Project
Report

Investigation Bureau

Bachelor of Technology
in
Computer Engineering

Submitted by

Roll No	Names of Students
151071016	Priya Nayak
151071033	Saumya Gangwar
151071043	Kausha Nisar
151071046	Dimple More

Under the guidance of
Prof.Pramila .M. Chawan

Department of Computer Engineering
VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE
Mumbai, Maharashtra, India – 400 031, Odd Semester 2017

Certificate

This is to certify that Ms. Priya Nayak, Ms. Saumya Gangwar, Ms. Kausha Nisar and Ms. Dimple More , students of Third year B.Tech (Computer Engineering), VJTI , Matunga ,Mumbai / 400031 have successfully completed the project “Investigation Bureau” under the guidance of Prof P.M. Chavan as a part of Software Project Management term work .

Prof. P.M.Chawan
Computer Engineering,VJTI

Date:

Statement by Candidate

We wish to state that work embodied in this special topic titled “Software Engineering” forms my own contribution to the work carried out under the guidance of Prof. P.M.Chawan at the ‘Veermata Jijabai Technological Instiute’, Matunga, Mumbai – 31. This work has not been submitted for any other Degree or Diploma of any University / Institute. Wherever references have been made to previous works of others, it has been clearly indicated.

Ms. Priya Nayak
Ms. Saumya Gangwar
Ms. Kausha Nisar
Ms. Dimple More

Contents

List of Figures

List of Tables

Chapter 1

Problem Statement

To develop a software to implement an Investigation Bureau.

1.1 Problem Elaboration

Investigation Bureau Management is designed to help police officers, lawyers, detectives, general public as well as the Judiciary branch of our Government to order, monitor, append, plead, analyze cases. It shall perform the following functions:

1. Analyze pending cases
2. Assign officers for investigation of pending cases
3. Keep track of the leads and way worked into any case
4. Keep records of all solved cases
5. Train new trainees
6. Keep in touch with lawyers regarding the cases
7. Forensics Department

1.2 Scoping

Functions identified:

1. Registration
2. Assign detectives

3. Record Leads in cases
4. Train new recruits
5. Inform lawyers related to cases
6. Interact with Forensics Department

1.3 Hardware and Software Requirements

Hardware Requirements: To be able to run the system, the minimum requirements of the hardware for this system are:

- CPU (desktops)
- 2 GB RAM
- HDD 60 GB min ,7200 RPM6 GB
- The hardware used must have a competent firewall to secure the data in the system

Software Requirements:

- The system was developed to serve as a database for the events' organizers.It is a stand/alone system; hence, it does not need an internet connection.
- However, the system requires minimum specifications for the software interfaces to be able to use it efficiently. The operating system (OS) required in order to use the system is at minimum Windows XP, but may also be Windows Vista, or Windows 7 must also be installed to their devices.
- These application software were used to make the database, thus, having them in the computers will make the system proceed successfully and run error/free.

1.4 Software process model

A software life cycle model (also called process model) is a descriptive and diagrammatic representation of the software life cycle.

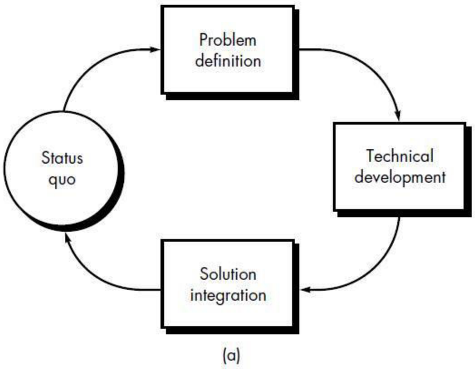


Figure 1.1: Software Process Model

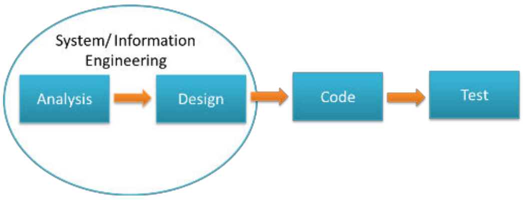


Figure 1.2: Linear Sequential Model

1.4.1 Linear Sequential Model

It is also called "Classic Life Cycle" or "Waterfall" model or "Software Life Cycle" suggests a systematic and sequential approach to software development that begins at the system level and progresses through analysis, design, coding, testing and support.

1.4.2 Advantages

1. Simple and a desirable approach when the requirements are clear and well understood at the beginning

2. This methodology is significantly better than the haphazard approach to develop software.It provides a template into which methods of analysis, design, coding, testing and maintenance can be placed.

1.4.3 Disadvantages

1. It is difficult for the customers to state the requirements clearly at the beginning. There is always certain degree of natural uncertainty at beginning of each project.

Chapter 2

Project Scope

2.1 Major Sofware Function

1. Log In/Sign Up

   This page is basically for Officers of that investigation Bureau to log in and update their work, like the progess on any case, closure of any case,etc. This page also gives newly joint officers to create their own account, with the access and privileges given by the administrator, after verifying.

2. Criminal Record

   This page is used for finding history of a person on whom a case has been registered or is suspicious. This page will ask for name or if not name the officer can upload the image for searching the identity.

3. Evidence Record

   This page will ask for case number, after providing it, a list of all evidences collected(in the form of photos, messages, phones taped, etc) will be displayed with the progress of the case.

4. Case History

   This page will ask for case number and accordingly the description of the case from the beginning to its present state will be displayed which will be helpful for a new officer who has been allotted the case. This page on request also shows the pending and completed cases.

5. Contact Us

This page is important for common man to get in touch with the investigation team.

6. Criminal Events

   This page shows the description of the newly and completely solved cases by the officers and also at the end giving an important message to the readers.

7. Online Portal

   This page is completely for the common man, where he can login in and without any fear share his grievances.

2.2 Product Characteristics/Requirements

1. Templates and tools

   The system will require a Graphical User Interface (GUI) which will help user interaction. These files will be in Microsoft Word, Project, html, mysql , java script, java or PDF format, as appropriate.

2. Accessibility

   The site must be accessible using a standard Internet browser. Users must have appropriate application software to open several of the templates and tools.The site must be available 24 hours a day, 7 days a week, with one hour for system maintenance and other periodic maintenance, as appropriate.

3. Account

   User needs to loginto the site before posting the personal details providing some minimal essential information.

Chapter 3

Estimation

The chapter estimates provides cost, effort and time for the project Online Investigation Bureau. Planning involves estimation to determine the amount of money, effort, and time required in building the software based product.

3.1 Estimation Techniques Applied

Here, two different estimation techniques are applied to the project. The first technique is Lines Of Code (LOC), a software metric used to measure the size of a computer program by counting the number of lines in the text of the program's source code. SLOC is typically used to predict the amount of effort that will be required to develop a program, as well as to estimate programming productivity or maintainability once the software is produced.

The second technique is function point analysis (FPA) which is a "unit of measurement" to express the amount of business functionality an information system (as a product) provides to a user. Function points are used to compute a functional size measurement (FSM) of software. The cost (in dollars or hours) of a single unit is calculated from past projects.

3.2 Problem Base Estimation

Lines of code and function points were described as measures from which productivity metrics can be computed. LOC and FP data are used in two ways during software project estimation: (1) as an estimation variable to "size" each element of the software and (2) as baseline metrics collected from past projects and used in conjunction with estimation variables to develop cost and effort projections. Problem/Based Estimation‘ LOC and FP estimation are distinct estimation techniques. Yet both have a number of characteristics in common. The project planner begins with a bounded statement of software scope and from this statement attempts to decompose software into problem functions that can each be estimated individually.

LOC or FP (the estimation variable) is then estimated for each function.Alternatively, the planner may choose another component for sizing such as classes or objects, changes, or business Empirical model. A typical Empirical model is derived using regression analysis on data collected from past software projects.

The overall structure of such models takes the form E = A + B x (ev)powerC where A, B, and C are empirically derived constants, E is effort in person-months, and ev is the estimation variable.

3.3 Line Of Code based Estimation

LOC / Lines of Code, usually referring to non/commentary lines, meaning pure whitespace and lines containing only comments are not included in the metric.

The number of lines of program code is wonderful metric. It's so easy to measure and almost impossible to interpret.

There are many different ways to count lines (e.g., with or without comments, counting statements rather than lines, or counting lines in a automatically formatted code)

3.3.1 Estimate for LOC method

Estimated loc = (optimistic + 4 Most Likely + pessimistic)/6

User interface and control = (1000 + 4 1100 + 1200)/6 = 1100

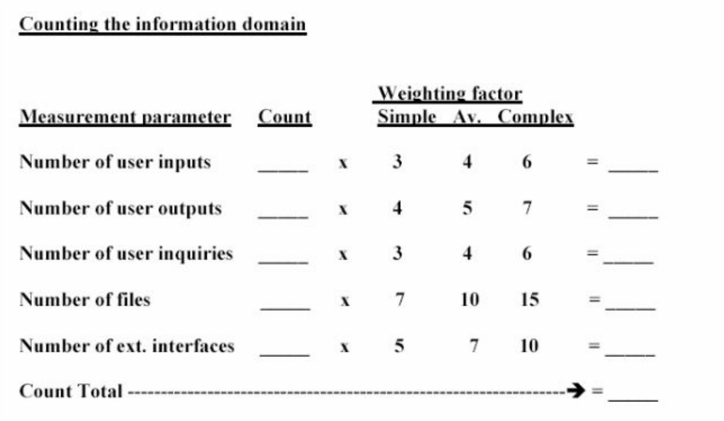
User Registration and Login = (1000 + 4 1200 + 1500)/6 = 1217

Showing Investigation details = (500 + 4 700 + 800)/6 = 684

Creating and updating accounts = (1000 + 4 1100 +1200)/6 = 1100

FUNCTION	Optimistic	Most Likely	Pessimistic	Estimate LOC
User Interface and control	1000	1100	1200	1100
User Registration and Login	1000	1200	1500	1200
Showing Investigation Details	500	700	800	700
Creating and updating criminal and officer account	1000	1100	1200	1100
Database management	400	500	600	500
Estimated lines of Code				4600

Table 3.1: Estimate for LOC Method



Counting the information domain

			Weighting factor				
Measurement parameter	Count		Simple	Av.	Complex		
Number of user inputs	____	x	3	4	6	=	____
Number of user outputs	____	x	4	5	7	=	____
Number of user inquiries	____	x	3	4	6	=	____
Number of files	____	x	7	10	15	=	____
Number of ext. interfaces	____	x	5	7	10	=	____

Count Total --→ = ____

Figure 3.1: Function point estimation

Database management = (400 + 4 500 + 600)/6 = 500

3.4 Function Point based estimation

Step 1: Calculate unadjusted FP
Inputs
3 simple x 2 = 6
Outputs
4 average x 5 = 20
Files
1 simple x 7 = 7
Inquiries
4 simple x 3 = 12
Interfaces
2 simple x 5 = 10

Unadjusted function points = 55

Step 2: Then we must consider some "complexity adjustment values" rated on a scale of 0 – 5 (Step 2 of FP Estimation)

- No influence 0

- Incidental 1
- Moderate 2
- Average 3

Chapter 4

Resources

For information system projects, more specific resources may include system developers,project managers,system analysts,stakeholders, development environments and information.

