**(Format of Cover Page and First Inner Page) : B.Tech (Not to be printed)**

**Project Report (font size 16)**

**on (font size 14)**

**Title of Project (font size 18)**

**Submitted as partial fulfillment for the award of (Font Size 14)**

**BACHELOR OF TECHNOLOGY (font 22)**

**DEGREE (font size 20)**

**Session 2017-18 (Font Size 12)**

**in**

**Name of discipline (font size 18)**

**By (font size 14)**

**STUDENT NAME (s) (font size 14)**

**Roll Number (s) (font size 14)**

**Under the guidance of (font size 16)**

**NAME OF SUPERVISOR (font size 14)**

**ABES ENGINEERING COLLEGE, GHAZIABAD (font 16)**

|  |  |  |
| --- | --- | --- |
| Description: C:\Users\es00952\Desktop\Logo.jpg | APJAK logo |  |

**AFFILIATED TO**

**DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, U.P., LUCKNOW**

**(Formerly UPTU)**

**(font size 14)**

**STUDENT’S DECLARATION**

I / We hereby declare that the work being presented in this report entitled “TITLE OF THE PROJECT” (font size 14) is an authentic record of my / our own work carried out under the supervision of Dr. /Mr. /Ms. “SUPERVISOR NAME” (font size 14)

The matter embodied in this report has not been submitted by me / us for the award of any other degree. (font size 12 Arial 1.5 Line Spacing)

**Dated:** **Signature of students(s)**

**(Name(s).......................)**

**Department:**

## This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

|  |  |
| --- | --- |
|  | Signature of Supervisor(Name : )(Designation: )(Computer Science & Engineering Department) |

## 

**TABLE OF CONTENTS**

**CHAPTER 1**

**INTRODUCTION**

It is the first chapter of the Report. The purpose of an introduction in the B. Tech Project Report is to justify the reasons for writing about the report. The goal in this section is to introduce the topic to the reader, provide an overview of previous research on the topic, and identify the own hypothesis. The goals just mentioned could, if abused, lead to write an introduction that is pages and pages long. It can be noted here that the introduction should not contain every bit of detail in the report, and it should not include support for the report. An introduction might, however, include the reasons for supporting the report.

In order for readers to trust the writer, the introduction must be well written with few errors. In order to keep readers reading, the writer needs to catch the attention of the reader and write in an interesting way. The unique language enhancement feature may suggest words to strengthen the writing. Strong writing may hold readers' attention.

In addition to well-written English with strong vocabulary, there are a few other strategies to hold readers' attention. It should be noted that to excite the readers' interest, one may also want to sound as though the readers know the topic that are considered in the report. Some of the following strategies in the bullet-list above may help.

•    To pose a specific question that can invite the readers to keep reading for the answer-- A provocative question works well to engage readers, so long as it doesn't put them off

•    To choose statistics to surprise readers or to go against the common belief about a topic

•    To mention a short, interesting anecdote (or story) related to the topic

•    To provide an interesting (and relevant) quote

•    To develop an unusual or unexpected comparison

It has been observed that the difficulty level to write the Introduction is average and it takes the variable amount of time. Following sub headings are to be included in this chapter depending on the project-

* 1. Problem Introduction
     1. Motivation
     2. Project Objective
     3. Scope of the Project
  2. Related Previous Work

It briefly includes previous work carried out in this field, researching the problem studied, summarization of the results obtained etc

* 1. Organization of the Report.

It provides the short description of the work reported in each chapter.

**CHAPTER 2**

**(For non-web based projects)**

**LITERATURE SURVEY**

**(LITERATURE SURVEY FORMAT FOR NON-WEB BASED PROJECTS)**

* A brief description of what is studied in the survey should be given in a paragraph of 6-10 lines (approx.) this paragraph should not contain any title.
* Then you can elaborate on each technique/algorithm/ technology depending on your project section wise. For example if literature survey is for face recognition then all existing approaches should be mentioned in a separate section.
* Each section should contain a heading and should be numbered. It can also contain images, formulae, flow chart and tabular data that are needed for explanation. Each section size should not exceed 1 to 1.5 pages.
* Use citations from the References section where ever required-

**Example-** A **pun**, or **paronomasia**, is a form of word play that deliberately exploits ambiguity between similar-sounding words for humorous or rhetorical effect. **[1]** (If the contents are taken from reference 1 of References section).

**All citations must be referred.**

* Any included image, formula, flow chart or tabular data should be numbered and should be referred in the respective section. Tables numbering should appear on the top of table and figure numbering should be below the figure.
* At the end of the literature survey chapter, include a summary section with the heading SUMMARY. In this section sum up the above studied techniques/algorithms/ technologies paragraph wise.

