Analysis and Design Document (ADD) Guide

This document is the continuation of the project specification document (PSD) and it aims to present analysis and design of the project. The document should include following sections based on the category of the project: *Software-Oriented* or *Academic-Oriented*. Note that the category of the project will be determined by your advisor.

The layout of the Document for a Software-Oriented Project

Title Page
1. Introduction
    1.1 Problem Description and Motivation (revised from PSD)
    1.2 Scope of the Project (revised from PSD)
    1.3 Definitions, acronyms, and abbreviations
2. Literature Survey (revised from PSD)
3. Project Requirements
    3.1 Functional Requirements
    3.2 Nonfunctional Requirements
4. System Design
    4.1 UML Use case Diagram(s) for the main use cases
    4.2 UML Class and/or Database ER diagram(s)
    4.3 User Interface (Preliminary version)
    4.4 Test Plan
5. Software Architecture (main aspects of data flow/control flow/modular design)
6. Tasks Accomplished
    6.1 Current state of the project (implementation and preliminary results)
    6.2 Task Log (information about meetings and activities, including date, short description and hours)
    6.3 Task Plan with Milestones (clear and well-defined descriptions of the work that must be completed before predetermined check points, illustrated by Gantt chart)
7. References

The layout of the Document for an Academic-Oriented Project

Title Page
1. Introduction
    1.1 Problem Description and Motivation (revised from PSD)
    1.2 Scope of the Project (revised from PSD)
    1.3 Definitions, acronyms, and abbreviations
2. Related Work (Comprehensive literature survey to present state-of-the art methods/algorithms)
3. System Design
    3.1 System Model
    3.2 Flowchart and/or pseudo code of proposed algorithms
    3.3 Comparison metrics (provide detailed explanation)
    3.4 Data sets or benchmarks (provide detailed explanation)
4. System Architecture (main aspects of data flow/control flow)
5. Experimental Study (draft version to summarize details of the experiments)
6. Tasks Accomplished
    6.1 Current state of the project (implementation and preliminary results)
    6.2 Task Log (information about meetings and activities, including date, short description and hours)
    6.3 Task Plan with Milestones (clear and well-defined descriptions of the work that must be completed before predetermined check points, illustrated by Gantt chart)
7. References

Details of the Selected Sections (for a Software-Oriented Project)

3. Project Requirements

Requirements are statements of the properties/services that software system of your project must deliver and the constraints under which it runs smoothly. They basically define the application of the software being specified. Therefore, they should be:

- Correct
- Unambiguous
- Verifiable (i.e., testable)
- Complete
- Consistent
- Uniquely identifiable via numbering

3.1 Functional Requirements:

Functional requirements reflect how the system of your project should respond to specific inputs and how it principally behaves in particular situations. They essentially describe functionality and system services in detail. Your project's functional requirements should be listed in the following numbering/sections:

3.1.1 Functional Requirement #1
3.1.1.1 Description (with one descriptive/informative sentence of the requirement)
3.1.1.2 Inputs (the data is entered/processed by the functional requirement)
3.1.1.3 Processing (expected operational steps/behaviors of the functional requirement)
3.1.1.4 Outputs
3.1.1.5 Error/Data Handling (expected error/data mechanism in order to control unexpected/risky situations of the functional requirement)

3.2 Nonfunctional Requirements:

Nonfunctional requirements show constraints, standards, properties of your project. You should present your project's non-functional requirements for the following attributes in measurable terms:

3.2.1 Performance (Number of transactions at a unit time, response time…etc.)
3.2.2 Reliability (Rate of failure occurrence, mean time to failure,)
3.2.3 Usability (Ease of use, help frames, warning/error messages, help manual)
3.2.4 Security (Unauthorized access to the system)
3.2.5 Maintainability (Ease of repair)
3.2.6 Portability (% target system dependent statements)

4. System Design (for a Software-Oriented Project)

4.1 UML Use case Diagram(s) for the main use cases

In this section you should draw use case diagram(s) of your project which describe the sequence of interactions between actors and the use cases in order to satisfy the goal of the determined behavior of your project. A use case is initiated by a user with a specific aim and completes successfully when that aim is fulfilled.

4.2 UML Class and/or Database ER diagram(s)

This section should include the class diagram(s) which shows the building blocks of your project. Class diagrams represent a static view of your model, describing what attributes and behavior it has for achieving targeted operations. You also better to use generalizations, aggregations, and associations in your class diagram(s) of your project.

