Technical Document for "Your Project Title"

Your Names

Supervised by Supervisors Names

Faculty of Computer Science
Misr International University, Cairo, Egypt

Abstract

Giving a short overview of the work in your project. Maximum one page.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Setting out the aims and objectives of your project, explaining the overall intention of the project and specific steps that will be taken to achieve that intention.

1.1 Motivation

Explaining the problem being solved.

1.1.1 Problem Statement

Problem statements lead the reader from a shared context to the perception of a problem, and on to a proposed solution.

1.2 Aims and Objectives

Aims and Objectives here.

1.3 Scope

Project scope here.

1.4 System Overview

Explaining what your project is meant to achieve, how it is meant to function, include a diagram for system overview.

1.4.1 Business Context

1.4.2 Users Characteristics

1.5 Project Management and Deliverables

1.5.1 Time Plan

The time plan of your project.

1.5.2 Deliverables

- What are the final outputs of your project (program, documents, etc.)

Chapter 2

Background and Related Work

Explaining what your project does that is new or is better than existing work in the same field.

2.1 Introduction

2.2 Background

Please start from very big domain for your problem then focus on some area inside this domain that match your interest.

2.3 Related Systems

Test citation [1].

2.4 Summary

short summary of the chapter...

Chapter 3

System Requirements Specification

Explaining what your project does that is new or is better than existing work in the same field.

3.1 Introduction

3.2 Functional Requirements

3.2.1 System Functions

Describes the general functionality of the product using **Use Cases diagrams**.

You also need to list project requirements. A good requirement states something that is **necessary**, **verifiable**, and **attainable**. Even if it is verifiable and attainable if it is not necessary, it is not a good requirement [2]. For example:

1. The user shall be able to search either all of the initial set of databases or select a subset from it.

2. The admin shall update security rules.

3. The user shall be able to create a new event.

3.2.2 Detailed Functional Specification

Show the details of all functions shown in section 3.2.1. Describe each function in the following structure.

Table 3.1: readFile Function Description

Name	Such as: readFile
Code	Such as FR01
Priority	Such as: Extreme, high, low
Critical	Describes how essential this requirement is to the overall system.
Description	A full description of the requirement such as: reads a file and writes it to the output buffer.
Input	Such as: File name
Output	Such as: Boolean (true for success, false otherwise)
Pre-condition	Such as: User must be logged in and file exist
Post-condition	Such as: Redirect to view file page and displays file content
Dependency	Dependencies with other requirements Describes interactions with other requirements.
Risk	Describes the circumstances under which this requirement might not able to be satisfied, and what actions can be taken to reduce the probability of this occurrence.

3.3 Interface Requirements

This section describes how the software interfaces with other software products or users for input or output. Examples of such interfaces include library routines, token streams, shared memory, data streams, and so forth.

3.3.1 User Interfaces

Use some software for primitive plan of your project. Describes how this product interfaces with the user.

GUI

Describes the graphical user interface if present. This section should include a set of screen dumps or mockups to illustrate user interface features. If the system is menu-driven, a descrip-

tion of all menus and their components should be provided.

CLI

Describes the command-line interface if present. For each command, a description of all arguments and example values and invocations should be provided.

3.3.2 Hardware Interfaces

Describes interfaces to hardware devices. (According to your Project)

3.3.3 Communications Interfaces

Describes network interfaces. (According to your Project)

3.3.4 Application Programming Interface (API)

Describes the application programming interface, if present. For each public interface function, the name, arguments, return values, examples of invocation, and interactions with other functions should be provided.

External APIs

List external APIs that you plan to use.
Give information on the API name, arguments, return values, and an example of invocation.

External Libraries

List external Libraries such as Python or Java with a brief description of each one and why it is needed in this system.

3.4 Design Constraints

Specifies any constraints for the design team using this document.

3.4.1 Standards Compliance

3.4.2 Software Constraints

Such as library/framework versions

3.4.3 Hardware Constraints

3.4.4 Other Constraints as appropriate

3.5 Non-functional Requirements

Specifies any other particular non-functional attributes required by the system. Such as: Security, Reliability, Maintainability, Portability, Extensibility

3.6 Summary

short summary of the chapter...

Chapter 4

System Design

Containing a comprehensive description of the design chosen, how it addresses the problem, and why it is designed the way it is.

4.1 Introduction

4.2 Architecture Design viewpoints

4.2.1 Context viewpoint

The Context view is often a starting point of system design. It provides a description of System's services and users. That context is defined by reference to actors that include users and other stakeholders, which interact with the design subject in its environment. The Context viewpoint provides a "black box" perspective on the design subject.

The purpose of the Context viewpoint is to identify a design subject's offered services, its actors (users and other interacting stakeholders), to establish the system boundary.

In this section you can:

- Write a description of the offered services and actors.

Design concerns: Systems services and users.