**CHAPTER 2**

# ( Web Based Projects)

# SOFTWARE REQUIREMENT SPECIFICATION

*Describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in section 3, and makes them easier to understand*. *In a sense, this section tells the requirements in plain English for the consumption of the customer. Section3 will contain a specification written for the developers.*

2.1 Product Perspective

*Put the product into perspective with other related products. If the product is independent and totally self-contained, it should be so stated here. If the SRS defines a product that is a component of a larger system, as frequently occurs, then this subsection relates the requirements of the larger system to functionality of the software and identifies interfaces between that system and the software. If you are building a real system, compare its similarity and differences to other systems in the marketplace. If you are doing a research-oriented project, what related research compares to the system you are planning to build.*

*A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful. This is not a design or architecture picture. It is more to provide context, especially if your system will interact with external actors. The system you are building should be shown as a black box. Let the design document present the internals.*

*The following subsections describe how the software operates inside various constraints*.

### 2.1.1 System Interfaces

*List each system interface and identify the functionality of the software to accomplish the system requirement and the interface description to match the system. These are external systems that you have to interact with. For instance, if you are building a business application that interfaces with the existing employee payroll system, what is the API to that system that designer’s will need to use?*

### 2.1.2 Interfaces

*Specify:*

1. *The logical characteristics of each interface between the software product and its users.*
2. *All the aspects of optimizing the interface with the person who must use the system*

*This is a description of how the system will interact with its users. Is there a GUI, a command line or some other type of interface? Are there special interface requirements? If you are designing for the general student population for instance, what is the impact of ADA (American with Disabilities Act) on your interface?*

### 2.1.3 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.*

### 2.1.4 Software Interfaces

*Specify the use of other required software products and interfaces with other application systems. For each required software product, include:*

1. *Name*
2. *Mnemonic*
3. *Specification number*
4. *Version number*
5. *Source*

*For each interface, provide:*

1. *Discussion of the purpose of the interfacing software as related to this software product*
2. *Definition of the interface in terms of message content and format*

*Here we document the APIs, versions of software that we do not have to write, but that our system has to use. For instance if your customer uses SQL Server 7 and you are required to use that, then you need to specify i.e.*

*2.1.4.1 Microsoft SQL Server 7. The system must use SQL Server as its database component. Communication with the DB is through ODBC connections. The system must provide SQL data table definintions to be provided to the company DBA for setup.*

*A key point to remember is that you do NOT want to specify software here that you think would be good to use. This is only for* ***customer-specified systems*** *that you* ***have*** *to interact with. Choosing SQL Server 7 as a DB without a customer requirement is a Design choice, not a requirement. This is a subtle but important point to writing good requirements and not over-constraining the design.*

### 2.1.5 Communications Interfaces

*Specify the various interfaces to communications such as local network protocols, etc. These are protocols you will need to directly interact with. If you happen to use web services transparently to your application then do not list it here. If you are using a custom protocol to communicate between systems, then document that protocol here so designers know what to design. If it is a standard protocol, you can reference an existing document or RFC.*

### 2.1.6 Memory Constraints

*Specify any applicable characteristics and limits on primary and secondary memory*. *Don’t just make up something here. If all the customer’s machines have only 128K of RAM, then your target design has got to come in under 128K so there is an actual requirement. You could also cite market research here for shrink-wrap type applications “Focus groups have determined that our target market has between 256-512M of RAM, therefore the design footprint should not exceed 256M.” If there are no memory constraints, so state.*

### 2.1.7 Operations

*Specify the normal and special operations required by the user such as:*

1. *The various modes of operations in the user organization*
2. *Periods of interactive operations and periods of unattended operations*
3. *Data processing support functions*
4. *Backup and recovery operations*

*(Note: This is sometimes specified as part of the User Interfaces section.) If you separate this from the UI stuff earlier, then cover business process type stuff that would impact the design. For instance, if the company brings all their systems down at midnight for data backup that might impact the design. These are all the work tasks that impact the design of an application, but which might not be located in software.*

### 2.1.8 Site Adaptation Requirements

*In this section:*

1. *Define the requirements for any data or initialization sequences that are specific to a given site, mission, or operational mode*
2. *Specify the site or mission-related features that should be modified to adapt the software to a particular installation*

*If any modifications to the customer’s work area would be required by your system, then document that here. For instance, “A 100Kw backup generator and 10000 BTU air conditioning system must be installed at the user site prior to software installation”.*

*This could also be software-specific like, “New data tables created for this system must be installed on the company’s existing DB server and populated prior to system activation.” Any equipment the customer would need to buy or any software setup that needs to be done so that your system will install and operate correctly should be documented here.*