4.1 Project Resources

These include labour equipment (e.g. workstations)

- materials
- space
- services
- Elapsed time can often be reduced by adding more staff
- Money: used to buy other resources

4.2 Resource Allocation

- Identify the resources needed for each activity
- Identify resource types / individuals are interchangeable within the group (e.g. ‘VB programmers’ as opposed to ‘software developers’)
- Allocate resource types to activities and examine the resource histogram

4.2.1 Human Resources

1. Project Manager:

   - A project manager is the person responsible for accomplishing the stated project objectives. They Concentrate on resources where there is a possibility that, without planning, they might not be sufficiently available when required.

2. Developers:

   - Their work includes researching,designing,developing,and testing software. A software developer may take part in design, computer programming or software project management.

In this project, a group of 4 members (Dimple More, Priya Nayak, Kausha Nisar, Saumya Gangwar) under the guidance of project managers (professors) have to complete the task in 2 months.Team members need to complete different task together assigned by the project manager.

4.3 System requirements

The success of the project also depends upon the quality and availability of the tools which are being used to develop the software. The development requirements are of two types:

1. Hardware requirements:

   - Laptop
   - Net connection

2. Software requirements:

   - Operating System
   - SQL Server
   - NetBeans IDE
   - Java Development Kit

Chapter 5

Risk Analysis and Management

Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty. A risk is a potential problem/ it might happen or it may not. But regardless of its outcome we must identify it, assess its probability of occurrence, estimate its impact and establish a contingency plan in order to reduce its impact on the software engineering process. The following are the steps involved in risk analysis:

5.1 Risk Identification

Risk identification is a systematic attempt to specify threats to the project plan. By identifying known and predictable risks, the project manager takes a first step toward avoiding them when possible and controlling them when necessary.

There are two distinct types of risks for each of the categories

1. Generic risks are potential threats to every software product.

2. Product/specific risks – any special characteristics of the product that may threaten the project plan.

One method for identifying risks is to create a risk item checklist. The checklist can be used for risk identification and focuses on some subset of known and predictable risks in the following generic subcategories: Product size, Business Impact, Customer Characteristics, Staff Size and Experience, Process Definition, Development Environment and Technology to be built. For a “Event Management System” the following are the main risks that could occur:

5.1.1 Generic Risks:

- Risks associated with the ”newness” of the technology and thus lack of project experience. Project budget could get exceeded.
- Delay in the completion of the project due to changes in the requirements made by the customer in the final stages of the project.
- Funding could be lost due to the excessive delay and the software can become obsolete. Compromise on the quality of the software due to tightening of deadlines.

5.1.2 Product Specific Risks:

- If the server goes down, records could be lost and the officers won’t be able to access the information.
- Too many Users could use the system at the same time and the server could be flooded with requests. This traffic will slow down the service tremendously.
- The information is very confidential. Therefore misuse of the information by hacking of database will be dangerous.
- Unupdated data may cause chaos among officers.

5.2 Risk Projection or Risk Estimation:

Risk projection, also called risk estimation, attempts to rate each risk in two ways—the likelihood or probability that the risk is real and the consequences of the problems associated with the risk, should it occur.

The project planner, along with other managers and technical staff, performs four risk projection activities:

1. Establish a scale that reflects the perceived likelihood of a risk,
2. Delineate the consequences of the risk,
3. Estimate the impact of the risk on the project and the product, and
4. Note the overall accuracy of the risk projection so that there will be no misunderstandings

Imapct values

Risk No	Risk Name	Risk Type	Probability	Impact
1	Project delay	Project	30%	2
2	System Failure	Technical	45%	3
3	Lack of Appropriate Credentials	Technical	70%	1
4	Access by Multiple Users	Technical	70%	2
5	Unupdated Data	Technical	20%	4
6	Exceeding Budget	Project	40%	3

Table 5.1: Risk Estimation

1. catastropic
2. critical
3. marginal
4. negligible

5.3 Risk Mitigation, Monitoring and Management:

All of the risk analysis activities presented to this point have a single goal—to assist the project team in developing a strategy for dealing with risk.

An effective strategy must consider three issues:

1. Risk avoidance
2. Risk monitoring
3. Risk management and contingency planning

Risk Id:1 (Project Delay) Probability:30% Impact:2
Description: A specified time period is allotted for the completion of the entire project.This risk is also caused due to lack of co-ordination between the team members and even sometimes due to last minute changes.
Refinement/Context: Due to the delay in the project the customer could lose confidence in the software being built. Funding for the project could be taken back. Negative publicity and hence loss of value of the product before it comes into the market.
Mitigation, Monitoring and Management or Contingency Plan: We can avoid the delay of the project by following the prepared project schedule strictly and ensuring that the project team consists of some experienced professionals along with newly recruited staff. In this way there will be sufficient guidance for the trainees to handle all the problems that will lead to a delay. The developer should always communicate with the customer in a timely manner,ask for his inputs the software prepared so far and should entertain the changes requested by the customer and also every member must coordinate while the software is been developed. In this way there will be no delay due to last minute changes proposed by the customer.

Risk Id:2 (System Failure) Probability:45% Impact:3

Description: Software is not the only one element of a large computer/based system. Ultimately, software is incorporated with other system elements (e.g.hardware, people, information) .A system is a set of interacting or interdependent components forming an integrated whole .If any one of the vital components of the system fail it might result in the malfunctioning of the entire system or in its failure.Even sometimes due to power failure the system can crash or some other unethical access can cause system failure.

Refinement/Context: It will lead to not only loss of vital data like criminal or civilian database but also results in loss of faith in the system by the customers.The customers could therefore stop using the software which may remove the application from the market completely.

Mitigation, Monitoring and Management or Contingency Plan:

Periodically backing up the database, either onto tapes or onto remote servers,to prevent loss of data in case of disasters such as flooding.

Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required. Monitoring jobs running on the database and ensuring that performance is not degraded by very expensive tasks submitted by some users.

Risk Id:3 (Lack of Appropriate Credentials) Probability:70% Impact:1

Description: IF we did not take the appropriate login credentials, software will be vulnerable to hacking and other security threats.

Refinement/Context: This may result in insecure software. Confidential information can be broken into and misused. Thus, consequences can be disastrous.

Mitigation, Monitoring and Management or Contingency Plan: We will try to secure our database. Suppose an invalid login occurs, we can inform the user with that ID that a login threat may have occured.

Risk Id:4 (Exceeding Budget) Probability:40% Impact:3

Description: The exceeding of the stated budget due to inaccurate estimation of time and money is a potential risk.

Refinement/Context: There is a threat of the project being stopped completely due to the extra money required for its completion. This will creates a bad reputation of the company in the market

Mitigation, Monitoring and Management or Contingency Plan:

Determine the budget process.

Determine what facilities are to be included within the project budget. The total project budget is the sum of the itemised estimated costs plus contingency for project risk. Typically, contingency (for project risk) can range from 5complex or long project.

Regularly review the budget, both on an informal and formal basis.

Chapter 6

Project Scheduling And Tracking

For the scheduling of project,we take following steps:
Split the project into tasks and estimate time and resources required to complete each task.

1. Organize tasks concurrently to make optimal use of workforce and time.
2. Minimize task dependencies to avoid delays caused by one task waiting for the other to complete.
3. Depends on the manager's intuition and experience.

The process model, framework activities and task set are included here.

6.1 Task set for the project

A task set is the collection of work tasks, milestones, and deliverables.Different set of task proves to be appropriate to different projects. An effective software process is the collection of task sets , each designed to meet the needs of different types of project Task sets are designed to accommodate different types of project and different degrees of rigor.
Typical Project Types

Concept Development Project

New application development

Project Application enchanement

Project Application maintenance

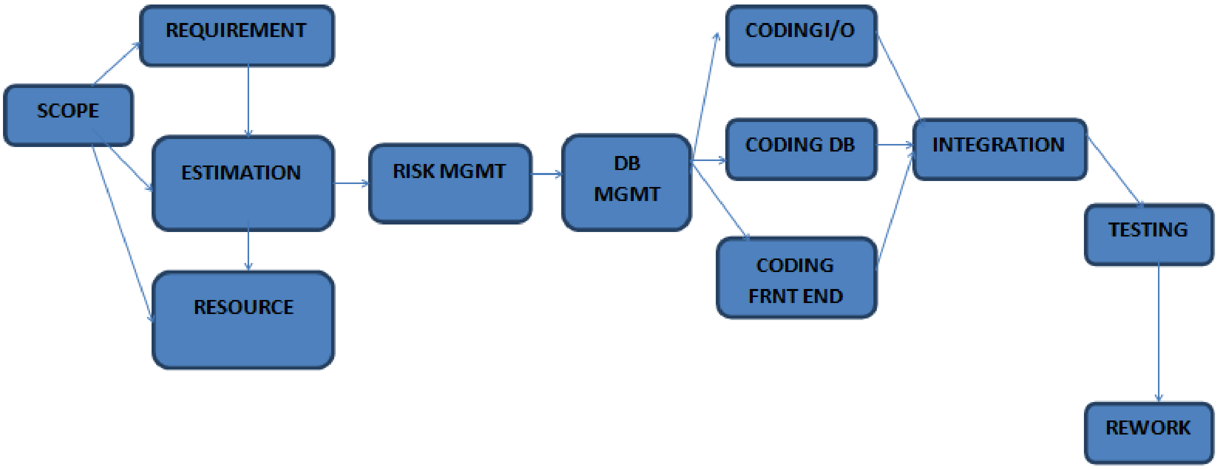


Figure 6.1: Activity Diagram

project reengineering
Degree of rigor Casual ,Structured,Strict or Quick reaction

Grades

0/nonincidental
1/minimal
2/low
3/medium
4/subordinate

Task Selector Value Computation

Task Selector Value=

6.2 Task Network

Project tasks and dependencies are noted diagrammatically in task network.Functional decomposition is as shown.

