Software Engineering and

project management

18csc206j

**Presented by:**

1)Akash Prabu.P (RA2011026020072)

2)Bharathwaj.M (RA2011026020065)

3)Nikhil Ravichandran (RA2011026020102)

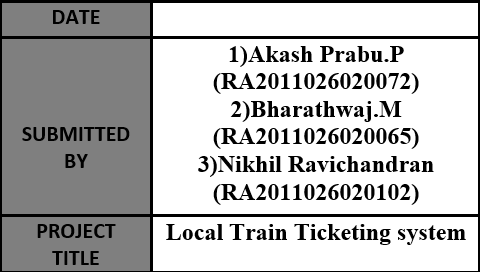
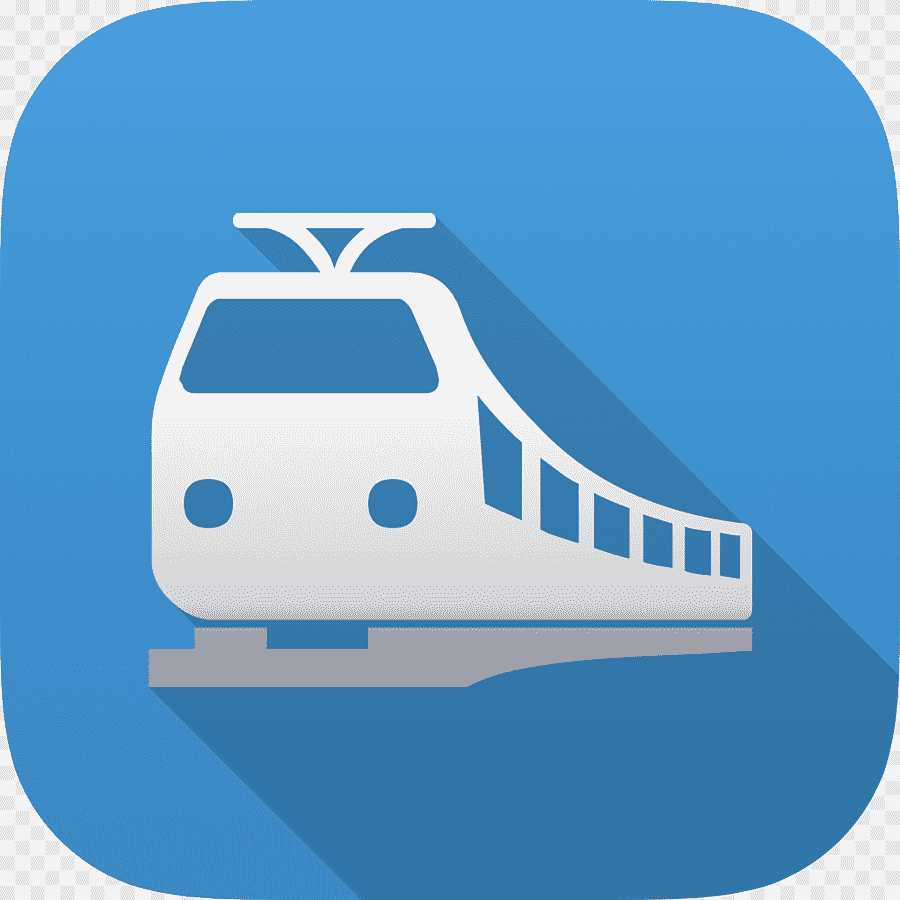
TABLE OF CONTENT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **DATE** | **CONTENTS** | **PAGE NO** | **SIGNATURE** |
| **1** |  | **PROBLEM STATEMENTS**   * + 1. Identify the software project problem statement     2. Create Business Class Template |  |  |
| **2** |  | **STACKHOLDERS & PROCESS MODELS**   * + 1. Identifying Stakeholders     2. User story     3. Identify the Appropriate Process Modules     4. Arrive at Problem Statement     5. Comparison between Agile and Waterfall Model |  |  |
| **3** |  | **IDENTIFYING THE REQUIREMENTS FROM THE PROBLEM STATEMENT**   * + 1. Requirements     2. Functional Requirements     3. Non-Functional Requirements     4. Hardware Requirements     5. Software Requirements |  |  |
| **4** |  | **PROJECT PLAN AND PROJECT EFFORT BASED ON RESOURCES**   * + 1. Project Plan     2. Identifying Job Roles and Responsibility |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **5** |  | **PROJECT EFFORT BASED ON RESOURCES**   * + 1. Work Breakdown structure     2. Risk Management |  |  |
| **6** |  | **ESTIMATION OF PROJECT METRICS**   * + 1. Function Point Analysis     2. COCOMO Model |  |  |
| **7** |  | **DESIGN**  7.1.1 System Architecture |  |  |
| **8** |  | **MODELLING UML USE CASE DIAGRAMS & CAPTURING USE CASE SCENARIOS**   * + 1. Use Case Diagram Description     2. Use Case Diagram |  |  |
| **9** |  | **ER MODELLING FROM THE PROBLEM STATEMENTS**   * + 1. ER Diagram Description     2. ER Diagram |  |  |
| **10** |  | **IDENTIFYING DOMAIN CLASSES FROM THE PROBLEM STATEMENTS**   * + 1. Domain Class Diagram Description     2. Domain Class Diagram |  |  |
| **11** |  | **STATECHART AND COMMUNICATION MODELLING**   * + 1. State Chart Diagram Description     2. State Chart Diagram     3. Communication Diagram Description     4. Communication Diagram |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **12** |  | **MODELLING UML CLASS DIAGRAM & SEQUENCE DIAGRAMS**   * + 1. Class Diagram Description     2. Class Diagram     3. Sequence Diagram Description     4. Sequence Diagram |  |  |
| **13** |  | **MODELLING DATA FLOW DIAGRAM**   * + 1. Data Flow Diagram (DFD) Description     2. Data Flow Diagram (DFD) |  |  |
| **14** |  | **IMPLEMENTATION**  14.1.1 Module Implementation |  |  |
| **15** |  | **ESTIMATION OF TEST COVERAGE METRICS & STRUCTURAL COMPLEXITY**   * + 1. Control Flow Graph     2. McCabe’s Cyclomatic Complexity     3. Optimum Value of Cyclomatic Complexity |  |  |
| **16** |  | **DESIGNING TEST SUITES**   * + 1. Software Testing     2. Testing Frameworks     3. Master Test Plan     4. Manual Testing |  |  |
| **17** |  | **CONCLUSION** |  |  |
| **18** |  | **REFERENCES** |  |  |