## 2.2 Product Functions

*Provide a summary of the major functions that the software will perform. Sometimes the function summary that is necessary for this part can be taken directly from the section of the higher-level specification (if one exists) that allocates particular functions to the software product.*

*For clarity:*

1. *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.*
2. *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.*

*AH, Finally the real meat of section 2. This describes the functionality of the system in the language of the customer. What specifically does the system that will be designed have to do? Drawings are good, but remember this is a description of what the system needs to do, not how you are going to build it. (That comes in the design document).*

## 2.3 User Characteristics

*Describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise. Do not state specific requirements but rather provide the reasons why certain specific requirements are later specified in section 3.*

*What is it about your potential user base that will impact the design? Their experience and comfort with technology will drive UI design. Other characteristics might actually influence internal design of the system.*

## 2.4 Constraints

*Provide a general description of any other items that will limit the developer's options. These can include:*

*(1) Regulatory policies*

*(2) Hardware limitations (for example, signal timing requirements)*

*(3) Interface to other applications*

*(4) Parallel operation*

*(5) Audit functions*

*(6) Control functions*

*(7) Higher-order language requirements*

1. *Signal handshake protocols (for example, XON-XOFF, ACK-NACK)*
2. *Reliability requirements*

*(10) Criticality of the application*

*(11) Safety and security considerations*

*This section captures non-functional requirements in the customers language. A more formal presentation of these will occur in section 3.*

## 2.5 Assumptions and Dependencies

*List each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system would be available on the hardware designated for the software product. If, in fact, the operating system were not available, the SRS would then have to change accordingly.*

*This section is catch-all for everything else that might influence the design of the system and that did not fit in any of the categories above.*

## 2.6 Apportioning of Requirements.

*Identify requirements that may be delayed until future versions of the system. After you look at the project plan and hours available, you may realize that you just cannot get everything done. This section divides the requirements into different sections for development and delivery. Remember to check with the customer – they should prioritize the requirements and decide what does and does not get done. This can also be useful if you are using an iterative life cycle model to specify which requirements will map to which interation.*

* 1. **Use case**

# Use case Model

* Some Guide Lines for use cases
* Place Your Primary Actor(S) In The Top-Left Corner Of The Diagram
* Draw Actors To The Outside Of A Use Case Diagram
* Name Actors With Singular, Business-Relevant Nouns
* Associate Each Actor With One Or More Use Cases
* Actors Model Roles, Not Positions
* Use <<system>> to Indicate System Actors
* Actors Don’t Interact With One Another
* Introduce an Actor Called “Time” to Initiate Scheduled Events
* Associations are depicted as lines connecting two modeling elements with an optional open-headed arrowhead on one end of the line indicating the direction of the initial invocation of the relationship. Generalizations are depicted as a close-headed arrow with the arrow pointing towards the more general modeling element.

**2.7.2 Use Case Diagram (you can use either use case diagram or scenario)**

**2.7.3 Use Case Scenario (Following details can be provided for a use case scenario)**

|  |  |
| --- | --- |
| Use Case Element | Description |
| Use Case Number | ID to represent your use case |
| Application | What system or application does this pertain to |
| Use Case Name | The name of your use case, keep it short and sweet |
| Use Case Description | Elaborate more on the name, in paragraph form. |
| Primary Actor | Who is the main actor that this use case represents |
| Precondition | What preconditions must be met before this use case can start |
| Trigger | What event triggers this use case |
| Basic Flow | The basic flow should be the events of the use case when everything is perfect; there are no errors, no exceptions. This is the "happy day scenario". The exceptions will be handled in the "Alternate Flows" section. |
| Alternate Flows | The most significant alternatives and exceptions |

# Sequence diagrams ( Example for Registration Process)

**CHAPTER 3**

**(For non-web based projects)**

**SYSTEM DESIGN AND METHODOLOGY**

* 1. **System Design**
     1. **System Architecture /Diagrammatical View**
     2. **DFD, Class Diagram, flow charts, ER Diagrams (which ever applicable depending on the project)**
  2. **Algorithm(s)**

# (if required, add any other section applicable for the methods and approaches you have followed)

# CHAPTER 3

**(For Web-Based Projects)**

# SYSTEM DESIGN

System Design should include the following sections (Refer each figure or table in some text). Figure number should be provided below the figure and the table numbering should be provided above the table.