Activity Diagram:

Adaption criteria	Grade	Weight	Product
Size of the project			
Number of potential users			
Mission criticality			
Application Longevity			
Stability of requirements			
Ease of communication			
Maturity of applicable technology			
Performance Constraint			
Embedded/non embedded characteristics			
Project Staffing			
Interoperability			
Reengineering factors			

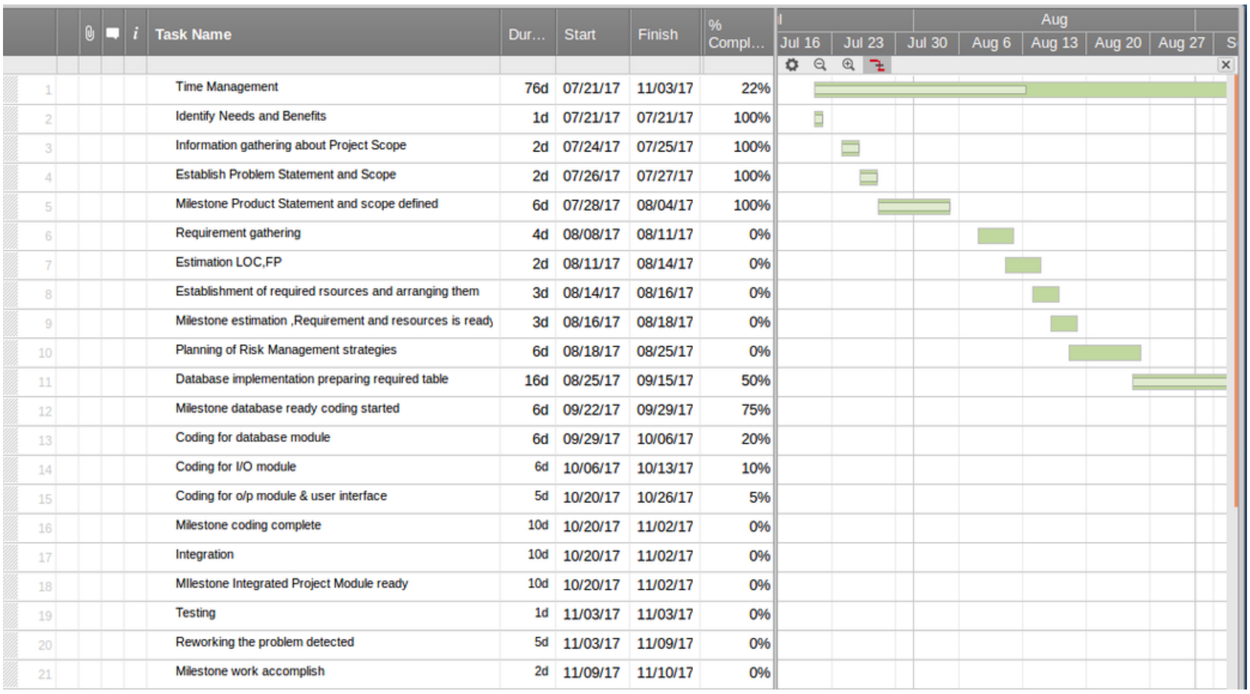
Table 6.1: Task Selector Value Computation

Task Selector Value	Degree of rigor
TSS ¡1.2	Casual
1.0 ¡ TSS ¡3.0	Structured
TSS ¿ 2.4	Strict

Table 6.2: Reference task selector values

Task	Description
T1	System Design
T2	Detail Design
T3	Database Implementation
T4	Webpage Design
T5	Coding Input Module
T6	Coding database related module
T7	Coding output module
T8	Test Planning
T9	Integration
T10	Testing
T11	Rework
T12	Final Check

Table 6.3: Task selector value computation



	Task Name	Dur…	Start	Finish	% Compl…
1	Time Management	76d	07/21/17	11/03/17	22%
2	Identify Needs and Benefits	1d	07/21/17	07/21/17	100%
3	Information gathering about Project Scope	2d	07/24/17	07/25/17	100%
4	Establish Problem Statement and Scope	2d	07/26/17	07/27/17	100%
5	Milestone Product Statement and scope defined	6d	07/28/17	08/04/17	100%
6	Requirement gathering	4d	08/08/17	08/11/17	0%
7	Estimation LOC,FP	2d	08/11/17	08/14/17	0%
8	Establishment of required rsources and arranging them	3d	08/14/17	08/16/17	0%
9	Milestone estimation ,Requirement and resources is read	3d	08/16/17	08/18/17	0%
10	Planning of Risk Management strategies	6d	08/18/17	08/25/17	0%
11	Database implementation preparing required table	16d	08/25/17	09/15/17	50%
12	Milestone database ready coding started	6d	09/22/17	09/29/17	75%
13	Coding for database module	6d	09/29/17	10/06/17	20%
14	Coding for I/O module	6d	10/06/17	10/13/17	10%
15	Coding for o/p module & user interface	5d	10/20/17	10/26/17	5%
16	Milestone coding complete	10d	10/20/17	11/02/17	0%
17	Integration	10d	10/20/17	11/02/17	0%
18	MIlestone Integrated Project Module ready	10d	10/20/17	11/02/17	0%
19	Testing	1d	11/03/17	11/03/17	0%
20	Reworking the problem detected	5d	11/03/17	11/09/17	0%
21	Milestone work accomplish	2d	11/09/17	11/10/17	0%

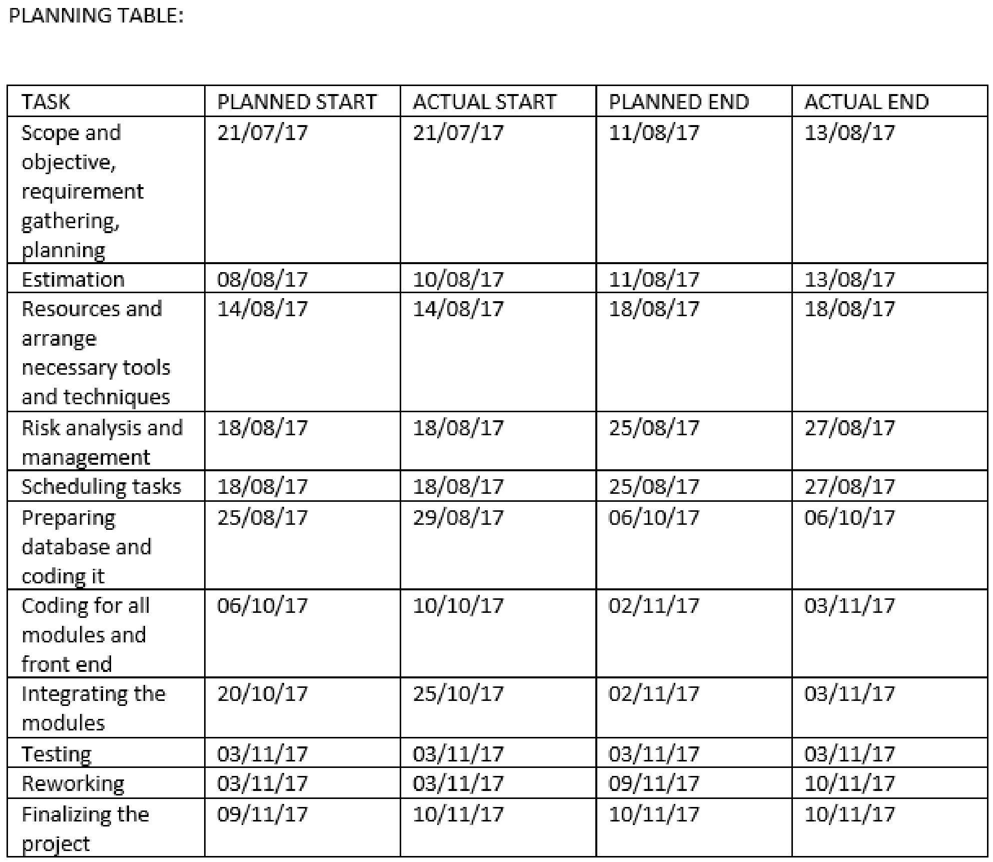
Figure 6.2: Gantt chart

6.3 Scheduling and timeline chart

Time Flow Chart

Tracking the Schedule flow of Project Schedule

- Establish the project constraint
- Make initial assessments of project parameters
- Define project milestones and deliverables.
- While project has not been completed or cancelled
- Draw up project schedule
- Initiate activities according to schedule
- Wait(for a while)
- Review project Progress
- Revise estimates of project parameters



PLANNING TABLE:

TASK	PLANNED START	ACTUAL START	PLANNED END	ACTUAL END
Scope and objective, requirement gathering, planning	21/07/17	21/07/17	11/08/17	13/08/17
Estimation	08/08/17	10/08/17	11/08/17	13/08/17
Resources and arrange necessary tools and techniques	14/08/17	14/08/17	18/08/17	18/08/17
Risk analysis and management	18/08/17	18/08/17	25/08/17	27/08/17
Scheduling tasks	18/08/17	18/08/17	25/08/17	27/08/17
Preparing database and coding it	25/08/17	29/08/17	06/10/17	06/10/17
Coding for all modules and front end	06/10/17	10/10/17	02/11/17	03/11/17
Integrating the modules	20/10/17	25/10/17	02/11/17	03/11/17
Testing	03/11/17	03/11/17	03/11/17	03/11/17
Reworking	03/11/17	03/11/17	09/11/17	10/11/17
Finalizing the project	09/11/17	10/11/17	10/11/17	10/11/17

Figure 6.3: Planning Table

- Update project schedule
- Re/negotiateproject constraints and deliverables
- If(problems arise)then
- Initialize technical review and revision
- End if
- End loop

Planning Table

6.4 Tracking and scheduling

When you are managing a project, one of the most important responsibilities is effectively scheduling tasks and tracking their completion. This helps ensure that your project is completed on time and within budgetary constraints. There are different ways of tracking project.

1. Conducting different periodic project status meetings where team members report progress and problems.
2. Evaluating results of all reviews throughout the process.
3. Determining whether the formal milestones are accomplished within time as depicted by Gantt chart.

