Graduation Project Guidelines

** This document is intended to serve as a guidelines book not a template.

The types of graduation projects at the College of Information Technology and Computer Engineering proposed by the students can be divided into three categories. The structure and the style of the project's report can vary based on the category. Below are these categories and the suggested outline for the each of the types.
(The projects mainly divided by category not major of study).

1. Research Based Projects (Some of the Computer Science and Computer Engineering)
   Example: Using ECG as a security measure.

   - The size of the report for this kind of projects is 20-25 pages (without appendices). Add appendices if needed.
   - Software engineering process is not required for this type.

   Outline:

   - Acknowledgement
   - Abstract:
     A short paragraph (250-300 words) that describes the larger work of the project. It should be concise, stand-alone, self-contained paragraph. It may contain the motivation, problem statement, approach, result, and conclusion.
     Abstract format:
     Arial, 12, single-spaced
   - Table of Content
   - List of Tables
   - List of Figures
   - Introduction:
     It might contain an overview, motivation, importance, objectives, and short description about the problem statement, limitations, and overview of the report sections.
   - Problem Statement: (can be an independent chapter or part of the introduction)
     Problem analysis, definition, list of requirements, expected results, contribution if any.

- Literature Review:
  This section should include a literature survey.

- Design/Method:
  It may include the detailed method description, detailed design, block diagrams if needed, flow charts if needed, algorithms/pseudo-code if needed, data collection methods, and any necessary information about the study.

- Results:
  It should include the experiment validation, results, and analysis.

- Discussion:
  It should include detailed discussion and analysis of the results in the context of the experiment/problem statement and the context of the literature described in previous sections. It may also include conclusion items and recommendations

- Conclusions:
  This section brings all parts together in an informative summary. It should be stand-alone and meaningfull if read independently. It should also include future directions and future work.

- References: [Decide on a style]

- Appendices: (if needed)

  Format:
  Text : Times New Roman, 12, 1.5-spaced

2. System Based Projects (Computer Engineering projects)
   Example: Using microcontroller to control home temperature.

   - The size of the report for this kind of projects is 20-25 pages (without appendices). Add appendices if needed.
   - Software engineering process is not required for this type.

   Outline:

   - Acknowledgement

   - Abstract:
     A short paragraph (250-300 words) that describes the larger work of the project. It should be concise, stand alone, self-contained paragraph. It may contain the motivation, problem statement, approach, result, and conclusion.
     Abstract format:
     Arial, 12, single-spaced

- Table of Content

- List of Tables
- List of Figures

- Introduction:
  It might contain an overview, motivation, importance, objectives, short description of the system, and overview of the rest of report sections.

- Problem Statement: (can be an independent chapter or part of the introduction)
  Problem analysis, definition, list of requirements, expected results.

- Background:
  Theoretical background and literature if needed, short description of the parts used in the system and why they are chosen, specification and design constraints, and any additional information the reader might need to understand the system.

- Design:
  It might include detailed conceptual description of the system, detailed design, schematic diagrams, block diagrams if needed, and any necessary information about the design.

- Software:
  Description of the software if needed, flow charts if needed, algorithms/pseudo-code if needed, and any necessary information about the software developed to drive the system.

- Validation and Discussion:
  It may include description of the implementation, implementation issues, and implementation challenges. It should also include description of the method used to validate the system, validation results, analysis and discussion about the results, and recommendations based on the results.

- Conclusion:
  This section brings all parts together in an informative summary. It should be stand-alone and meaning full if read independently. It should also include future directions and future work.

- References: [Decide on a style]

- Appendices: (if needed)

  Format:

Text: Times New Roman, 12, 1.5-spaced

3. Software Engineering Projects (IT and some Computer Science projects)
   Example: PPU Registration System

   - The size of the report for this kind of projects is 40-50 pages. Add appendices if needed.
   - Software engineering process is mandatory for this type.
     In this type the students and their advisors should be aware that not all software engineering artifacts fit all projects or needed by all projects. For example, a state machine diagram might fit with some software projects but does not fit others. It depends on the students' decisions in the design/analysis phase. Use proper UML diagrams where necessary only.

Outline

- Acknowledgement

- Abstract: (executive summary)
  A short paragraph (250-300 words) that describes the larger work of the project. It should be concise, stand alone, self-contained paragraph. It may contain the motivation, scope, objectives, and conclusion.
  Abstract format:
  Arial, 12, single-spaced

- Table of Content

- List of Tables

- List of Figures

- Introduction:
  It may contain an overview, motivation, scope of the project, objectives and the procedure to achieve them, short description of the system, and overview of the document. The objectives discussion should be based upon the project's context and the end user needs.

- Requirements Specification:
  Functional, non-functional, domain requirements, and any supportive information necessary to understand the project's requirements.

- Software Design:

System model, tools, architectural design, object identification, database design. Description of design decisions and why they were taken.
The system model might include graphical system models to show the relationships between the system components, the system, and its environment. Examples of possible models are object models, data-flow models, or semantic data models. See Ian Sommerville's software engineering book for more details.

- Software Demonstration: *(discuss omitting this section)
  This chapter is intended to demonstrate the developed software. It may include implementation details, implementations issues, snapshots and short description of the UI. It might also include any important information and description about the implementation.

- Testing:
  Test plan, test design specification, test case specification, test procedure. Explain the criteria used to evaluate the requirements, the design, and the implementation.
  The test plan should include features to be tested, features not to be tested, and description of the testing environment and tools.
  The student should design a proper test case format that at least includes tracking criteria, purpose, inputs, expected output and pass/fail criteria, and execution criteria.
  The student should include in an appendix a log that includes the result of the test executions. The log may include test case result, bugs found and reported, bugs fixed,…

  This chapter should include a section to show that the implementation matches the proposed design. For example this section might show that the class diagram matches the implemented classes.

- Conclusion:
  Summary of the delivered artifacts, challenges faced, interesting decisions taken, recommendations to modify/update the software, and list of features to be added in future releases.

- References: [Decide on a style]

- Appendices: (if needed)

  Format:
  Text: Times New Roman, 12, 1.5-spaced

[Develop assessment criteria]

Recommendations:

- Students should discuss with their advisor which format the best match for their project.
- Encourage analytical projects/studies for Computer Science such as study the characteristic of good libraries, balancing between agility and discipline,…
- Develop an assessment criteria and rubric for each of the projects categories.