**NOTE:**

Do Not Change the formatting of any page. All pages have already been formatted for you. Just start writing. Before writing, YOU MUST CHECK the formatting before starting writeup, so that you may remember what is the font size of heading and its text, sub-heading and its text, sub-sub-heading and its text, the text of table and so on. This box is just for hint. Remove it once you understand the hint.

# CHAPTER 1: INTRODUCTION

<*It is highly recommended that you write brief summary of every chapter as a preamble explaining what a reader would find in this chapter. Thus, when a reader reads summary of your chapter at the beginning, the forthcoming contents shall become clear.*>

## System Introduction

<*Provide a short description of the software being specified and its purpose*

TO DO: 1-2 paragraphs describing the software being specified. >

## Background of the System

<*Provide brief description of the similar software in the domain under study and specify how this project is different from the existing software.*

TO DO: 1-2 to paragraphs>

## Objectives of the System

<*This section should include major objective of the software being specified*

TO DO: List down all objectives in bullets form>

## Significance of the System

<*What will be the importance of your software and different application areas where it may play important role.*

TO DO: Provide paragraph or list>

# CHAPTER 2: REQUIREMENT SPECIFICATIONS

<*It is highly recommended that you write brief summary of every chapter as a preamble explaining what a reader would find in this chapter. Thus, when a reader reads summary of your chapter at the beginning, the forthcoming contents shall become clear.*>

## Product Scope

<*This section should include a short description of the software such that boundaries (what to do? and what not to do?) of the software can easily be identified.*

TO DO: 1-2 paragraphs describing the scope of the product.>

## Product Description

### Product Perspective

<*Describe the context and origin of the product being specified. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface.*

TO DO: Provide at least one paragraph describing product perspective. Provide a general diagram that will illustrate how your product interacts with the environment and in what context it is being used.>

### Product Functionality

<*Summarize the major functions, the product must perform or must let the user perform. Details will be provided in Section 2.3.1, so only a high level summary is needed here. Organize the functions to make them understandable to any reader.*

TO DO: Provide a bulleted list of all the major functions of the system. >

### Users and Characteristics

<*Identify the various users that you anticipate will use this product. Users may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience.*

TO DO:

1. Describe the pertinent characteristics of each user. Certain requirements may pertain only to certain users.

3. Distinguish the most important users for this product from those who are less important to satisfy.>

### Operating Environment

<*Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.*

TO DO: Describe the environment your system will have to operate in. Make sure to include the minimum platform requirements for your system. >

## Specific Requirements

### Functional Requirements

< *This section is the direct continuation of section 2.2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions with specific explanations regarding every function.*

TO DO: Break the functional requirements to several functional areas and divide this section into subsections accordingly. Provide a detailed list of all product operations related to these functional areas.

### Behavioral Requirements

< *Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform.*

TO DO: Provide a system level use case diagram which will encapsulate the entire system and all possible actors. Make sure to include a short description of major use-cases, the actors in your diagram, use fully dressed use-cases along with above mentioned use case diagram>

### External Interface Requirements

#### User Interface

<*Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, standard buttons and functions (e.g., Cancel) that will appear on every screen, error message display standards, and so on. Define the software components for which a user interface is needed.*

TO DO: Describe in words the different User Interfaces and the different screens that will be available to the user. >

#### Other Interfaces (if any)

<*may include Hardware Interfaces, Software Interfaces and communication Interfaces*>

<*For hardware interfaces, Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware. You are not required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.*

TO DO: Please provide a short description of the different hardware interfaces. If you will be using some special libraries to communicate with your software mention them here. In case you have more than one hardware interface divide this section into subsections.>

<*For software interfaces, Describe the connections between this product and other specific software components (name and version), including databases, operating systems (Windows? Linux? Etc…), tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.*   
TO DO: To make things simpler, you are only required to describe the specific interface with the operating system or any other software, also specify APIs to be used.>

<*for communication interfaces, Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.*

TO DO: Do not go into too much detail, but provide 1-2 paragraphs were you will outline the major communication standards. For example, if you decide to use encryption there is no need to specify the exact encryption standards, but rather, specify the fact that the data will be encrypted and name what standards you consider using. >

## Non-functional Requirements

<*If anything out of the followings*>

### Performance Requirements

<*If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.*

TO DO: Provide different performance requirements based on the information you collected from the client. For example, you can say “1. Any transaction will not take more than 10 seconds, etc…>

### Safety and Security Requirements

<*Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.*

TODO:

Provide at least different safety requirements based on your interview with the client Describe briefly what level of security is expected from this product by your client and provide a bulleted (or numbered) list of the major security requirements.>

### Software Quality Attributes

<*Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.*

TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Portability, etc…) provide requirements related to the different software quality attributes. Make sure, that you do not just write “This software shall be maintainable…” Indicate how you plan to achieve it, & etc…Do not forget to include such attributes as the design for change. Please note that you need to include quality attributes.>

# CHAPTER 3: DESIGN SPECIFICATIONS

<*It is highly recommended that you write brief summary of every chapter as a preamble explaining what a reader would find in this chapter. Thus, when a reader reads summary of your chapter at the beginning, the forthcoming contents shall become clear.*>

## System Design

<*It identifies the major design constituents of system under study, for instance, composition and modular assembly of systems in terms of subsystems. The UML diagrams used to serve the purpose are “Package Diagram (Logical)”, and “Deployment Diagram (Physical)”.*

TO DO: Package Diagram or Deployment Diagram of the complete system>

## Logical Design

<*The purpose of the Logical design is to elaborate existing and designed types and their implementations as classes and interfaces with their structural static relationships, for instance, static structure (Classes, Interfaces and their relationships). The UML diagram used to serve the purpose is class diagram.*

TO DO: a complete class diagram of software>>

## System Architecture

TO DO: Entity Relationship Diagram (ERD) in case of software having database for information storage>> or system diagram if project is a hardware based one or system model (algorithm and flow design) if project is a machine learning or a deep learning based one. For gaming based projects flow diagram.