Chapter 7

Project Plan

The software project plan is produced at the culmination of the planning task. It provides baseline cost and scheduling information that will be used throughout the engineering process. The software project plan is relatively brief document that is addressed to diverse audience. An outline of the plan is presented below.

7.1 Project planning process

1. A statement of work (SOW) that describes all work produce that will be produced and a list of all resources people who will perform that work.

2. A resource list that contains a list of all resources that will be needed for the product and their availability.

3. A work breakdown structure and set of estimates.

4. We framed a project schedule and task network to plan the work considering the given deadlines.

5. A risk plan that identifies any risks that might be encountered and indicates how those risks would be handled should they occur.

The outline of the project is as presented below:

1. **Definition and Introduction of Problem Statement:**
   A problem statement is a clear concise description of the issue(s) that need(s) to be addressed by a problem solving team. Problem statement for this project is Online Event Management System.

2. **Project Scope and objective:**
   Software scope describes the data and control to be processed, functions, performance requirements, constraints, interfaces, and reliability requirements. Scope is defined by answering the following questions:

   (a) Context

   (b) Information Objectives

   (c) Function and Performance

3. **Project Estimation :**
   Software project estimation is a form of problem solving where the problem to be solved is too complex. For this reason, we decompose the problem.

   (a) Estimation Techniques: Could be direct or indirect.

   (b) Problem Based Estimation: The project planner begins with a bounded statement of software scope and from this statement attempts to decompose software into problem functions that can each be estimated individually. LOC or FP (the estimation variable) is then estimated for each function.

   (c) LOC Based Estimation: A method to measure size by counting number of lines of Code.. When LOC is used as the estimation variable, decomposition is absolutely essential. The greater the degree of partitioning, the more likely reasonably accurate estimates of LOC can be developed.

   (d) FP Based Estimation: Function Point Analysis (FPA): is method to measure the functional size of an information system. The functional size reflects the amount of functionality that is relevant to and recognized by the user in the business.

   (e) Process Based Estimation: The process is decomposed into a relatively small set of tasks and the effort required to accomplish each task is estimated.

4. **Project Resources:**
   Resources are commonly thought of as sources of supply or support, such as money, people, materials, technology, and space

   (a) Human: It is personnel pool like developers, managers, planners, testers etc available to an organization.

   (b) Reusable: Reusable resources are the sources already available while doing some other projects or creating the new component to reuse it in some other project.

   (c) System Requirements: It consists of planning ,managing and making the necessary software/hardware components available.

5. **Risk Analysis and Management:**
   Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty. It consists of the following steps:

   (a) Identification: Risk identification is a systematic attempt to specify threats to the project plan. By identifying known and predictable risks, the project manager takes a first step toward avoiding them when possible and controlling them when necessary

   (b) Projection: Risk projection, also called risk estimation, attempts to rate each risk in two ways—the likelihood or probability that the risk is real and the consequences of the problems associated with the risk, should it occur.

   (c) Risk Mitigation, Monitoring and Management: All of the risk analysis activities presented to this point has a single goal—to assist the project team in developing a strategy for dealing with risk.

6. **Project Scheduling and Planning:**

   (a) Defining Task Set for Software Project : A task set is the collection of work tasks, milestones, and deliverables.

   (b) Task Network: Project tasks and dependencies are noted diagrammatically in task network according to its functional dependencies.

   (c) Time Line chart: Project milestones can be shown in a simple time line chart .While the chart doesn't look complicated, it provides good amount of information on project progress in a simple and understandable chart.

   (d) Tracking the Schedule: Tracking helps ensure that your project is completed on time and within budgetary constraints.

7. **Project Plan:**
   Goal is to establish a pragmatic strategy for controlling, tracking, and monitoring a complex technical project.The purpose of project planning

is to ensure that the end result is completed on time, within budget, and exhibits quality. The steps involved are:Project Scope, Estimates, Risks, Schedule Control strategy.

8. **Software Quality Assurance:**
   Software quality assurance (SQA) is an umbrella activity that is applied throughout the software process. SQA encompasses:

   (a) a quality management approach,

   (b) effective software engineering technology (methods and tools),

   (c) formal technical reviews that are applied throughout the software process,

   (d) a multi-tier testing strategy,

   (e) control of software documentation and the changes made to it,

   (f) a procedure to ensure compliance with software development standards

   (g) measurement and reporting mechanisms.

- **Quality Concept:** Quality is a characteristic or attribute of something. In our software project it is of 2 types: quality of design which encompasses requirements and quality of conformance that focuses on implementation.
- **Quality Control:** Quality control involves the series of inspections, reviews, and tests used throughout the software process to ensure each work product meets the requirements placed upon it.
- **Cost of Quality:** The cost of quality includes all costs incurred in the pursuit of quality or in performing quality/related activities.
- **SQA:** Stands for Software Quality Assurance. Quality assurance consists of the auditing and reporting functions of management.
- **SQA Plan:** The plan is developed during project planning and is reviewed by all interested parties. Quality assurance activities performed by the software engineering team and the SQA group are governed by the plan.

9. **Software Configuration Management:**
   Software configuration management (SCM) is an umbrella activity that is applied throughout the software process. Because change can occur at any time, SCM activities are developed to:

   (a) identify change,
   (b) control change,
   (c) ensure that change is being properly implemented,
   (d) report changes to others who may have an interest.

   - SCM :
     Software configuration management (SCM) is a set of activities designed to control change by identifying the work products that are likely to change,establishing relationships among them,defining mechanisms for managing different versions of these work products, controlling the changes imposed, and auditing and reporting on the changes made.
   - SCM Process:
     The software configuration and management process consists of the following five SCM tasks: identification, version control, and change control, configuration auditing,and reporting.

10. **Requirement Analysis Modeling:**

(a) Data Modeling : ER Diagram: An entity relationship diagram is an excellent tool for planning and designing a database. The entity relationship model starts with the entities, data normalization starts with the attributes and the two tools tend to verify each other. The entity relationship model's entities, attributes and relationships map smoothly to a physical database.

(b) Functional Modeling: Data Flow Diagram: DFDs are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing . A DFD shows what kinds of information will be input to and output from the system,where the data will come from and go to, and where the data will be stored

(c) Data Dictionary: A data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them.

11. **Design:**
Designing is a constructive task. Software should be designed in a manner that uses anti bugging techniques. That is, software should be capable of diagnosing certain classes of errors. In addition, the design should accommodate automated testing and regression testing.

12. **Project maintenance and Re-usability:**

(a) Software maintenance in software engineering is the modification of a software product after delivery to correct faults, to improve performance or other attributes. The key software maintenance issues are both managerial and technical.

(b) Reusability is the likelihood a segment of source code that can be used again to add new functionalities with slight or no modification. Reusable modules and classes reduce implementation time, increase the likelihood that prior testing and use has eliminated bugs and localizes code modifications when a change in implementation is required

Chapter 8

Software Quality Assurance

SQA is the process of evaluating the quality of a product and enforcing adherence to software product standards and procedures. It is an umbrella activity that ensures conformance to standards and procedures throughout the SDLC of software product. There are a large number of tasks involved in SQA activities

8.1 Quality Concepts:

Software quality assurance (SQA) is an umbrella activity that is applied throughout the software process. SQA encompasses

1. Formulating a quality management plan
2. Applying software engineering techniques
3. Conducting formal technical reviews
4. Applying a multi-tiered testing strategy
5. Enforcing process adherence
6. Controlling change
7. Measuring impact of change
8. Performing SQA audits
9. Keeping records and reporting

Quality refers to measurable characteristics of software. These items can be compared based on the given standards:

- Quality of design:

  Refers to the characteristics that designers specify for an item.The grade of materials, tolerances, and performance specifications all contribute to the quality of design. As higher/grade materials are used, tighter tolerances and greater levels of performance are specified, the design quality of a product increases, if the product is manufactured according to specifications.

- Quality of conformance:

  It is the degree to which the design specifications are followed during manufacturing. Again, the greater the degree of conformance, the higher is the level of quality of conformance. Crucial is customer satisfaction (quality is only a part of it): User Satisfaction= Compliant product +Quality +Delivery within budget and Schedule

8.2 Quality Control:

Quality control involves the series of inspections, reviews, and tests used throughout the software process to ensure each work product meets the requirements placed upon it. Quality control includes a feedback loop to the process that created the work product. A key concept of quality control is that all work products have defined, measurable specifications to which we may compare the output of each process. The feedback loop is essential to minimize the defects produced. Quality assurance consists of the auditing and reporting functions of management. The goal of quality assurance is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confidence that product quality is meeting its goals.

8.3 Cost of Quality:

Cost of quality includes all costs incurred in the pursuit of quality or perform quality relate work Quality cost includes: Prevention costs include:

1. Test equipment
2. Formal technical reviews
3. Test equipment
4. Training

Appraisal costs include activities to gain insight into product condition the “first time through” each process. Appraisal costs include :

1. In-process and Inter-process inspection

2. Equipment calibration and maintenance

3. Testing

Failure costs are those that would disappear if no defects appeared before shipping a product to customers.Failure costs may be subdivided into internal failure costs and external failure costs. Internal failure costs are incurred when we detect a defect in our product prior to shipment. External failure costs are associated with defects found after the product has been shipped to the customer.

8.4 SQA

Software Quality: Quality definition: “Conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that expected of al professional developed software.” Three import points for quality measurement: - Use requirements as the foundation - Use specified standards as the criteria - Considering implicit requirements

8.5 SQA Plan

1. Introduction

   The SQA plan provides a road map for instituting software quality assurance.

   (a) Purpose: This document outlines the actions of our team in order to make our system “Event Management System” and other related artifacts conform to the requirements of the customers and the qualitative standards within the specified project resources and constraints following IEEE standards.

   (b) Scope: The scope of this document is to outline all procedures, techniques and tools to be used for quality assurance of this project. This plan:

      i. Identifies the SQA responsibilities of the project developer and the SQA consultant

   ii. Lists the activities, processes, and work products that the SQA consultant will review and audit
   iii. Identifies the SQA work products

2. Management

   A description of each major element of the organization and a description of the SQA tasks and their relationships. It consists of the following topics:

   (a) Organization: This project is a team work where strength of each team is 2 members.The tasks are divided among the members by mutual co/ordination.
   (b) Tasks and Responsibilities: The responsibilities of the developers are as follows:
      i. Develop the requirement specification and cost estimation for the project
      ii. Develop the design plan and test plan for testing the tool
      iii. Implement and test the application and deliver the application along with the necessary documentation.
      iv. Give a formal presentation to the committee on completion of the analysis, design and testing phases. The committee reviews the developer's work and provides feedback/suggestions.
      v. Planning, coordinating, testing and assessing all aspects of quality issues.