4.3 User Interface (Preliminary version)

In this section, you are expected to provide sample screen-shots of your project's Graphical User Interfaces.

4.4 Test Plan

Test plan should summarize the software items and their related combinations of your project to be tested including the following properties: the approach to testing, any necessary test support tool(s), and necessary properties of the test environment (hardware, software libraries ...etc.). Furthermore, test plan should also include estimated calendar time required to do each testing task/milestone.

5 Software Architecture (main aspects of data flow/control flow/modular design)

The software architecture section of this document should include the high-level structure of your project's software system. The software architecture of your system needs to identify a set of components and their corresponding relations/interactions that collaborate to achieve the system goals.

6 Tasks Accomplished

6.1 Current state of the project (implementation and preliminary results)

Here, you should clearly indicate which parts of your project you have completed so far. You need to be sure that there are enough details of the completed part of your implementation. Furthermore, you also should present your preliminary experimental results about the tasks accomplished so far.

6.2 Task Log (information about meetings and activities, including date, short description and hours)

In this section, you need to list all the related meetings and events in a format that you deal with your advisor. There can be different acceptable formats for this part. One sample format is as follows:

Meeting#1:
Date:
Location:
Period:
Attendees:
Objectives:
Decisions and Notes:

6.3 Task Plan with Milestones (clear and well-defined descriptions of the work that must be completed before predetermined check points, illustrated by Gantt chart)

In this part, you are expected to fill the below Task-Schedule Table with your project's tasks and their corresponding details.

Table1 Task/Calendar-Schedule

Task No	Task Description	Expected Output	MONTHS					
			1	2	3	4	5	6

Details of the Selected Sections (for an Academic-Oriented Project)

2. Related Work (Comprehensive literature survey to present state-of-the art methods/algorithms)

In this section of the document, first you need to give related background concepts/information about your project's topic. You should make an extensive literature survey on your project's topic under the supervision of your advisor. Then you should summarize each study with its aim, methodology, experiment setting, experimental results and its strength and weakness in compare to other approaches in the literature. You need to be sure that, you write this section with completely in your own sentences with required references. This is important to prevent any plagiarism in your document.

3. System Design

3.1 System Model

You should present your methodology with all its theoretical details, formulas, terms…etc.

3.2 Flowchart and/or pseudo code of proposed algorithms

You should give prepare Flowchart and/or pseudo code of proposed algorithms neatly. You need to be sure that, you give enough details; maybe it will be better to prepare a separate flowchart and /or pseudo code for different modules of your project.

3.3 Comparison metrics (provide detailed explanation)

Here, you should select a comparison metric (i.e., Accuracy calculation, F-score, Precision/Recall…etc.) with your advisor of your algorithm which will be suitable to compare your results with those of the baseline algorithms in the literature. It will be also a good idea to calculate gain calculation or student's t-test in order to highlight the performance differences between baseline algorithms and your approach.

3.4 Data sets or benchmarks (provide detailed explanation)

Here, the datasets and their features need to be listed.

4. System Architecture (main aspects of data flow/control flow)

The software architecture section of this document should include the high-level structure of your project's software system. The software architecture of your system needs to identify a set of components and their corresponding relations/interactions that collaborate to achieve the system goals.

5. Experimental Study (draft version to summarize details of the experiments)

You should detail this section under the following subtitles:

Experimental Setup: The experimental environment, conditions…etc. need to be detailed.
Experimental Results: The experimental results of your algorithms and the baseline algorithms required to be reported here.
Discussions: In this section, you should explain the difference of the experimental results between your algorithm and the baseline algorithm in the literature. You should report your notes from your observations on how your algorithm gets these experimental results in compare to the other methods.

6. Tasks Accomplished

6.1 Current state of the project (implementation and preliminary results)

Here, you are expected to give the details of the completed part of your implementation. You also need to report your preliminary experimental results.

6.2 Task Log (information about meetings and activities, including date, short description and hours)

In this section, you need to list all the related meetings and events in a format that you deal with your advisor. There can be different acceptable formats for this part. One sample format is as follows:

Meeting#1:
Date:
Location:
Period:
Attendees:
Objectives:
Decisions and Notes:

6.3 Task Plan with Milestones (clear and well-defined descriptions of the work that must be completed before predetermined check points, illustrated by Gantt chart)

In this part, you are expected to fill the below Task-Schedule Table with your project's tasks and their corresponding details.

Table1 Task/Calendar-Schedule

Task No	Task Description	Expected Output	MONTHS					
			1	2	3	4	5	6