## System Interaction and Use Cases

<*The Interaction Viewpoint defines strategies for interaction among objects. This could include designing with concurrent tasks and/or asynchronous messaging, messaging among objects, etc. The UML diagram used to serve the purpose is Sequence Diagram.*

TO DO: Provide sequence diagrams for every use-case >>

## Algorithmic Viewpoint

<*The detailed design description of operations (methods, functions), this applies to components, classes, and individual methods as design entities. The UML diagrams used to serve the purpose are Decision Table, Pseudo Code.*

TO DO: Provide Decision Table or Pseudo Code for the complete software>>

# CHAPTER 4: DEVELOPMENT AND TOOLS

*<Describe overview of importance of tools and techniques used in computer based projects and systems >*

## Introduction

< *Overview of system implementations.*

*<Briefly describe what this chapter includes.*

*TO DO: One paragraph of consisting of 3-5 lines>*>

## Development

### Tools and Technologies Used

*< Describe language used to build the system.*

*Discuss about database used or some other methodology used to data handling*

*Discuss Tools used for graphics design*

*>*

### External APIs/ External Hardware

*< Discuss if additional or some external APIs are used in the system. Provide API requirements and API handling implemented in the system*

*Provide details if some additional hardware is used. Discuss system interaction with the hardware>*

## System Implementation

*< Describe Step by step implementation details.*

*Discuss how algorithm is implemented.*

*Discuss how user interface is implemented and how it is linked with system.*

*Provide details of admin side development and user side development.*

*>*

## Data and Information flow in the system

*< Discuss how data flows between different modules of system*

*How system modules communicate with each other*

*>*

## User Interfaces

## <*Discuss user interface implementations and access details. Mention HCI related concepts. What easiness is provided to the system>*

## Additional Modules and API Provided

*< Discuss if some additional modules are provided or if some API is provided for external world.*

*You can also provide future plan.>*

# CHAPTER 5: QUALITY ASSURANCE

<*It is highly recommended that you write brief summary of every chapter as a preamble explaining what a reader would find in this chapter. Thus, when a reader reads summary of your chapter at the beginning, the forthcoming contents shall become clear.*>

## Introduction

<*Briefly describe what this chapter includes*.

TO DO: One paragraph of consisting of 3-5 lines>

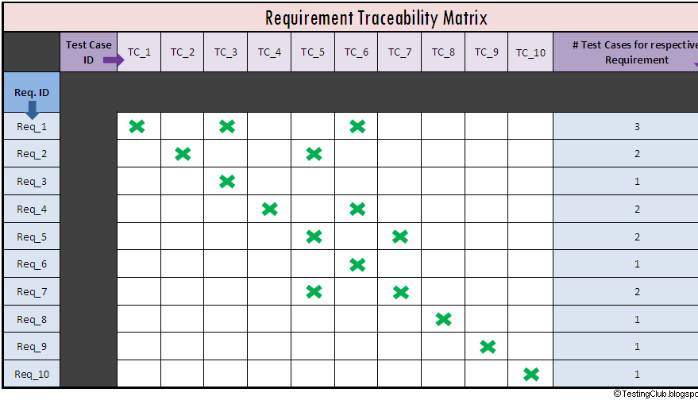
EXAMPLE:

In quality assurance phase which is mainly based on Test plan including testing strategies and types of testing applied to ensure the reliability and accuracy of the application to give the user a great and error free learning experience. Since satisfaction of end user is a first and foremost priority, thus to ensure it, a proper testing mechanism was devised and the results are tabulated in the form of test cases and to trace each test case against desired functional requirement a requirement traceability matrix have been devised which include test case ID against each and every functional requirement desired by user.

## Traceability Matrix

< *The requirement traceability matrix for each test case against functional requirement is to be provided in this section.*

TO DO: Traceability Matrix as per format given below>



## Test Plan

< *Test plan contains the testing mechanism and the entire tests that have been conducted to test the application. This section should include all the test cases conducted for quality assurance of each function requirement.*

TO DO: Heading of each functional requirement and test case in tabulated form as shown in the example below. This table should be repeated for every test case>

EXAMPLE:

Table 5.1: Test case for Application start up

|  |  |
| --- | --- |
| Test ID | ABC-1 |
| Test name | Application start up |
| Date of test | 10/11/2013 |
| Name of application | Kids’ Android Teacher |
| Description | Home screen will be displayed where user will select learning of Urdu or English alphabet letters or view progress record or exit application. |
| Input | Tap on the application icon |
| Expected output | Home screen displayed |
| Actual output | Home screen displayed |
| Test Role (Actor) | Team Member |
| Test verified by | Team Member/Supervisor |