3. Documentation

   In addition to this document, the essential documentation will include: The Software Requirements Specification (SRS), which prescribes each of the essential requirements (functions, performances, design constraints and attributes) of the software and external interfaces The Formal Specification Document, which gives the formal description of the product design specified in Object Constraint Language (OCL). The Software Design Description (SDD) which depicts how the software will be structured. Software Test Plan: Describes the test cases that will be employed to test the product. Software User Manual (SUM) which will identify the required data and control inputs, input sequences, options, program limitations or other actions. Standards, practices, conventions, and metrics: Metrics: LOC / lines of code and Function Points (FP) are used to measure the size of the software.

4. Reviews and audits:

   The Committee will perform reviews at various stages of the project. This review will determine whether the requirements have been met for the deliverable, check that the product meets the requirements, ensure that the SQA plan has been adhered to, verify the performance of the software and ensure that acceptance testing is carried out. In addition the developer will conduct a Formal Technical Review after the design phase. A design checklist will be used and the developer will check to see whether his/her design meets the checklist criteria.

5. Test:

   Testing will be carried out in accordance with the Software Testing Plan (STP). Testing documentation will be sufficient to demonstrate that testing objectives and software requirements have been met. Test results will be documented and discussed in the final phase of the project.

6. Problem reporting and corrective action

   Problem reporting and corrective action;

   (a) problem identification and correction occurring during software development to verify early detection of actual or potential problems
   (b) reporting of the problem to the proper authority,
   (c) analysis of the problem to propose corrective measures,
   (d) Timely and complete corrective action
   (e) the recording and follow/up of each problem's status.

7. Tools, techniques, and methodologies:

   The tools are evaluated for adequacy by assessing whether they perform the desired functions and for applicability by assessing whether the tool capabilities are needed for the software development or support. Planned tools are evaluated for feasibility.

8. Code control:

   Code control includes the items listed below:

   (a) Identifying, labeling, and cataloging the software to be controlled
   (b) Identifying the physical location of the software under control
   (c) Identifying the location, maintenance, and use of backup copies

(d) Distributing copies of the code

(e) Identifying the documentation that is affected by a change

(f) Establishing a new version

(g) Regulating user access to the code.

9. Records collection, maintenance, and retention:

   SQA activities are documented by records and reports that provide a history of product quality throughout the software life cycle. Measurement data collected will be reviewed for trends and process improvement. All SQA records will be collected and maintained in the SDL or archival storage for the life cycle of the product or a minimum of few years.

10. .Risk management:

    Risk Management is done according to software documentation audit of project.

Chapter 9

Software Configuration Management

Software configuration management is the discipline of managing the evolution of complex software systems. It is also defined as 'the process of controlling and monitoring change to work products'

9.1 Configuration Management:

The items that comprise all information produced as part of the software process are collectively called a software configuration for eg:computer programs,documents and data.

Changes are inevitable and in most of the cases, justified . Customers may have modified requirements.Developers may want to modify the technical approach. Managers want to modify the project strategy Changes should be

1. Analyzed in advance
2. Recorded before implementation
3. Reported by the need/to know basis
4. Controlled to improve quality and reduce errors.

The sources of changes can be stated as:

1. New business or market conditions
2. New customer needs demand modification of data produced by information systems, functionality delivered by products, or services delivered by a computer based system.

3. Reorganization or business growth/downsizing causes

4. Budgetary or scheduling constraints

9.2 Software Configuration Management (SCM) Process:

SCM Umbrella Activity has following subtasks

- Identifies
- controls
- audits
- reports modifications

that invariably occur while software is being developed and after it has been released to a customer. All information produced as part of software engineering becomes part of a software configuration.The configuration is organized in a manner that enables orderly control of change.
SCI: Software Configuration Items: There are growing number of artifacts for manage sharing the SE process. They are

1. **Programs:** source level and executable

2. **Documents:** Technical practitioners and users

3. **Data:** external and internal

Types Of SCI:

1. Requirements specification

2. Project Plan

3. Preliminary User Manual

4. Design Specification

5. User code listing

6. Test specifications

7. Installations/operations

8. Executable programs
9. Database description
10. As/built user manual
11. Maintenance documents
12. Standards and procedures

SCM Tasks:

1. **Identifying change:**
   The identification scheme for software objects must recognize that objects throughout the software process .Evolution graphs for each SCIs are used .For example in EMS,the changes can be in terms of number of people accessing site simultaneously may increase. For eg:
   (a) Officers can handle more than one cases and give inputs in other cases as well. That is suggest links in other cases. Change in no of officers that have access to the case.
   (b) Customers can put their cases in fast track with a certain criteria .
   (c) Every officer will have some specialty. Cases will be assigned according to the specialty.
2. **Version control:**
   Version control combines procedures and tools to manage different versions of configuration objects that are created during the software process.
3. **Control change:**
   An engineering change order (ECO) is generated for each approved change. The ECO describes the change to be made, the constraints that must be respected, and the criteria for review and audit For eg:
   (a) Approved changes : every officer will have some specialty .Cases will be assigned according to the specialty.
   (b) ECO:(ENGINEERING CHANGE ORDER) Every officer will have some specialty .Cases will be assigned according to the specialty.
   (c) Constraints : an officer can handle 2 cases at maximum

4. **Configuration Auditing:**
   A software configuration audit complements the formal technical review by assessing a configuration object.

5. **Reporting changes to others**

Chapter 10

Requirement Analysis Modeling

Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, analyzing and managing software or system requirements.

Requirement analysis modeling uses combination of text and diagrammatic form to depict requirements for data, function and behavior in a way that is relatively easy to understand and to review for correctness, completeness and convenience.

10.1 Data Modeling: Entity Relationship Diagram

ERD depicts relationships between data objects. The ERD is the notation that is used to conduct the data modeling activity.The attributes of each data objects noted in the ERD can be described using data object description.
Use Case Analysis

