**COMP102 - Software Development Processes**

**Term Project Report**

**Submitted By:**

**Submitted To: Asst. Prof. Dr. Ali Cihan Keleş**

**Date:**

**Section:**

**Project Title (Topic) (Common for all group members):**

* Clearly state the topic or purpose of your project in the report.
* Make it concise, relevant, and descriptive.

**Group Members (Common for all group members):**

* List the names of all group members involved in the preparation of your project.
* Include roles or contributions for each member.

**Abstract (Common for all group members): *(75 – 100 words)***

* Provide a concise summary of your project's purpose, scope, key findings, methods, and conclusions.
* Ensure this section gives a quick overview of the entire report of your project.

**Introduction (Common for all group members):**

* Set the stage for the report of your project by explaining the context, objectives, and importance of the topic.
* Provide background information to help understand the subject matter.
* Research and literature review provided during the improvements of your project
* Citations for the articles, books journals or periodicals
* Key and brief explanations about each section and stage of the report of your project etc.

**Planning and Requirements Analysis (Common for all group members):**

* Define the software's purpose and scope. Understand the end-users' needs and the goals the software should meet. Essentially, need to ask, "What problem will this software solve?" and "What value will it offer to the user?"
* Evaluate technical and financial challenges that might affect the software's development or success (feasibility study).
* Define clear roles, responsibilities, and expectations, it lays a solid foundation for an efficient software development process.
* For requirements analysis, identify and record the precise requirements of the final users. In this phase, the team is looking to answer, "What are the expectations of our users from our software?" This is also called requirements gathering.
* Collect information to understand the user's expectations and needs. Define the software's functionality, performance, security, and interface needs.
* Outline the software's purpose, features, and functionalities, and provide cost estimates if needed. Ensure its reliability, validated for accuracy, comprehensiveness, and feasibility.

**Design (Common for all group members):**

* Outline the conceptual or technical design involved in your project. This could include schematics, diagrams (e.g., class diagrams, database diagrams, use case diagrams), or design rationale if applicable.
* Outline the software's structure, navigation, user interfaces, and database design.
* Identify system dependencies and integration points. Set the software's limitations, such as hardware constraints, performance requirements, and other system-related factors (e.g., details about the software's design, from system architecture to data design, and even user interface specifics)

**Methodology (Common for all group members):**

* Describe the processes, tools, and techniques used to carry out your project (e.g. pseudo codes, algorithms, flow-charts, appropriate programming languages, libraries, modules, key programming parts, repositories, databases, frameworks, logging, APIs, microservices, App. Controllers and services).
* Be specific and include any data collection methods, research techniques, or technical procedures.

**Implementation (Coding) and Analysis (Testing) (Common for all group members):**

* Explain how the design and methodology were executed.
* Present the results or findings and analyze them in detail.
* Use visual aids like charts, tables, or graphs where necessary.
* Examine each other's ideas or work to identify any possible bugs or inconsistencies meet.
* Consider and state how to uphold high code standards, ensuring the software's reliability and robustness and preliminary internal testing to confirm the software's basic functionality.
* Identify the necessary software conditions and outline diverse scenarios to examine these conditions to create an efficient testing strategy.
* Consider and state how to operate and sustain various types of tests, including unit testing, security testing, integration testing, system testing, and acceptance testing to provide persistent refinement until the software complies with all predetermined parameters and to ensure the software's robustness and reliability.

**Deployment and Maintenance (Common for all group members):**

* Deployment isn't just about launching the software. It's about ensuring users can operate it with ease to involve rolling out the meticulously tested and fine-tuned software to its end-users (e.g., creating user manuals, conducting training sessions, or offering on-site support). Consider and state how to shift from a project phase to a product phase, where the software begins to fulfill its purpose.
* Maintenance is a continuous process that ensures that the software product is up-to-date and running efficiently. Consider and define how to encompass frequent software updates, implementing patches, and fixing bugs. User support is also a crucial component, offering help and guidance to users facing difficulties with the software.
* Consider and state how to adapt to the software's changing needs (e.g., long-term strategies, for instance, upgrading or replacing the software).

**Further Studies and Recommendations (Common for all group members):**

* Suggest areas for further research or improvements.
* Provide actionable recommendations based on the analysis and findings.

**Conclusion and Discussion (Individual for each group members):**

* Summarize the key findings, their significance, and their alignment with your project’s objectives.
* Discuss any challenges faced and how they were addressed, along with key takeaways.

**References and Documentation (Common for all group members):**

* Cite all sources, references, and supporting materials used in the report of your project.
* Ensure citations follow a consistent style (e.g., IEEE (considerably recommended), APA, MLA, Chicago).
* Include documentation such as logs or detailed notes if relevant.

**Appendices (if necessary) (Common for all group members):**

* Include supplementary materials like raw data, additional charts, or detailed explanations that support the report of your project but are not essential to the main sections.