**T.Y. B. Tech.**

**CS 303: Software Engineering Laboratory**

Assignment No: 3

**System Requirement Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| Project Group Information | | | |
| Roll. No. | **Gr. No.** | **Name** | **Roles** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Approved By:**

**Academic Year: 2017-18 Semester: I**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Title** | **Page** |
| 1 | Introduction |  |
| 2 | Terms of Reference |  |
| 3 | Problem Description |  |
| 4 | Functional Hierarchy |  |
| 5 | User Interfaces |  |
| 6 | Hardware Interfaces |  |
| 7 | Software Interfaces |  |
| 8 | Logical Databases |  |
| 9 | Non-Functional Requirements |  |

# INTRODUCTION

*This provides an overview of the entire information described in SRS. This involves purpose and the scope of SRS, which states the functions to be performed by the system. In addition, it describes definitions, abbreviations, and the acronyms used. The references used in SRS provide a list of documents that is referenced in the document.*

|  |  |
| --- | --- |
| Item | Description |
| Purpose |  |
| Audiences |  |
| SRS Scope |  |
| Project Scope |  |

# TERMS OF REFERENCE

*Terms of Reference (TOR) provide a statement of the background, objectives, and purpose of a proposed project. A TOR template includes a range of criteria that are necessary for strategic project decision making. This document defines the activities to be done and indicate the issues, budget and expertise related to the project.*

*Terms of Reference (TOR) are a strategy-level document that defines the tasks and duties required of a project contractor, and highlights project background and objectives at high level. The document also states the planned activities, expected inputs and outputs, project budget, working schedules, and job descriptions. It is used to judge about the performance of contractors, consultants, experts, and other project stakeholders. The purpose of TOR is to specify the amount and type of work to be accomplished by the project. It is a governance document that establishes and determines the relationships between all project stakeholders. Terms of Reference are developed once a project has been identified, defined and planned. TOR of a project provides a clear description of the following critical information:*

* 1. *The rationale behind undertaking the project*
  2. *The proposed methodology of project management along with work plans and activity schedules*
  3. *The expected resource requirements, primarily regarding personnel*
  4. *Reporting rules and requirements*

*The development of Project Terms of Reference is required for making the decision on whether or not to allocate necessary funds to a proposed project. It is the result of the project proposal process, and TOR serves as the primary report of this process. TOR is usually required for:*

* 1. *Pre-feasibility and feasibility analyses*
  2. *Appraisal activity*
  3. *Implementation contracts designing and monitoring*
  4. *Evaluation studies*
  5. *Reporting and audit*
  6. *Other advisory work required at any project stage*

1. *Background*

*The background of a project provides an overview of the history behind the project. It should clearly state why perform the project and refer to a programming context. The purpose is to provide the reader with a brief explanation of the need behind the project.*

*The Background section of a TOR template usually includes several paragraphs which address the following issues:*

* 1. *Describe the project in the context of a related business need*
  2. *State the general role of stakeholders in doing project activities*
  3. *Highlight a brief overview of the project to date*

1. *Objectives*

*The objectives of a project are those desired accomplishments that can be reasonably delivered upon project completion, with consumption of available resources and within an expected timeframe. They should clearly identify and define what is expected from the project and who the target audience is.*

*The Objectives section of a Terms of Reference template should describe desired achievements at different stages of project lifecycle. It should also state the primary objectives of the project, which must be achieved upon success project completion. Here’s an example of how it should look like.*

1. *Issues*

*Any project involves a number of issues and problematic areas that must be addressed in order for the project to be implemented smoothly. The issues are the points of discussion or dispute throughout the project lifecycle. They cover any concern, query, request for change, or anything else that requires a resolution during the project. Unresolved issues may cause project failure.*

*The Issues section of a TOR template should highlight key issues to be studied and disputed at every stage of the project lifecycle. Usually TOR includes a range of evaluation criteria to be used for issue analysis and solving. Here are generic issue evaluation criteria for most projects:*

* 1. *Efficiency – this criterion determines how well a given activity transforms available resources into desired outputs in terms of quantity, quality and time*
  2. *Relevance – it help analyse whether a given activity is being performed with the desired benefits*
  3. *Effectiveness – it concerns how far the project’s outputs have been utilized and whether the project’s purpose has been realized*
  4. *Impact – this measure helps figure out the extent to which the project’s benefits received by the target audience have an overall effect on larger numbers of people concerned*
  5. *Sustainability – this criterion identifies whether the project’s positive outcomes will continue after funding ends.*