4.2.2 Composition viewpoint

The Composition viewpoint describes the way the design subject is structured into constituent parts and establishes the roles of those parts. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together. The Composition viewpoint supports the recording of the part-whole relationships between design entities using realization, dependency, aggregation, composition, and generalization relationships.

In this section you can:

- Provide a description of the architectural design, used architecture patterns such as MVC or layered architecture. Describe *design entities* such as: system decomposition into subsystems, components, modules; also used libraries, frameworks, software repositories.
- Write a high level overview of how responsibilities of the system were partitioned and then assigned to subsystems. Identify each high level subsystem and the roles or responsibilities assigned to it.
- Describe how the subsystems collaborate with each other in order to achieve the desired functionality.
- Use diagrams to show the major subsystems, data repositories and their interconnections such as: architectural design and/or UML package diagram.

Design concerns: Composition and modular assembly of systems in terms of subsystems and (pluggable) components, buy vs. build, reuse of components.

Design Rationale

Discuss the rationale for selecting the architecture described in section 4.2.2 including critical issues and trade/offs that were considered. You may discuss other architectures that were considered, provided that you explain why you didn't choose them.

4.2.3 Logical viewpoint

The purpose of the Logical viewpoint is to elaborate existing and designed types and their implementations as classes and interfaces with their structural static relationships.

Design concerns: The Logical viewpoint is used to address the development and reuse of adequate abstractions and their implementations. For any implementation platform, a set of types is readily available for the domain abstractions of interest in a design subject, and a number of new types is to be designed, some of which may be considered for reuse. The main concern is the proper choice of abstractions and their expression in terms of existing types.

In this section you can:

- Provide a UML class diagram to Illustrate static structure (classes, interfaces, and their relationships).
- Provide a table to describe each of your classes.

Table 4.1: ClassName

Abstract or Concrete:	xxxx
Superclasses	xxxx
Subclasses	xxxx
Purpose	xxxx
Collaborations	xxxx
Attributes	xxxx
Operations	xxxx

4.2.4 Patterns use viewpoint

This viewpoint addresses design ideas focusing on the used design patterns. UML class diagram and the UML package diagram can be used here to illustrate the used design patterns.

Design Rationale

You need to provide the design rationale for using these design patterns.

4.2.5 Algorithm viewpoint

In this section, provide closer look at what each component does in a more systematic way. If you gave a functional description, provide a summary of your algorithm for each function listed in section 4.2.3 in procedural description language (PDL) or pseudo-code. If you gave an OOP description, summarize each object member function for all the objects listed in 4.2.3.
Decision tables and flowcharts, "pseudo-code," or (actual) system code may also be used.

4.2.6 Interaction viewpoint

Provides description of Object communication, messaging using UML sequence diagrams

4.2.7 Interface viewpoint

The Interface viewpoint provides programmers, and testers the means to know how to correctly use the provided services. This viewpoint consists of a set of interface specifications for each entity.
NOTE: User interfaces are addressed separately in section 4.4.

4.3 Data Design

4.3.1 Data Description

Explain how the information domain of your system is transformed into data structures. Describe how the major data or system entities are stored, processed and organized.
You may consider the answer for the following questions:

- What is the original format of the data (paper based, excel files, old system database)?
- Identify the method used to capture the data into your system(web page forms, import documents etc..)?
- How large is the database expected to be (in numbers of rows and columns for the main entities)?

- What is the expected number of users/customers for the system?
- How is your entity keys(ids) constructed (is there a specific code or format for ID definition)?
- Does your data contain date/time, if yes which format are they presented in?

Provide data models diagrams such as Entity Relationship Diagram (ERD) in this section.

4.3.2 Dataset Description

If your project includes the use of a dataset provide a clear description in this section. You need to provide a general information on the dataset including Dataset name, link, type, and size. You may also provide further details such as number of images, tweets, search words, etc. Provide the number of classes in your dataset and the number of items, such as images, per class. You can customize table 4.2 given the nature of your dataset.

Table 4.2: Project name time plan

Dataset Name	name
Link	Dataset link
Size	Dataset Size
Number of classes	The number of classes will help you choose and set up a ML/DL algorithm.
Image Dimensions	Width, height (min, average max)
Number of images	xx
Image file size	(min, average max) MB
Notes	Additional information or notes about the dataset.

4.3.3 Database Design

Describe any databases (provide database schema diagram) and/or description of other data storage items.

4.4 Human Interface Design

4.4.1 User Interface

Describe the functionality of the system from the user s perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.

4.4.2 Screen Images

Display screenshots showing the interface from the user s perspective. These can be hand drawn or you can use a wireframe tool such as Adobe XD or you can include actual screen shots of your GUI if already developed.

4.4.3 Screen Objects and Actions

A discussion of screen objects and actions associated with those objects.

4.5 Implementation

Containing a comprehensive description of the implementation of your software, including the language(s) and platform chosen, problems encountered, any changes made to the design as a result of the implementation, etc.