# Architecture diagrams

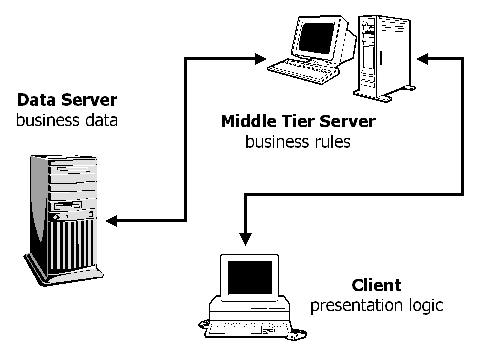


Figure 3.1 3-Tier Architecture Diagram example

# Class diagrams

* 1. **Data Flow Diagram**
  2. **Activity Diagram (Example for Registration and Login)**

# ER Diagrams

# Database schema diagrams

**References**

Thumb rules followed to refer some one’s work are given below.

1. Cite all ideas, concepts, text, data that are not own by the project group
2. If author makes a statement, he must back it up with his own data or a reference
3. All references cited in the text must be listed
4. List all references cited in the text in alphabetical
5. Follow the format or citation style as discussed in chapter 4.

**(Example of References using the Numeric System)**

**Examples of Journal Article referencing:**

1. Drucker, D. C., "Photoelastic Separation of Principal Stresses by Oblique Incidence", *Journal of Applied Mechanics*, Vol. 65, pp. 156‑160, 1943.

2. Maiers, J., and Sherif, Y. S. , "Application of Fuzzy Set Theory," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC‑15, No.1, pp. 41‑48, 1985.

**Example of Book referencing:**

3. Doe, N., *Control System Principles*, New York: John Wiley, 1999.

**Example of Referencing of an Article in a Book:**

4. Hwang, C. J., "Rule‑based Process Control," in E. Kumarmangalam and L. A. Zadeh (Eds.), *Approximate Reasoning in Intelligent Systems, Decision and Control*, pp. 145‑158, Oxford: Pergamon Press, 1987.

**Example of referencing of a B. Tech. Report:**

5. Nayak, T., "Application of Neural Networks to Nuclear Reactors," M.Sc. Report, U.P. Technical University, 2005.

**Example of referencing of a Ph. D. Dissertation:**

6. Muskìn, H. L., "Development of A Knowledge‑Based System for a Nuclear Power Plant," Ph.D. Dissertation, U. P. Technical University, 2003.

**Example of referencing of a Conference Paper :**

7. Lokhande, R., Arya, K. V., and Gupta, P., "Identification of Parameters and Restoration of Motion Blurred Images", *Proceedings of the 2006 ACM Symposium on Applied Computing (SAC 2006)*, pp. 89‑95, Dijon, France, April 2- 7, 2006.

**Example of referencing of a Paper presented at Conference but not Published :**

8. Lokhande, R., and Gupta, P., "Identification of Parameters of Motion Images", presented at 5th International Conference on Cyber Systems, New Delhi, India, April 12- 17, 2004

**Example of referencing of a Report [Technical, Internal, or Memoranda]: :**

9. Das, A. R., Murthy D., and Badrinath J., A Comparison of Different Biometrics Traits, RSRE Memorandum No. 4157, RSRE Malvern, 2001.

**Example of referencing of a Manual**

10. Bell Telephone Laboratories Technical Staff, Transmission System for Communications, Bell Telephone Laboratories, 1995.

**Example of referencing of a Class Note**

11. "Signal integrity and interconnects for high-speed applications," class notes for ECE 497- JS, Department of Electrical and Computer Engineering, University of Illinois at Urbana- Champaign, Winter 1997.

**Example of referencing of a Private Communication**

12. Banerjee, T., (Private Communication), 1998

**Example of referencing of an Article from Internet**

13. Biometrics Group, Indian Institute of Technology Kanpur, "Multimodal Biometrics System," December 2006, http://www.cse.iitk.ac.in/users/biometrics.html

14. Gupta, P. (pg@iitk.ac.in), "Biometrics System," Usenet post to sci.electronics.design, July 4, 2007.

**Example of referencing of an Article from Catalog**

15. Catalog No. MWM-1, Microwave Components, M. W. Microwave Corp., Brooklyn, NY

**Example of referencing of an Article from Application Note**

16. Hewlett-Packard, Appl. Note 935, pp. 25-29.

**Example of referencing of an Article from Application Note**

1. Kar, K. and Majumder, D., "Fuzzy Controller Component," U. S. Patent 23,160,040, December 21, 2006.

**Figures and Tables**:

Figures should be oriented vertically, in portrait mode, wherever possible. If they must be oriented horizontally, in landscape mode, so that one can read them from the right, not from the left, where the binding will be. Examples are given below.



Figure 2.2. A typical neural network.

TABLE 9.12. Comparison Of Various Data Structures.

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Sequential List | Linked List | AVL-Tree |
| Search | O(log n) | O(n) | O(log n) |
| Delete | O(n) | O(1) | O(log n) |
| Insert | O(n) | O(1) | O(log n) |