ER Diagram

10.2 Functional Modelling

Data Flow diagram

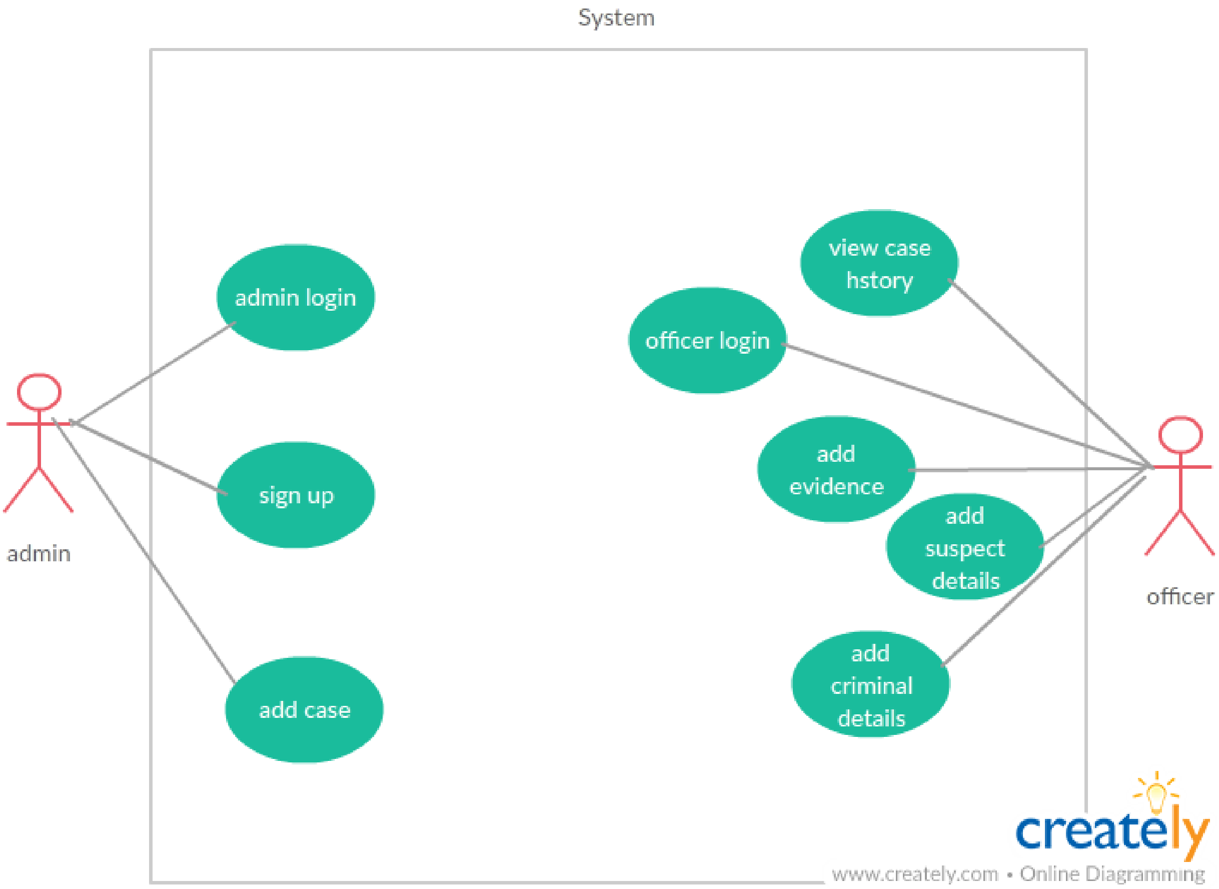


Figure 10.1: Usecase diagram

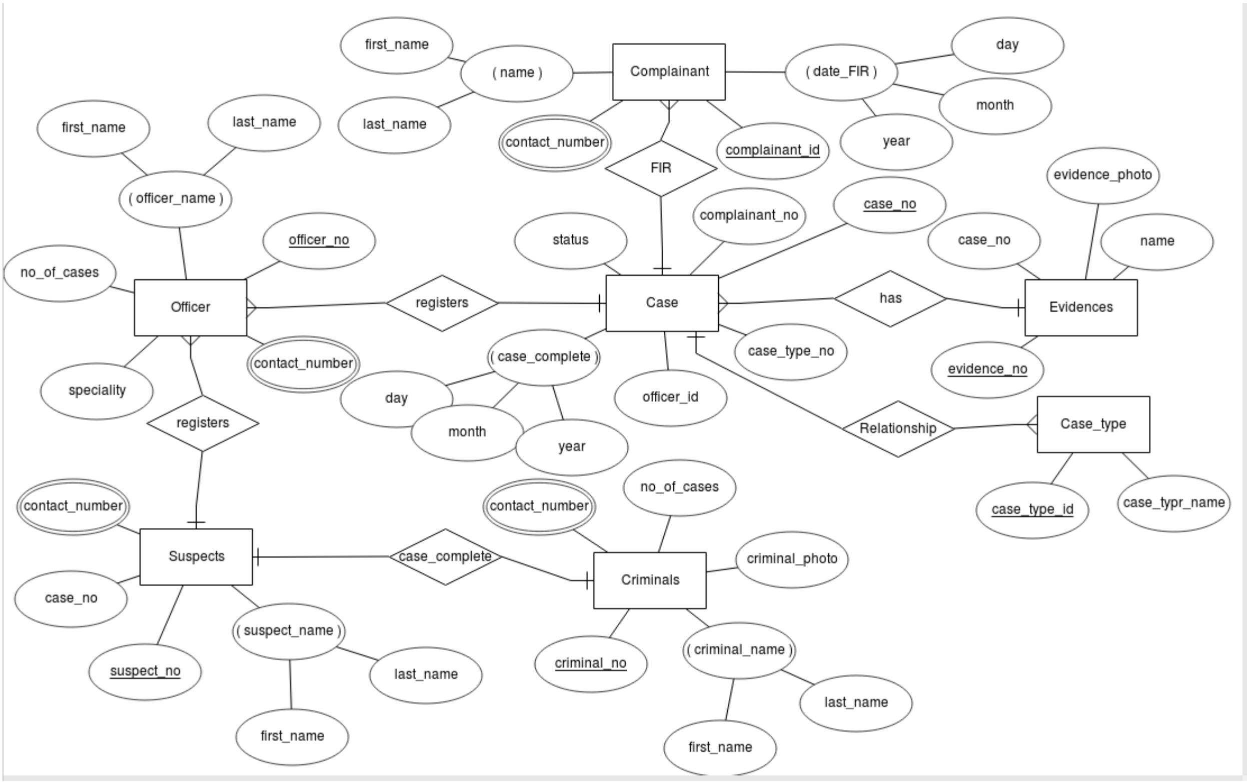


Figure 10.2: ER diagram

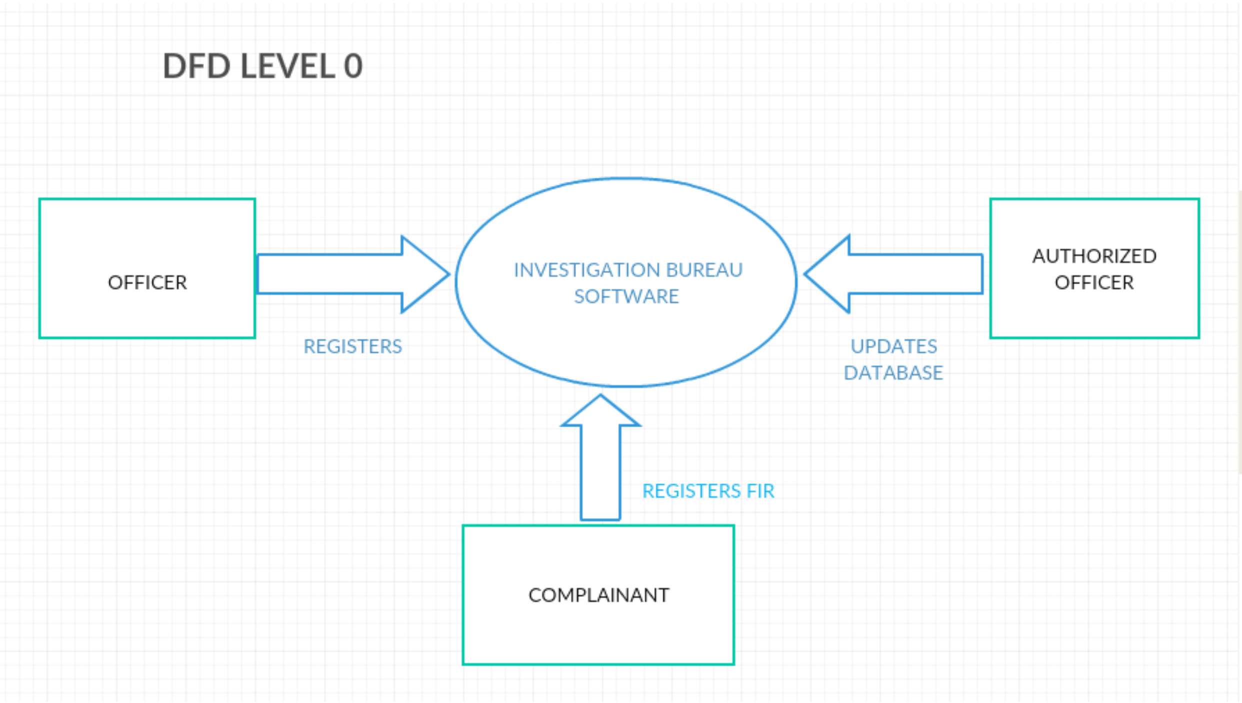


Figure 10.3: DFD Level 0

A data flow diagram (DFD) is a graphical representation of the ”flow” of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

10.3 Behavorial Modeling: State Transition Diagram

Behavioral modeling is an operational principle for all requirements analysis methods.The state transition diagram represents the behavior of a system by depicting its states and the events that cause the system to change state. In addition, the STD indicates what actions (e.g., process activation) are taken as a consequence of a particular event.
State Transition Diagram

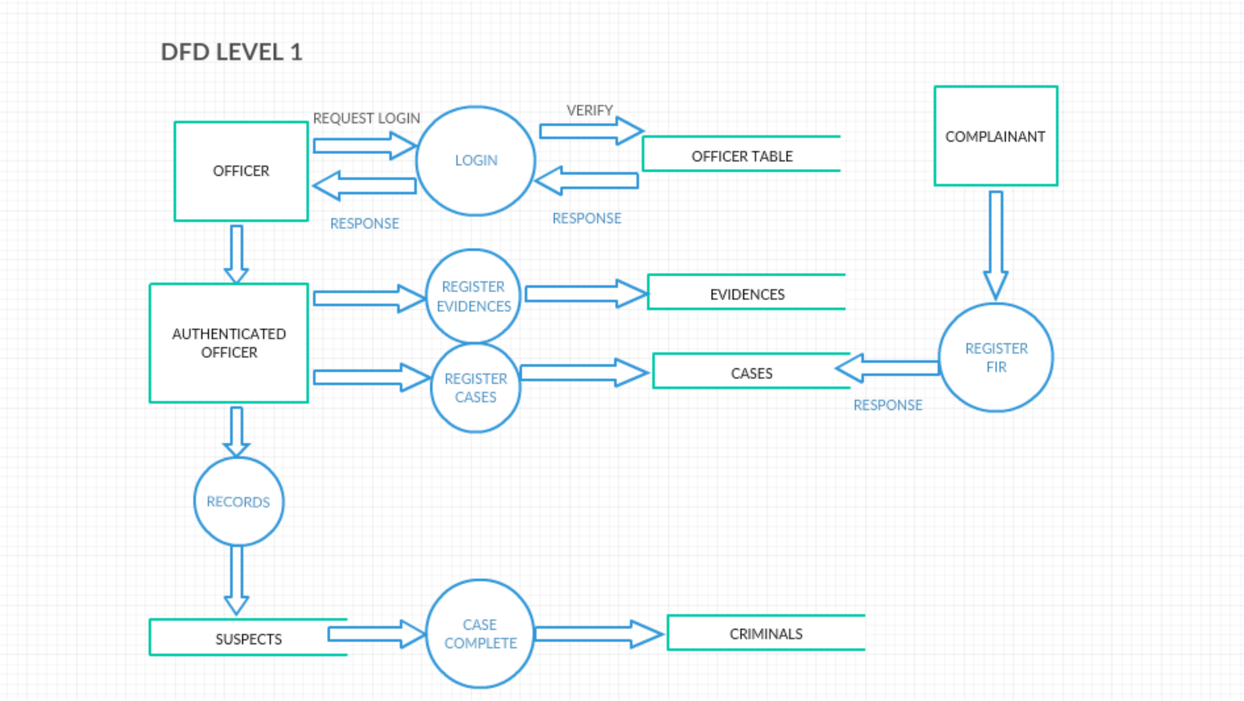


Figure 10.4: DFD Level 1

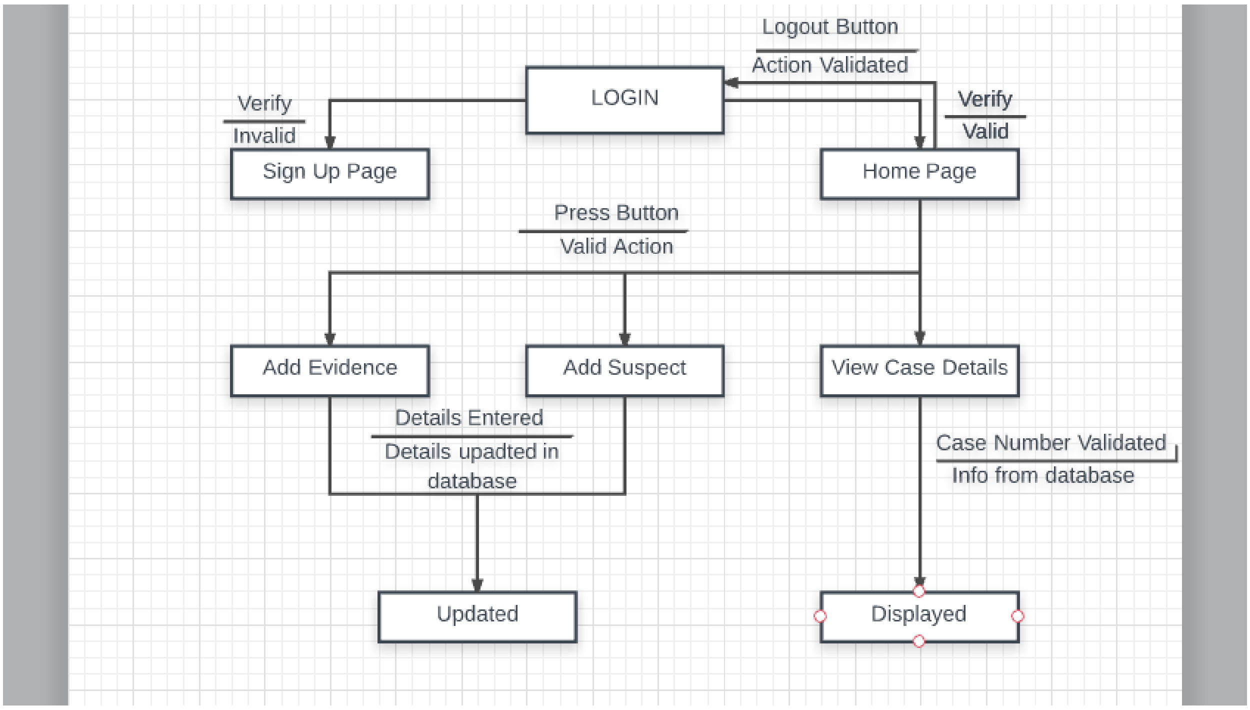


Figure 10.5: State Transition Diagram

10.4 Data Dictionary

A data dictionary, or data repository, is a central storehouse of information about the system's data. The main purpose of a data dictionary is to describe, document and organize facts about the:

