

Research and Innovation Supervision Guidelines

The purpose of this guide is to ensure that there is uniformity on how different supervisors conduct their supervision and also to blend with what is being taught in lectures. Lectures are intended to cover some of the more technical details on how to write the reports, such as the writing style, team dynamics, editing with latex etc.

Supervisor

The supervisor is the one that is going to mark the documentation; therefore, they are supposed to guide the student in producing a document that is up to the expected standard. They will accomplish this by constantly meeting with their project groups to check if the group is following their project plan.

The supervisor is also the client in these types of project. They are supposed to agree with students on the requirements to put on the selected project. The requirements should be well outlined and be used as guidelines on the development of the system. The supervisor is also supposed to agree on a software development methodology with students and help them adopt it to their projects. The methodology should be followed through out the project to the end and even be documented on report.

Example of methodologies could be: **Spiral, XP, Scrum, RUP** etc. Students can be asked to research on any methodology of their choice and adopt it to the project with the help of the supervisor. The supervisor or the students are not limited to only the above methodologies; they can select any methodology of their choice.

Supervisions

The supervisor is supposed to keep a file of all the submissions made to them by their groups. The submissions will be used later on in accessing the students or determining if naughty members need to be awarded marks.

Project Supervision Guide

| Week No. | Meeting action points/ notes | Deliverable for following week |
|--------------------------------------|---|--|
| Week 1 (Understanding the System) | <ul style="list-style-type: none"> Familiarize students with the project domain to establish if they understand the system they are supposed to develop. Discuss with students on the kind of possible reports users of the system will like to have. Arrange meeting times with students Ask students to look for existing systems on the problem domain. These will help them on establishing requirements of their own Ask them to write the project brief on a single page, which summarizes their project. This will be used as a quick guide to explain what they are trying to accomplish Ask students to choose their group leader and report editor. The editor should do little coding as they will compile the report Ask students to always take minutes of the meetings that they hold, indicating the members that are present | <ul style="list-style-type: none"> Requirements Software development methodology chosen |
| Week 2 (Requirements Analysis) | <ul style="list-style-type: none"> Revise the requirements with students to establish if they are feasible Pay attention to the reports from the system, which should be detailed out Discuss the software development methodology chosen for project: Timelines and deliverables, process flows, adaptation to project etc. Discuss project brief and file it. (It will be handy in subsequent meetings) Ask students to do a through analysis of the requirements, possibly mentioning use case scenarios of how activities will follow. Ask them to break them down into a numbered list, which will help them trace the requirements Ask students to think about the various units that will be needed to make the system function fully. | <ul style="list-style-type: none"> Gantt chart with timelines Requirements analysis: Use case scenarios, Use case diagrams, Activity Diagram Ask them to draft the introduction section of the report |
| Week 3 (System Design) | <ul style="list-style-type: none"> Discuss with students any existing system or system that offers similar solution. This will help them write the background literature chapter on their report. | <ul style="list-style-type: none"> Background Literature chapter System Design: Class Diagrams, |

| | | |
|---|---|--|
| | <ul style="list-style-type: none"> • Discuss with students on how they intend to build their system. The discussion should cover at length how the various components of the system will address the requirements stated. • Emphasis that no coding should be done at this point and much time should be spent on modeling, which will make it easier for them to write the code. All requirements should be accounted for in the various classes established. • Discuss with students how they intend to store the data on the database. They should use mysql or ms sql server for database. No MS Access • Check the meeting minutes made by the group, to see who has been coming for the meetings • Discuss the Gantt chart, to see if its feasible | ERD Diagrams |
| Week 4 (Design Thinking, Mind Map) | <ul style="list-style-type: none"> • Students should be encouraged to follow design thinking approach discusses in class. Things like mind maps should be created at this point. These should give a clear visual effect of the system in question. • Ensure that students have created mind maps for the system and they all have a clear visual understanding of the organization of the system. • These should feed the system design created in the previous weeks. | <ul style="list-style-type: none"> • Mind Maps mapping to the design of the system. |
| Week 5 (Coding & Reporting) | <ul style="list-style-type: none"> • Discuss the ERD diagrams they had made. All the data storage should be accounted for by a table • Discuss the class diagrams or any other models that students have done. Pay attention to the requirements, they should all be represented in class diagrams. • If students have not yet done some sequence diagrams, ask them to use them to clarify certain use case scenarios. • If the models are clear, advice students to start coding. • Ask students to show you how tasks have been allocated from the design of the system | <ul style="list-style-type: none"> • Preliminary documentation with introduction, background & design section |
| Week 6, 7 (Coding & Reporting <i>cont..</i>) | <ul style="list-style-type: none"> • Discuss the report with student, noting content structure, references, images etc. They should use LATEX only. • Ask students to show you the work they have done on the system, noting the screens if they are appealing, if the system is upto standard etc. • Take note of the execution of the tasks by group members | <ul style="list-style-type: none"> • Implementation section on report |

| | |
|--|--|
| | <ul style="list-style-type: none"> • Ask students to make use of a spelling checkers on their reports |
| Week 8 (Validation of requirements) | <ul style="list-style-type: none"> • Ask students to trace their requirements on the system, taking note of those left behind. • Ask them to update their gant chart as they go along. There should be gant chart at the end, that shows exactly how long each activity took |
| Week 9 (Reporting) | <ul style="list-style-type: none"> • Validate the report for proper flow from one chapter to another. Check if the editor is doing their job by keeping consistency across the report in terms of what has been written. |