1. *Methodology*

*The implementation methodology of a project provides a set of broad principles and rules from which specific procedures will be derived in order to define how to carry out the project in a cost-effective way. It describes the main methods of project implementation.*

*The Methodology section of a Project Terms of Reference template should therefore include a description of the following items:*

* 1. *Key phases of the project implementation process*
  2. *The required level of stakeholder involvement that ensures smooth implementation*
  3. *The content and duration of project activities and tasks*
  4. *The information collection tools to be used throughout the project for monitoring purposes*
  5. *Data analysis rules*

1. *Expertise*

*The expertise needed for doing a project defines a set of professional requirements for the individuals and teams involved in project implementation. It will be the basis for team building, including training and skill assessment.*

*The Expertise section of a Project Terms of Reference template should identify the following:*

* 1. *The type of work involved in the project*
  2. *The type of skills and abilities required to do project work*
  3. *The exact number of individuals involved, including a description of their qualifications, experience, and other professional attributes*
  4. *The period of engagement of each team member*
  5. *A description of the duties and responsibility per teammate*
  6. *The relationship between the team members, including leadership roles*

1. *Reporting*

*Reports provide valued information about project performance over a certain period. Reporting is a process that starts once a project is launched and continues until the project is completed and its product is handed over. Reporting requirements will define how to write and submit project reports and what information to include. The Reporting Requirements section of a Terms of Reference template should clearly specify the requirements for the reporting process, and might include the details of:*

* 1. *Table of contents for project reports/ Rules for composing annexes*
  2. *Report templates/ The language to be used in reports*
  3. *Computer software programmes to be used/ Submission dates*
  4. *People responsible for reporting and approving*

1. *Work Plan*

*A work plan is a kind of strategy that aims to help solve problems throughout a project and boost employee drive and focus. It determines what actions need to be taken to start, implement, and complete the project within a specified time period and under defined budget. It is often used as a general guide for developing a project implementation plan. The Work Plan section of a Project Terms of Reference template should set out the activities and necessary resources required for achieving the project’s results and purpose. It should therefore include a summary of the anticipated work and time schedule, which are based upon the following:*

* 1. *An analysis of the issues, in terms of the evaluation criteria*
  2. *The proposed implementation methodology/ The reporting requirements*

|  |  |
| --- | --- |
| 1. ***Background*** | * *Provide an overview of the history behind the proposed project* * *Clearly state why perform the project, and also refer to a programming context* * *State the general role of stakeholders in doing project activities* * *Write a brief explanation of the need behind the project* |
| 1. ***Objectives*** | * *State the major objectives of the proposed project* * *Describe the intended achievements to be gained at different stages of the project lifecycle* * *Provide an overview of the resources required* * *Clearly identify and define what is expected from the project and who the target audience is* |
| 1. ***Issues*** | * *Highlight the key issues to be studied and disputed at every stage of the project lifecycle* * *List the criteria (including Efficiency, Relevance, Effectiveness, Impact, Sustainability) against which the issues will be analyzed and evaluated* |
| 1. ***Methodology*** | * *Define the key phases of the project implementation process* * *Specify the required level of stakeholder involvement* * *Describe the content and duration of project activities* * *List the information collection tools necessary for monitoring purposes* * *Provide data analysis rules* |
| 1. ***Expertise*** | * *Specify the type of work involved in the proposed project* * *Describe the type of skills and abilities required* * *Define the exact number of individuals involved* * *Point at the period of engagement of each team member* * *Describe the duties and responsibility per teammate* * *Identify the relationships between the team members* |
| 1. ***Reporting*** | * *Provide the Table of Contents for project reports* * *Define rules for composing annexes* * *Add report templates/ Set submission dates* * *List the computer software programmes to be used for report writing* * *Refer to people responsible for reporting and approving* * *Provide other sufficient information such as number of copies to be created, responsibilities for report production and presentation, etc.* |
| 1. ***Work plan*** | * *Provide a summary of the anticipated work* * *Describe the activities and necessary resources required for achieving the project’s results and purpose* * *Provide the activity schedule template* * *Describe The finance resources allocated to the project* |

