**TABLE OF CONTENTS**

# 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Title page** | | | |  |
| **Certificate page** | | | |  |
| **Sponsorship letter if applicable** | | | |  |
| **Acknowledgement** | | | |  |
| **Abstract** | | | |  |
| **List of Figures** | | | |  |
| **List of Tables** | | | |  |
| **Abbreviation** | | | |  |
| **ACM Keyword** | | | |  |
| **1.** |  | **INTRODUCTION** (12, bold, Upper Case) | | Page no. |
|  | 1.1 | Background and basics <<Cover background and Fundamentals>> | |  |
|  | 1.2 | Literature Survey << ( Summarized existing approaches to solve the problem)>> | |  |
|  | 1.3 | Project Undertaken | |  |
|  |  | 1.3.1 | Problem definition |  |
|  |  | 1.3.2 | Scope Statement |  |
|  | 1.4 | Organization Of Project Report | |  |
| **2.** | **PROJECT PLANNING AND MANAGEMENT** (12, bold, Uppercase) | | |  |
|  | **2.1 Introduction**  <<This chapter covers the project planning and management details. It also covers System Requirement specifications. SRS is considered as the base for the effort estimations and project scheduling>>  <<Above line should be there in the Introduction section of this chapter>> | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 2.2 | System Requirement Specification (SRS)  (Scope should not be repeated. Functional, Non-functional, H/w, s/w, interface and other requirements should be elaborated as per SRS template.)   * 1. Detail System Requirement Specification (SRS)   /\*\*\*\*\*\*Below part should not be listed in the INDEX. It is for your information\*\*\*\*\*\*\*\*/  /\*\*\*(Introduction, Scope statement, reading suggestions part should not be there in this section as it is covered in first chapter\*\*\*\*\*/  /\*\*\*\*\*\* Below part should be included\*\*\*/  Overview of system should be given i.e. mention about the nature of the system. E.g. if it is web based system, stand- alone system, or if is going to be part of some other bigger system.  So **2.1.1:** will be **System Overview (This line should be listed in index)**  **2.1.2:** **Functional Requirements (This line should be listed in index)**  After overview, all major functional requirements **MUST be Listed and ELABORATED as follows in this chapter**.  E.g.  System Featture1 :  <<stepwise Flow of user’s interaction for System feature1>>  System Featture2 :  <<stepwise Flow of user’s interaction for System feature2>>  E.g. **Withdraw Money 🡨**(Name of functional Requirement)  **<< Below is Flow user’s interaction for above system feature i.e. functional requirement>>**  **Main Flow:**  User shall insert the card.  System shall validate the card. If card id authorized and not damaged, then system shall prompt user for PIN.  User shall input the PIN.  System shall validate the PIN. If PIN is correct, System shall show options for transactions.  User shall select “Withdraw” option.  System shall prompt for entering amount.  User shall enter the amount.  System shall check the balance. If sufficient balance is there then cash shall be dispensed.  **Exceptional Flow:**  **-**If card is damaged then system shall display the message.  - If incorrect PIN is entered system shall display the message “Incorrect PIN”  - If sufficient balance is not there, then system shall display the message “Insufficient Funds”.  **<< This is how stepwise NORMAL flow of interaction and** **EXCEPTIONAL flow should be written for every system feature i.e. functional requirement.**)  **2.1.3: Non- Functional Requirements** **(This line should be listed in index)**  For non functional requirements listed under quality attributes section, you have to check which one are applicable to your concept/project.  **2.1.4 : Deployment Environment:** **(This line should be listed in index)**  After non functional reuirements, deployment environment i.e. hardware and software requirement should listed.  **2.1.5 : External Interface Requirements:** **(This line should be listed in index)**  After this all external interface requirements should be described. Do check if all interface requirements are applicable to your project.  \*\*\*\*\*/  **2.1.6 : Other Requirements:** **(if applicable , this line should be listed in index)** |  |
|  | 2.2 | Project Process Modeling  << Mention which process model is followed. Waterfall/ Incremental/ Evolutionary/Agile. Describe about the model you have chosen for your project. You should have at least theoretical knowledge of all above models as can be asked in oral >>> |  |