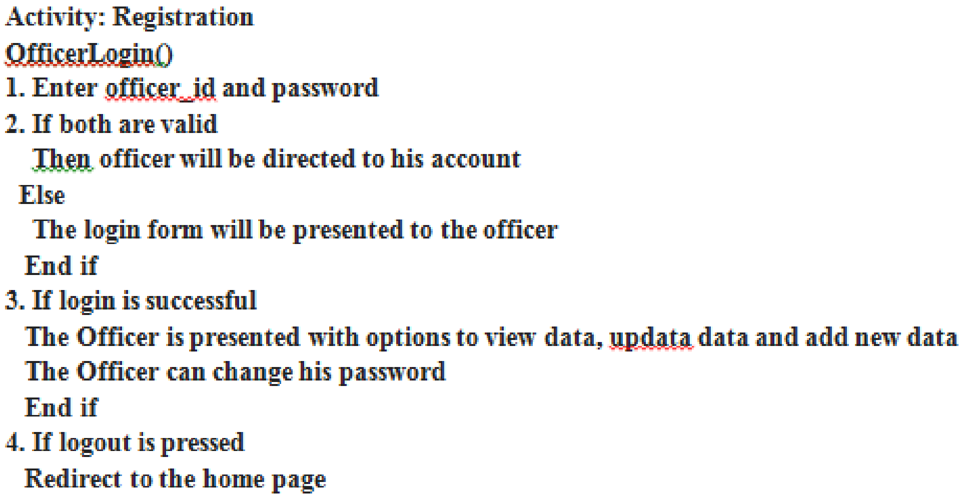
- data flows
- data stores
- processes
- external entities

Title of Object class	Description
Complainant	This person lodges a FIR which is made into a case by the admin officer
Case	It contains all the information regaring the case,like its case id,type of case,officer alloted,descriOfficern given by the complainant,etc
Officer	This contains all the information regarding the officer ie.name,address,mob no,email id,photo,etc
Suspect	This contains all the information of the suspect pointed out by the investigating officer like name,address, mob no,photo,relation and description
Evidence	This contains all the information regarding the physical or logical evidence type got at the crime site like evidence type,photo of crime scene,suspect,points to rate the evidence,etc
Criminal	This is got from the evidence table where points are considered for predicting the criminal

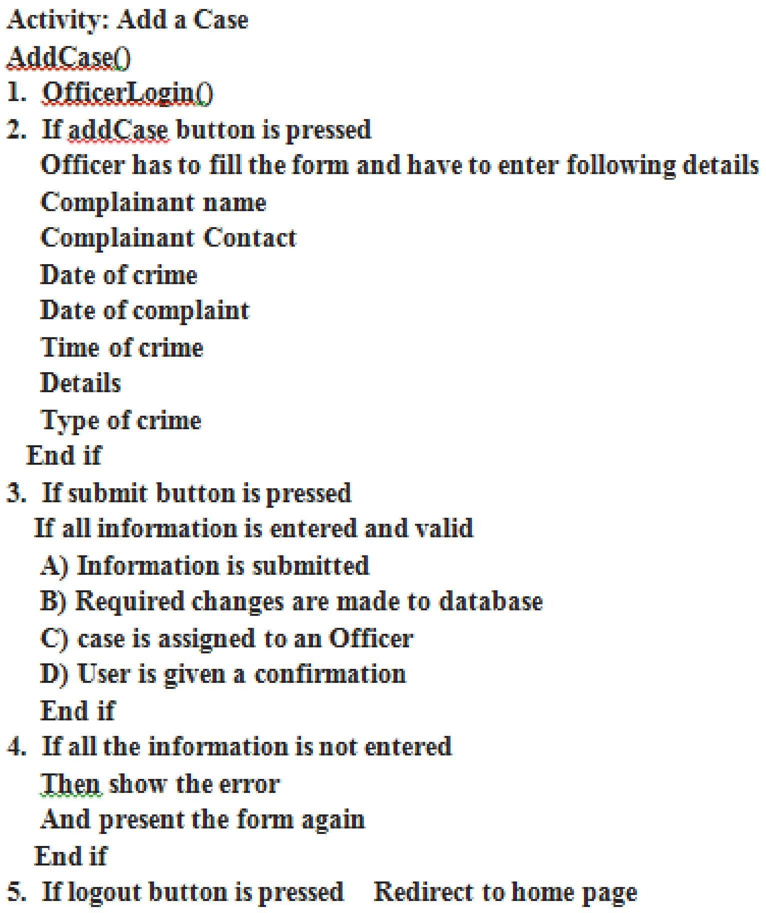
10.5 Algorithm

For Officers:

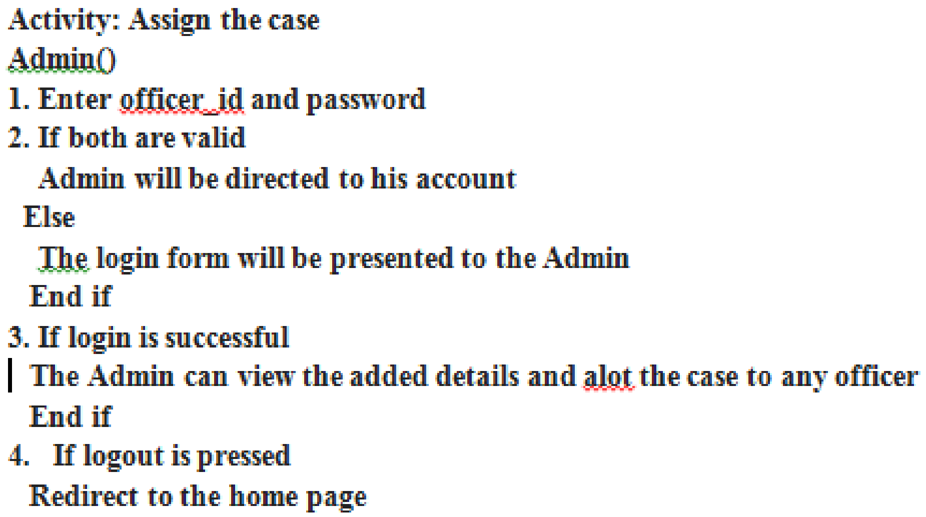
Register by entering personal detains and by assigning a suitable officer_id and password. A login form is displayed



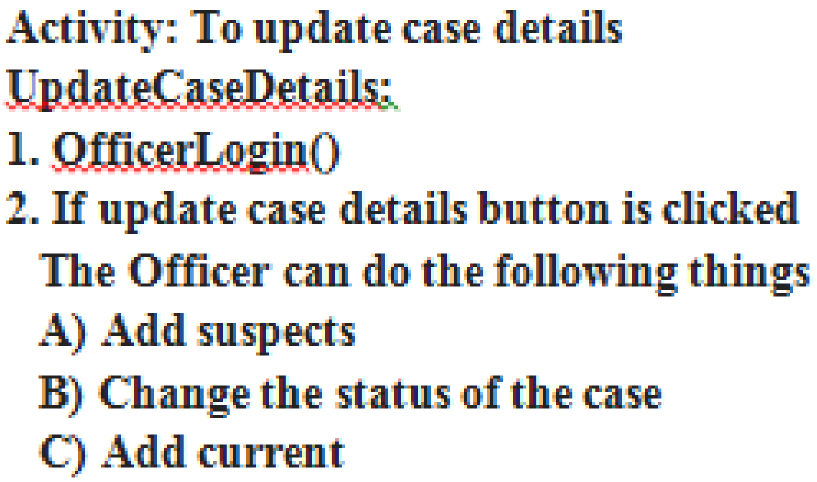
```
Activity: Registration
OfficerLogin()
1. Enter officer_id and password
2. If both are valid
     Then officer will be directed to his account
   Else
     The login form will be presented to the officer
    End if
3. If login is successful
    The Officer is presented with options to view data, updata data and add new data
    The Officer can change his password
    End if
4. If logout is pressed
    Redirect to the home page
```