# PROBLEM DESCRIPTION

*This should include (a) a clear statement that the problem exists, (b) evidence that supports the existence of the problem, (c) evidence of an existing trend that has led to the problem, (d) definitions of major concepts and terms (this can be provided below in a subsection), (e) a clear description of the setting, (f) probable causes related to the problem, and (g) a specific and feasible statement.*

|  |  |
| --- | --- |
| The problem of | (describe the problem) |
| Affects | (the stakeholders affected by the problem). |
| The impact of which is | (what is the impact of the problem). |
| A successful solution would | (list some key benefits of a successful solution). |

|  |  |
| --- | --- |
| For | (target customer) |
| Who | (statement of the need or opportunity) |
| The (product name) | is a (product category) |
| That | (statement of key benefit - that is - compelling reason to buy) |
| Unlike | (primary competitive alternative) |
| Our product | (statement of primary differentiation) |

# FUNCTIONAL HIERARCHY

*Provide a summary of the major functions that the software will perform. The functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time. Textual or graphic methods can be used to show the different functions and their relationships. Such a diagram is not intended to show a design of a product but simply shows the logical relationships among variables.*

|  |  |  |  |
| --- | --- | --- | --- |
| Goal-ID | 1 | Goal Name | Description |
| Objective ID | 1 | Objective Name |  |
| Process ID: 1 |  |
| Process ID: 2 |  |
| Objective ID | 2 | Objective Name |  |
| Process ID: 1 |  |
| Process ID: 2 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Goal-ID | 2 | Goal Name | Description |
| Objective ID | 1 | Objective Name |  |
| Process ID: 1 |  |
| Process ID: 2 |  |
| Objective ID | 2 | Objective Name |  |
| Process ID: 1 |  |
| Process ID: 2 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Goal-ID | 3 | Goal Name | Description |
| Objective ID | 1 | Objective Name |  |
| Process ID: 1 |  |
| Process ID: 2 |  |
| Objective ID | 2 | Objective Name |  |
| Process ID: 1 |  |
| Process ID: 2 |  |

# USER INTERFACES

*5.1 Abbreviated UI, it is the junction between a user and a computer program. An interface is a set of commands or menus through which a user communicates with a program. A command-driven interface is one in which you enter commands. A menu-driven interface is one in which you select command choices from various menus displayed on the screen.*

*The user interface is one of the most important parts of any program because it determines how easily you can make the program do what you want. A powerful program with a poorly designed user interface has little value. Graphical user interfaces (GUIs) that use windows, icons, and pop-up menus have become standard on personal computers.*

*GUI is a program interface that takes advantage of the computer's graphics capabilities to make the program easier to use. Well-designed graphical user interfaces can free the user from learning complex command languages. On the other hand, many users find that they work more effectively with a command-driven interface, especially if they already know the command language.*

*Graphical user interfaces, such as Microsoft Windows and the one used by the Apple Macintosh, feature the following basic components:*

* *Pointer: A symbol that appears on the display screen and that you move to select objects and commands. Usually, the pointer appears as a small angled arrow. Text -processing applications, however, use an I-beam pointer that is shaped like a capital I.*
* *Pointing device: A device, such as a mouse or trackball, that enables you to select objects on the display screen.*
* *Icons: Small pictures that represent commands, files, or windows. By moving the pointer to the icon and pressing a mouse button, you can execute a command or convert the icon into a window. You can also move the icons around the display screen as if they were real objects on your desk.*
* *Desktop: The area on the display screen where icons are grouped is often referred to as the desktop because the icons are intended to represent real objects on a real desktop.*
* *Windows: You can divide the screen into different areas. In each window, you can run a different program or display a different file. You can move windows around the display screen, and change their shape and size at will.*
* *Menus: Most graphical user interfaces let you execute commands by selecting a choice from a menu.*

*In addition to their visual components, graphical user interfaces also make it easier to move data from one application to another. A true GUI includes standard formats for representing text and graphics. Because the formats are well-defined, different programs that run under a common GUI can share data. This makes it possible, for example, to copy a graph created by a spreadsheet program into a document created by a word processor.*