# PROBLEM STATEMENTS

## PROBLEM STATEMENT

##### INTRODUCTION:

A local train ticketing system project for local trains that allows users to book local train tickets and get ticket receipt online. This local train project provides login rights for normal users and admin. A normal user may login and get a ticket online, print it and travel by train. The ticketing process consists of a ticket booking form. The form allows the user to choose his source and destination. The source is the station from where the user will be boarding the train. Destination is the station he needs to get down at. The project database is already filled with stations on Mumbai western central and harbor line. It can be modified for any other city as needed. The system also consists of an option to select weather ticket is a single journey or a return ticket and the journey will be commenced on first class or a second class. It also consists of an admin account. The admin may recharge user account balance and check for various journey tickets processed in the system.

##### PROBLEMS:

The following problems were faced during the manual process of Train ticketing system:

* **TEDIOUS PROCESS**: The existing system is highly manual involving a lot of paper work and calculation and therefore may be erroneous. Chances of error and workload on each individuals is more.

* **TIME CONSUMPTION**: Takes a lot of time when it is done manually. Search for particular data also leads to unnecessary delays.

* **DATABASE MANAGEMENT**: Due to the increasing number of applicants, the management and the processing of data are difficult to update and maintain with time.

* **SECURITY:** More accessibility to data, which made tampering much easier.

##### PROPOSAL:

The Scope of the project is to minimize the booking time at ticket counters so that the android application allows the customers to pay from mobile itself. In practice, the application can be used widely to enhance the quick booking of tickets using wallet

## ONE PAGE BUSINESS CASE TEMPLATE

##### THE PROJECT:

* + - * Since, the number of customers using trains is gradually increasing every year, manual process becomes very much complicated.
      * The development of this **“Local Train Ticketing system”** makes the entire process automated keeping in mind the view of database integration approach.
      * Authentication is provided where only the registered users can access.

##### THE HISTORY:

* + - * The manual system does not provide secured registration and profile management of the Applicants properly**.**
      * In manual system, all details are taken in the form of documents, which can be misplaced anytime.
      * Every individual had to stand in queue, which is more time consuming and involves more manpower.

##### CONSTRAINTS:

 The Applicants require a computer to submit their information.

 Prior knowledge of computers and English language should be known

 Although the security is given high importance, there is always a chance of intrusion in the web world which requires constant monitoring.

 The Applicant must be careful while submitting the information.

##### APPROACH:

* + - * + We approach this solution from the technical point of view. So, we are keen in maintaining an easy way to execute interface for the applicant as well as the back-end maintenance.
        + The development of this project will require front end and back-end programming knowledge. Thus, it is important to equip with the same.

##### BENEFITS:

* Security is provided.
* Less time consumption.
* Improves the efficiency.
* Reduces the complexities.
* Elimination of human error.