|  | 2.3 | Cost & Efforts Estimates << (Mandatory)>  <<Write about the model used for effort estimation. Effort MUST be estimated according to any effort estimation model/tool>>  <<(Function Point analysis or LOC Analysis).>> |  |
|  | 2.4 | Project Scheduling <<(Time line chart)>> |  |
|  |  |  |  |
| **3.** | **ANALYSIS & DESIGN** (12, bold, Uppercase) | |  |
|  | **3.1** Introduction  <<This chapter covers the analysis and design of the considered system>>  <<Above line should be there in the Introduction section of this chapter>> | |  |
|  | **3.2** | IDEA matrix |  |
|  | **3.3** | Mathematical Model Model (Mention : Algorithm and Methodology and then listing of all the identified sets) |  |
|  | **3.4** | Feasibility Analysis (NP Completeness Analysis) |  |
|  | **3.5** | Architecture Diagram (Overview diagram of the system, covering all the modules of the system) |  |
|  | **3.6** | UML diagrams  3.6.1.Use-Case Diagrams (if applicable)  <<Write 2-3 lines about every diagram. E.g. Use-Case diagram represents the functional requirements of the system. This use case diagram covers following functional requirements:   1. Requirement1 2. Requirement2 so on….   3.6.2 Activity Diagram  3.6.3 Class Diagrams  << Write in brief responsibility of every class i.e. what tasks that class performs>>  3.6.4 ER Diagrams (if applicable)  3.6.5 Sequence Diagrams Or ( DFDs if applicable)  3.6.6 Component/Interface diagram  (Interface diagram : showing how modules are communicating with other modules. Hint. Provided and Required interface should be should shown. You can refer book for UML diagrams)  3.6.7 State Machine Diagrams  3.6.8 Deployment Diagrams |  |
| **4.** | **IMPLEMENTION & CODING** (12, bold, Uppercase) | |  |
|  | 4.1 | Introduction  <<This chapter covers the role of various subsystems/modules/classes along with implementation details listing of the code for the major functionalities>>  <<Above line should be there in the Introduction section of this chapter>> |  |
|  | 4.2 | Database schema (If Applicable )  <<List the table along with column names.  Explain purpose of every table>> |  |
|  | 4.3 | 4.3.1 Operational Details  <<(module-wise description).  (E.g. This module covers these functionalities. It consists of the following classes.)>>  4.3.2 Major classes  <<(List the major classes from each module with their responsibilities)  (E.g. DatabaseManager class :  It deals with all the database related queries for insertion, deletion, modification and retrieval of the records.)>>  4.3.3 Code Listing  <<Sample code for the major functional requirements. Code should be written inside rectangles like how we find code listing in text books>> |  |
|  | 4.4 | Screen shots  << (For important/major functionalities); According to flow of the system>> |  |
| **5.** | **TESTING** (12 Bold, Uppercase. Give actual test cases) | |  |
|  |  | |  |
|  | 5.1 | Introduction  <<This chapter covers the testing approach used and the test cases>>  <<Above line should be there in the Introduction section of this chapter>> |  |
|  | 5.2 | Unit Testing  <<Write two-three lines giving definition of Unit Testing.  List Unit test cases for every module in tabular form.  Include positive, negative test cases for every module. Also include test cases for boundary conditions, for performance testing etc>>  **Format for Test cases in tabular form is as follows:**  Test case-id Brief description Input Expected result Actual output |  |
|  | 5.3 | Integration Testing  <<Write two-three lines giving definition of Integration Testing>>  <<One test case per module when it is to be integrated with other module>> |  |
|  | 5.4 | Acceptance Testing  <<Write two-three lines giving definition of Acceptance Testing  (One test case per functional requirement and one test case per non-functional requirement)>> |  |
| **6.** | **RESULTS & DISCUSSIONs** (12, bold, Upper Case) | |  |
|  | 6.1  6.2  6.3 | Main GUI snapshots <<(if applicable) showing the results.>>  Table of the results/findings or graphs <<(if applicable)>>  Discussions |  |
| 7. | **CONCLUSION** (12, bold, Upper Case) | |  |
| 8. | **FUTURE WORK (if applicable)** | |  |
| **References** | | |  |
| **Appendices** | | |  |

**Hide all these table borders before printing**