*5.2 Characteristics of Successful User Interfaces*

* ***Clear****: Clarity is the most important element of user interface design. Indeed, the whole purpose of user interface design is to enable people to interact with your system by communicating meaning and function. If people can’t figure out how your application works or where to go on your website they’ll get confused and frustrated.*
* ***Concise****: Clarity in a user interface is great, however, you should be careful not to fall into the trap of over-clarifying. It is easy to add definitions and explanations, but every time you do that you add mass. Your interface grows. Add too many explanations and your users will have to spend too much time reading through them. Keep things clear but also keep things concise. When you can explain a feature in one sentence instead of three, do it. When you can label an item with one word instead of two, do it. Save the valuable time of your users by keeping things concise. Keeping things clear and concise at the same time isn’t easy and takes time and effort to achieve, but the rewards are great.*
* ***Familiar****: Many designers strive to make their interfaces ‘intuitive’. But what does intuitive really mean? It means something that can be naturally and instinctively understood and comprehended. But how can you make something intuitive? You do it by making it ‘familiar’. Familiar is just that: something which appears like something else you’ve encountered before. When you’re familiar with something, you know how it behaves – you know what to expect. Identify things that are familiar to your users and integrate them into your user interface.*
* ***Responsive****: Responsive means a couple of things. First of all, responsive means fast. The interface, if not the software behind it, should work fast. Waiting for things to load and using slaggy and slow interfaces is frustrating. Seeing things load quickly, or at the very least, an interface that loads quickly (even if the content is yet to catch up) improves the user experience. Responsive also means the interface provides some form of feedback. The interface should talk back to the user to inform them about what’s happening. Have you pressed that button successfully? How would you know? The button should display a ‘pressed’ state to give that feedback.*
* ***Consistent****: Consistent interfaces allow users to develop usage patterns – they’ll learn what the different buttons, tabs, icons and other interface elements look like and will recognize them and realize what they do in different contexts. They’ll also learn how certain things work, and will be able to work out how to operate new features quicker, extrapolating from those previous experiences.*
* ***Attractive****: This one may be a little controversial but I believe a good interface should be attractive. Attractive in a sense that it makes the use of that interface enjoyable. Yes, you can make your UI simple, easy to use, efficient and responsive, and it will do its job well – but if you can go that extra step further and make it attractive, then you will make the experience of using that interface truly satisfying. When your software is pleasant to use, your customers or staff will not simply be using it – they’ll look forward to using it. There are of course many different types of software and websites, all produced for different markets and audiences. What looks ‘good’ for any one particular audience will vary. This means that you should fashion the look and feel of your interface for your audience. Also, aesthetics should be used in moderation and to reinforce function. Adding a level of polish to the interface is different to loading it with superfluous eye-candy.*
* ***Efficient****: A user interface is the vehicle that takes you places. Those places are the different functions of the software application or website. A good interface should allow you to perform those functions faster and with less effort. Now, ‘efficient’ sounds like a fairly vague attribute – if you combine all of the other things on this list, surely the interface will end up being efficient? Almost, but not quite. What you really need to do to make an interface efficient is to figure out what exactly the user is trying to achieve, and then let them do exactly that without any fuss. You have to identify how your application should ‘work’ – what functions does it need to have, what are the goals you’re trying to achieve? Implement an interface that lets people easily accomplish what they want instead of simply implementing access to a list of features.*
* ***Forgiving****: Nobody is perfect, and people are bound to make mistakes when using your software or website. How well you can handle those mistakes will be an important indicator of your software’s quality. Don’t punish the user – build a forgiving interface to remedy issues that come up. A forgiving interface is one that can save your users from costly mistakes. For example, if someone deletes an important piece of information, can they easily retrieve it or undo this action? When someone navigates to a broken or non-existent page on your website, what do they see? Are they greeted with a cryptic error or do they get a helpful list of alternative destinations?*

|  |  |  |  |
| --- | --- | --- | --- |
| UI-ID | UI Name | Type | Scope |
| 1 |  | **Input** |  |
| 2 |  | **Forms** |  |
| 3 |  | **Navigation** |  |
| 4 |  | **Command** |  |
| 5 |  | **NL** |  |
| 6 |  | **Menu** |  |
| 7 |  |  |  |

