**Project Phase 3 - Tasks** (Project Management)

Keep all your work in the repo; do **not** submit anything on Canvas.

|  |
| --- |
| **Task 1 - Quality Checklist** |

A project quality checklist is a tool used to aid the project team in ensuring they consider all aspects of project and/or process quality.  The purpose of well-planned and repeatable quality management is to ensure the delivery of products or services which are acceptable to the customer based on some agreed upon standard of quality.  To help achieve consistency many organizations use a standard checklist to verify that all quality considerations have been met during the project planning, execution, and monitoring/controlling phases.  Based on the project some checklists may differ, but it is often advantageous to utilize a standard checklist when appropriate to achieve consistency.

Please click the link below to download the template of the Quality Checklist.

[**CPT-200 - Project Phase 3 - Template of Quality Checklist - DOC**](https://stchas.instructure.com/courses/43392/files/9702053/download)

|  |
| --- |
| **Task 2 - Logical Design Documents** |

•Use Case Document

The Use Case Document is a business document which provides a story of how a system, and its actors, will be utilized to achieve a specific goal.  An effective use case should provide a detailed step-by-step description of how the system will be used by its actors to achieve the planned outcome.  The purpose of the use case is to tie the business needs of the system to the design parameters of the system to ensure that the completed system achieves the goals established by the business requirements.  The level of detail in use cases may vary greatly depending on the size and complexity of the system being designed.

Please click the link below to download the template of the Use Case Document.

[**CPT-200 - Project Phase 3 - Template of Use Case Document - DOC**](https://stchas.instructure.com/courses/43392/files/9702410/download)

•Entity Relationship Diagrams (ERD)

This data modeling technique provides a precise method for detailing and illuminating the interrelationships of the data used by a system.  You can depict the "entities" in the data you are modeling and the relationships between them by drawing them onto an entity relationship diagram (ERD).  The data model (ERD) shows the major data objects of an application and how they fit together using the relationship.  Create an ERD for your system.

Please click the link below to access the document that helps you better understand the ERD.

[**CPT-200 - Project Phase 3 - Entity Relationship Diagrams - PDF**](https://stcharlescommunity-my.sharepoint.com/:b:/g/personal/dwang_stchas_edu/ETJp7oZhShtNhHtrwpR1H7MBexmXXNcAv4ZaR04yWldbIw?download=1)

•Data Flow Diagram (DFD)

Data Flow Diagrams (DFDs) reveal relationships among and between the various components in a program or system.  DFDs are an important technique for modeling a system's high-level detail by showing how input data is transformed to output results through a sequence of functional transformations.  DFDs consist of four major components: entities, processes, data stores, and data flows.  Create Level 0 and Level 1 DFDs for your system.

Please click the link below to access the document that helps you better understand the DFD.

[**CPT-200 - Project Phase 3 - Understanding Data Flow Diagrams - PDF**](https://stcharlescommunity-my.sharepoint.com/:b:/g/personal/dwang_stchas_edu/ETPwIrd-zUxLhACVX4ieO5MBAxkjYh7IzkcrQRJe4eVJoA?download=1)

|  |
| --- |
| **Task 4 - Teamwork** |

•While the developers are working on the coding part of the system, the project manager is responsible to communicate with them, asking for any changes of the system, e.g., system architecture change, new services added, changes of features, and so on.  Please note that any change of the system **requires** changes in related documents, even for those documents already completed in previous phases.  Every time when code-document consistency is broken, the project manager needs to update (or let someone else update) the documents in a timely manner.

•The project manager should be the first user of the system.  For any progress in development, the project manager needs to experience the new added features or other changes.  Then, necessary feedback and communication should be done.

•The project manager **cannot** put aside the system development.  Even if the project manager does **not** understand a lot of code, he/she still needs to understand the development process.  The more a project manager understands the system, the better he/she can manage the project.

•Although the project manager is assumed the major part of the documentation work, he/she is **not** supposed to complete all the documents by himself/herself.  The project manager, however, needs to find the right person to do the right thing.  For example, it is good for the project manager to ask the developer who designed the system architecture to draw the architecture diagram, instead of drawing the diagram by himself/herself without an in-depth understanding of the system.  At the end of this project phase, it is the project manager's responsibility to check and finalize all the documents.

Keep all your work in the repo; do **not** submit anything on Canvas.