Analysis [requirements, use case templates, analysis classes, sequence diagrams] | **/15** |  |
| Design and Architecture [design classes, architecture, deployment diagram] | **/10** |  |
| Mock-up Inteface | **/5** |  |
| Conclusions and Future Work / Ethics principles adopted [proper references and citation, no plagiarism] | **/5** |  |
| **Sub-Total (A1)** | **/65** |  |
| **A2: Oral Presentation [20%]** |
| Clarity of Purpose | **/2** |  |
| Technical Content [Identify problems faced/ solution approaches/ depth of concept] | **/10** |  |
| Presentation Skills [Spoke clearly/ made eye contact/ stayed in time limit/ looked at audience] | **/5** |  |
| Discussion [Participated in question answering/ interactive]­ | **/3** |  |
| **Sub-Total (A2)** | **/20** |  |
| **A3**: **Competency, Dedication, and Team Work [15%]**  |
| Regular meeting with advisor/Team work | **/5** |  |
| Timely submission of required deliverables | **/10** |  |
| **Sub-Total (A3)** | **/15** |  |
| **Total Marks Earned [A1+A2+A3]:** | **/100** |  |

**Comments by Advisor:**

**Name of Advisor**: **Signature**: **Date**:

**Checklist SWE 496 Final Report Evaluation**

The following are characteristics of excellent work on the corresponding key criteria in the evaluation of the written report part of SWE 496 – Graduation Project Evaluation Form.

**Overall Organization and Presentation**

**English**

* Report should be coherent and well organized following the provided report template.
* Grammar, spelling, punctuation and formatting should be flawless, which allows the reader to focus on the message.
* Figures and tables should be numbered appropriately and captions should be used to explain the corresponding figures and tables.

**QA plan**

* The quality plan should list the types of reviews (inspections, walkthroughs, etc.) that will be performed.
* Sample written reports of the review findings and follow up on the inclusion/correction of all findings should be included.
* The testing section should include a rough test plan. The test plan should also define several test cases for the important use cases.

**Realistic constraints and risk assessment**

Constraints:

* Relevant technical and/or resource constraints should be listed.
* Listed constraints should be professional; personal and subjective constraints like “we have several courses this session” should not be mentioned.

Risk Assessment:

* Relevant technical, human resource, and project stakeholders and environment’s risks should be listed.
* Likelihood and impact should be numerically stated.
* Avoidance and/or mitigation plans should be determined.
* Monitoring procedures should be clearly defined.

**System analysis**

Requirements:

* Important requirements should not be missed.
* Requirement statements should be clear (there should be no ambiguity).
* Requirement statements should be written using a consistent style.
* Non-functional requirements should be measurable (fit criteria should be defined).
* Requirements should be well-organized. Merely listing a large number of requirements without any organizing effort should be avoided.
* Types of requirements should be defined (functional, non-functional, and design constraints)
* Requirements should be abstract (avoiding making unnecessary design decisions).

Use Case Model:

* All use cases should be found. The elaborated use cases should meet all functional requirements.
* Use cases should have unique, intuitive, and explanatory names.
* The UC diagram should use correct UML notation.
* Use case relationships should be correct (e.g., *includes* and *extends* relationships).
* UC descriptions of basic and alternative flows should be clear and correct. In particular, use case preconditions and post-conditions should be stated correctly.
* UC descriptions of basic and alterative flows should be abstract (avoiding making unnecessary design decisions). In particular, statements like ‘click a button’, ‘move mouse’, ‘open window’, etc. should be avoided.

Analysis Class Model:

* An interaction diagram (sequence diagram) should be used to develop the analysis class model individually for every important use case.
* Analysis class diagram should use correct UML notation.
* Analysis class diagram should include all necessary boundary, control, and entity classes.
* Analysis classes should include all necessary attributes and operations. Attribute types and method signatures need not be defined at this stage.

**Design and Architecture:**

Architecture Design:

* Logical layers/subsystem design should be clearly defined.
* The design should show relationships/connections between layers/subsystems.
* Appropriate architectural style should be chosen. The choice should be justified.
* Alternative styles should be discussed (if applicable).
* Constraints that affect the way the architecture can be implemented should be discussed (if applicable).

Detailed Design:

* Detailed class diagram should be based on the analysis classes identified earlier.
* Detailed information of each class should be clearly defined such as:
	+ Data attributes
	+ Operations (parameter and return types should be defined)
	+ Data structures
	+ Algorithms (if applicable)
* Database design should describe the necessary tables and columns.

**Mock-up Interface:**

* User interface mockups should be used to illustrate the most important use cases.