# HARDWARE INTERFACES

*Specify the logical characteristics of each interface between the software product and the hardware components of the system. This includes configuration characteristics. It also covers such matters as what devices are to be supported, how they are to be supported and protocols. This is not a description of hardware requirements in the sense that “This program must run on a Mac with 64M of RAM”. This section is for detailing the actual hardware devices your application will interact with and control. For instance, if you are controlling X10 type home devices, what is the interface to those devices? Designers should be able to look at this and know what hardware they need to worry about in the design. Many business type applications will have no hardware interfaces. If none, just state “The system has no hardware interface requirements” If you just delete sections that are not applicable, then readers do not know if: a. this does not apply or b. you forgot to include the section in the first place.*

|  |  |
| --- | --- |
| Profile | Description |
| Processor |  |
| RAM |  |
| Server Side Technology |  |
| Client Side Technology |  |
| External Devices |  |
|  |  |
|  |  |

# SOFTWARE INTERFACES

|  |  |
| --- | --- |
| Profile | Description |
| Front-end Capabilities |  |
| Back-end Capabilities |  |
| Programming Languages |  |
| Operating Environment |  |
| Software Platform |  |
| Database Servers |  |
| Framework Resources |  |
| API (If Any) |  |
| Other Services/Resources |  |
| Communication Interfaces |  |

# LOGICAL DATABASES

*This section specifies the logical requirements for any information that is to be placed into a database. This may include following parameters:*

*• Types of information used by various functions*

*• Frequency of use*

*• Accessing capabilities*

*• Data entities and their relationships*

*• Integrity constraints*

*• Data retention requirements*

|  |  |  |
| --- | --- | --- |
| Database Name | Parameter | Scope |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# NON-FUNCTIONAL REQUIREMENTS

* *Reliability: Specify the factors required to establish the required reliability of the software system at time of delivery. If you have MTBF requirements, express them here. This doesn’t refer to just having a program that does not crash. This has a specific engineering meaning.*
* *Availability: Specify the factors required to guarantee a defined availability level for the entire system such as checkpoint, recovery, and restart. This is somewhat related to reliability. Some systems run only infrequently on-demand (like MS Word). Some systems have to run 24/7 (like an e-commerce web site). The required availability will greatly impact the design. What are the requirements for system recovery from a failure? “The system shall allow users to restart the application after failure with the loss of at most 12 characters of input”.*
* *Security: Specify the factors that would protect the software from accidental or malicious access, use, modification, destruction, or disclosure. Specific requirements in this area could include the need to:*
  + *Utilize certain cryptographic techniques*
  + *Keep specific log or history data sets*
  + *Assign certain functions to different modules*
  + *Restrict communications between some areas of the program*
  + *Check data integrity for critical variables*
* *Maintainability: Specify attributes of software that relate to the ease of maintenance of the software itself. There may be some requirement for certain modularity, interfaces, complexity, etc. Requirements should not be placed here just because they are thought to be good design practices. If someone else will maintain the system*
* *Portability: Specify attributes of software that relate to the ease of porting the software to other host machines and/or operating systems. This may include:*
  + *Percentage of components with host-dependent code*
  + *Percentage of code that is host dependent*
  + *Use of a proven portable language*
  + *Use of a particular compiler or language subset*
  + *Use of a particular operating system*
* *Correctness - extent to which program satisfies specifications, fulfills user’s mission objectives*
* *Efficiency - amount of computing resources and code required to perform function*
* *Flexibility - effort needed to modify operational program*
* *Interoperability - effort needed to couple one system with another*
* *Reliability - extent to which program performs with required precision*
* *Reusability - extent to which it can be reused in another application*
* *Testability - effort needed to test to ensure performs as intended*
* *Usability - effort required to learn, operate, prepare input, and interpret output*

*Once the relevant characteristics are selected, a subsection should be written for each, explaining the rationale for including this characteristic and how it will be tested and measured. A chart like this might be used to identify the key characteristics (rating them High or Medium), then identifying which are preferred when trading off design or implementation decisions (with the ID of the preferred one indicated in the chart to the right). The chart below is optional (it can be confusing) and is for demonstrating trade-off analysis between different non-functional requirements. H/M/L is the relative priority of that non-functional requirement.*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Characteristic** | **H/M/L** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| 1 | Correctness |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Efficiency |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Flexibility |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Integrity/Security |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Interoperability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Maintainability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Portability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | Reliability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | Reusability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | Testability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | Usability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 | Availability |  |  |  |  |  |  |  |  |  |  |  |  |  |