4.6 Testing Plan

4.6.1 Test Scenario X

Describe possible users' actions and behaviors. Determine the technical aspects of the requirement. Ascertain possible scenarios of system failure. You may refer to use cases in SRS and Sequence diagrams in SDD.

Test Cases

Test Cases for the scenario mention in section 4.6.1 shown in Table 4.3

Table 4.3: Test Cases for Scenario x

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC01	xxxx	FR01		
TC02	xxxx	FR01		

4.6.2 Test Scenario Y

Describe possible users' actions and behaviors. Determine the technical aspects of the requirement. Ascertain possible scenarios of system failure. You may refer to use cases in SRS and Sequence diagrams in SDD.

Test Cases

Test Cases for the scenario mention in section 4.6.2 shown in Table 4.4

Table 4.4: Test Cases for Scenario y

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC03	xxxx	FR02		
TC04	xxxx	FR02		

4.7 Requirements Matrix

Provide a cross reference that traces components and data structures to the requirements in your requirement chapter. Use a tabular format to show which system components satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS as shown in Table 4.5.

Table 4.5: Requirements Ratrix

Req. ID	Req Desc	Class	Test Cases ID	Status
FR01	xxxx	class name	TC01, TC02	In Progress
FR02	xxxx	class name	TC03, TC04	Developed

4.8 System Deployment

Describe your system deployment.

4.8.1 Frameworks

Describe the frameworks used.

4.8.2 Tools

Describe the tools used in this system.

4.8.3 Technologies

Describe the technologies used in this system.

4.9 Summary

short summary of the chapter...

Bibliography

[1] CORMEN, T. H. *Introduction to algorithms*. MIT press, 2009.

[2] HOOKS, I. Writing good requirements. In *INCOSE International Symposium* (1994), vol. 4, Wiley Online Library, pp. 1247–1253.

Appendix A

Git Repository

Link:

Add screenshots from your repository showing your project. Also, add some charts from the insights as shown in Figure A.1

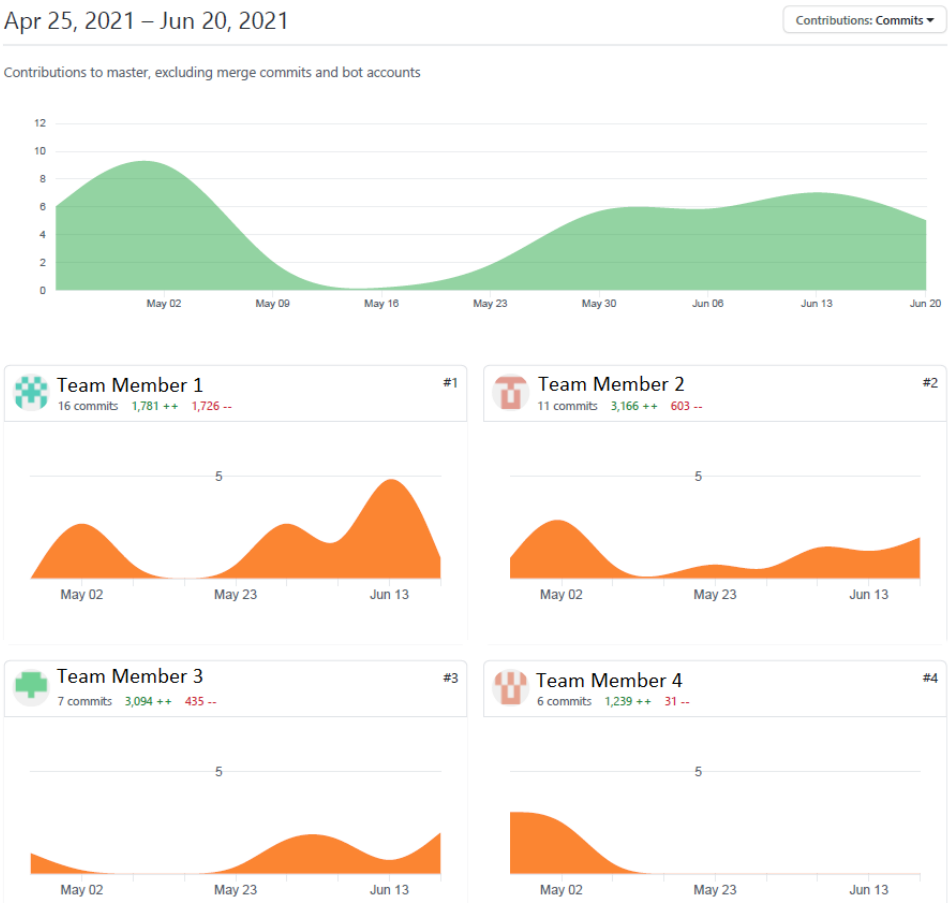

Figure A.1: Git Insights