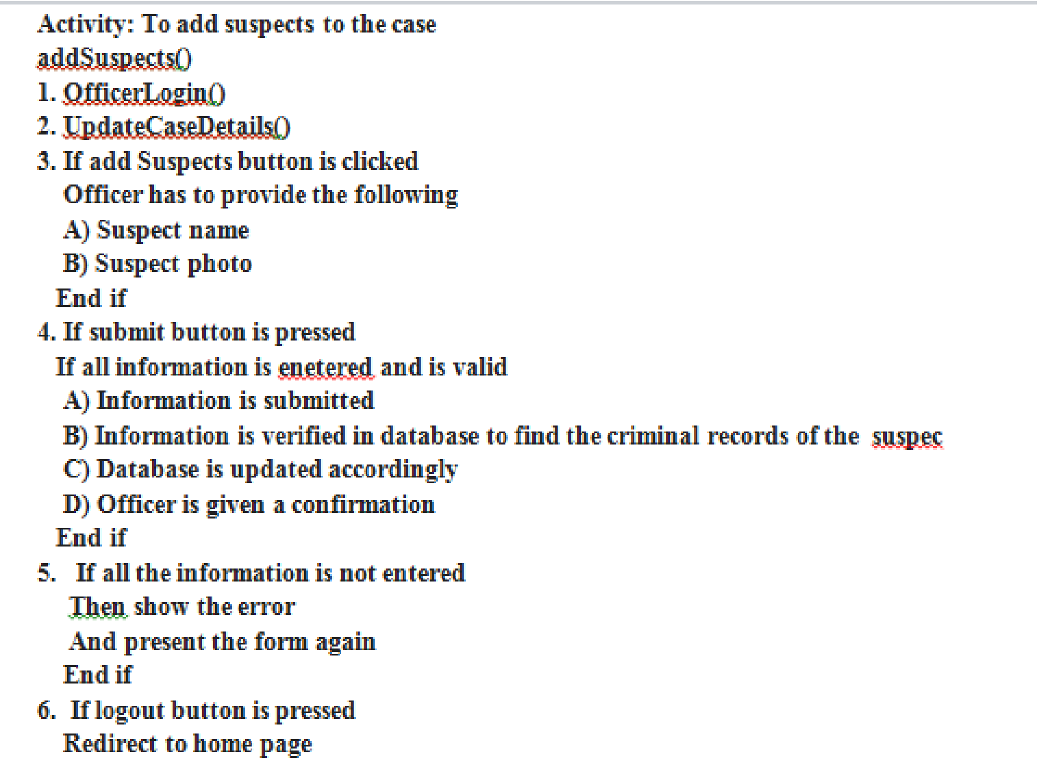
```
Activity: Add a Case
AddCase()
1.  OfficerLogin()
2.  If addCase button is pressed
      Officer has to fill the form and have to enter following details
      Complainant name
      Complainant Contact
      Date of crime
      Date of complaint
      Time of crime
      Details
      Type of crime
    End if
3.  If submit button is pressed
     If all information is entered and valid
      A) Information is submitted
      B) Required changes are made to database
      C) case is assigned to an Officer
      D) User is given a confirmation
      End if
4.  If all the information is not entered
      Then show the error
      And present the form again
     End if
5.  If logout button is pressed    Redirect to home page
```



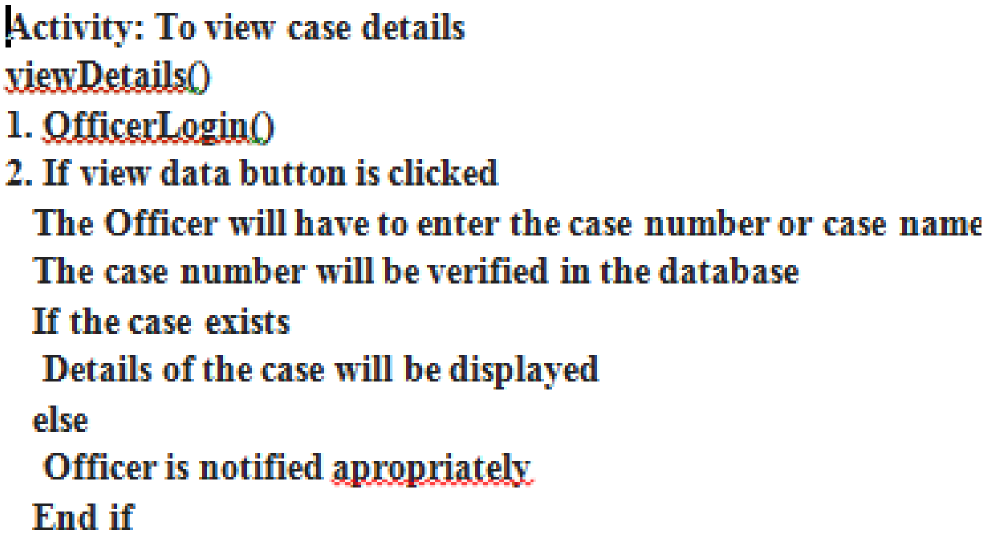
```
Activity: Assign the case
Admin()
1. Enter officer_id and password
2. If both are valid
     Admin will be directed to his account
   Else
     The login form will be presented to the Admin
    End if
3. If login is successful
    The Admin can view the added details and alot the case to any officer
    End if
4.   If logout is pressed
    Redirect to the home page
```



```
Activity: To update case details
UpdateCaseDetails:
1. OfficerLogin()
2. If update case details button is clicked
   The Officer can do the following things
   A) Add suspects
   B) Change the status of the case
   C) Add current
```



```
Activity: To add suspects to the case
addSuspects()
1. OfficerLogin()
2. UpdateCaseDetails()
3. If add Suspects button is clicked
   Officer has to provide the following
   A) Suspect name
   B) Suspect photo
  End if
4. If submit button is pressed
  If all information is enetered and is valid
   A) Information is submitted
   B) Information is verified in database to find the criminal records of the  suspec
   C) Database is updated accordingly
   D) Officer is given a confirmation
  End if
5.  If all the information is not entered
    Then show the error
    And present the form again
   End if
6.  If logout button is pressed
   Redirect to home page
```



```
Activity: To view case details
viewDetails()
1. OfficerLogin()
2. If view data button is clicked
   The Officer will have to enter the case number or case name
   The case number will be verified in the database
   If the case exists
    Details of the case will be displayed
   else
    Officer is notified apropriately
   End if
```

10.6 Software Requirements Specifications

10.6.1 Introduction

Purpose

The purpose of this project is to create a static application for the clients wherein the clients can register their firs and the cases are solved by the officers

Document Conventions

This document uses the following conventions.

- DB Database
- DDB Distributed Database
- ER Entity Relationship

Intended Audience and Reading Suggestions

This project is a prototype for the investigation bureau and it is restricted within the country. This has been implemented under the guidance of college professors. This project is useful for the officers and as well as to the common people who will register their cases and report the crime.

Product Scope

This project aims at building an efficient software wherein the common people can register their cases and the officers are aloted to the solve the cases and the case history as well as case status is available.

References

- https://krazytech.com/projects
- Fundamentals of database systems by ramez elmarsi and shamkant b.navathe

10.6.2 Overall Description

Product Perspective

A distributed investigation bureau database system stores the following information. Case details : it includes the case details ,the evidences ,the suspects and case history and case status.
Officer details :this includes the officer details
criminal records :this includes criminal details

Product Functions

The major functions of Investigation Bureau database system as shown in below entity–relationship model (ER)

User Classes and Characteristics

Users of the system should be able to register a case. The officer should be able to do the following functions:

- add a case
- view the evidences
- add evidences
- view case history
- add suspects

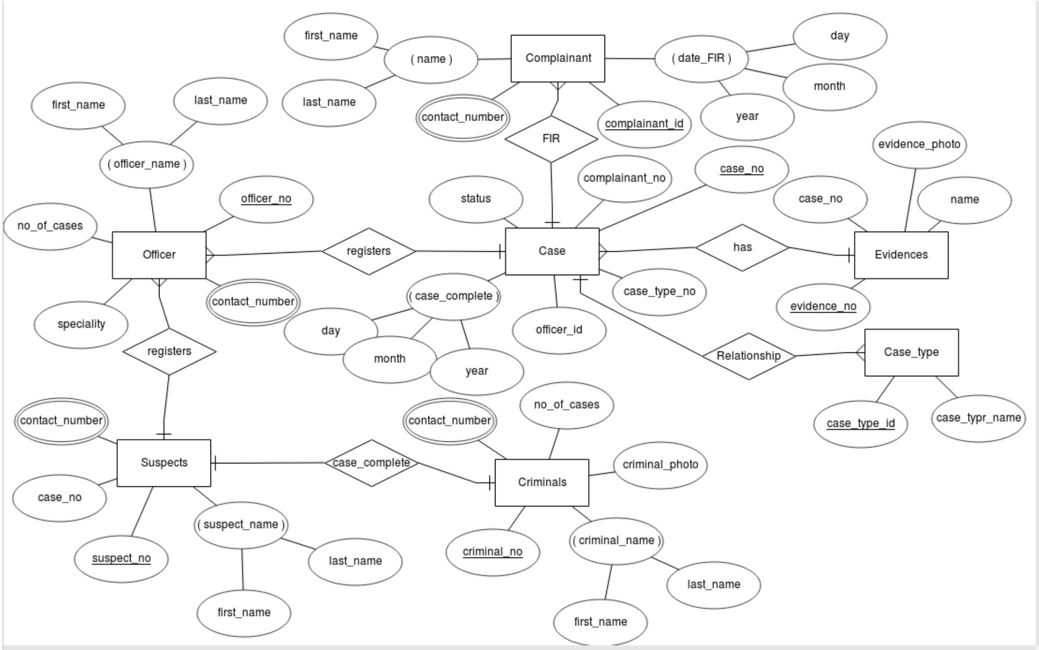


Figure 10.6: ER Diagram

- view criminal record

Operating Environment

Operating environment for the airline management system is as listed below.

- distributed database
- client/server system
- Operating system: mac.
- database: sql+ database
- platform: vb.net/Java/PHP

Design and Implementation Constraints

1. The global schema, fragmentation schema, and allocation schema.
2. SQL commands for above queries/applications

Software used	Description
Operating system	We have chosen Mac operating system for its best support and user-fri
Database	To save the case records, criminal records we have chosen SQL+ dat
java	To implement the project we have chosen java language for its more intera

Table 10.1: Software Interfaces

3. How the response for application 1 and 2 will be generated. Assuming these are global queries. Explain how various fragments will be combined to do so.

4. Implement the database at least using a centralized database management system.

User Documentation

A documentation of the software must be prepared.

10.6.3 External Interface Requirements

User Interfaces

- Front-end software: java
- Back-end software: SQL+

Hardware Interfaces

- Mac.
- A browser which supports CGI, HTML Javascript.

Software Interfaces

Following are the software used for the investigation bureau online application.

Communications Interfaces

This project supports all types of web browsers.

10.6.4 System features

System Feature 1

DESCRIPTION and PRIORITY
The investigation bureau system maintains information on cases and

- **STIMULUS/RESPONSE SEQUENCES**
- **FUNCTIONAL REQUIREMENTS**

Other system features include:

DISTRIBUTED DATABASE:
Distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases
.
CLIENT/SERVER SYSTEM
The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).
A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

10.6.5 Other Nonfunctional Requirements

Performance Requirements

1. The major features of airline database system as shown in below entity–relationship model (ER model)

2. **NORMALIZATION:**
   The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored. If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data

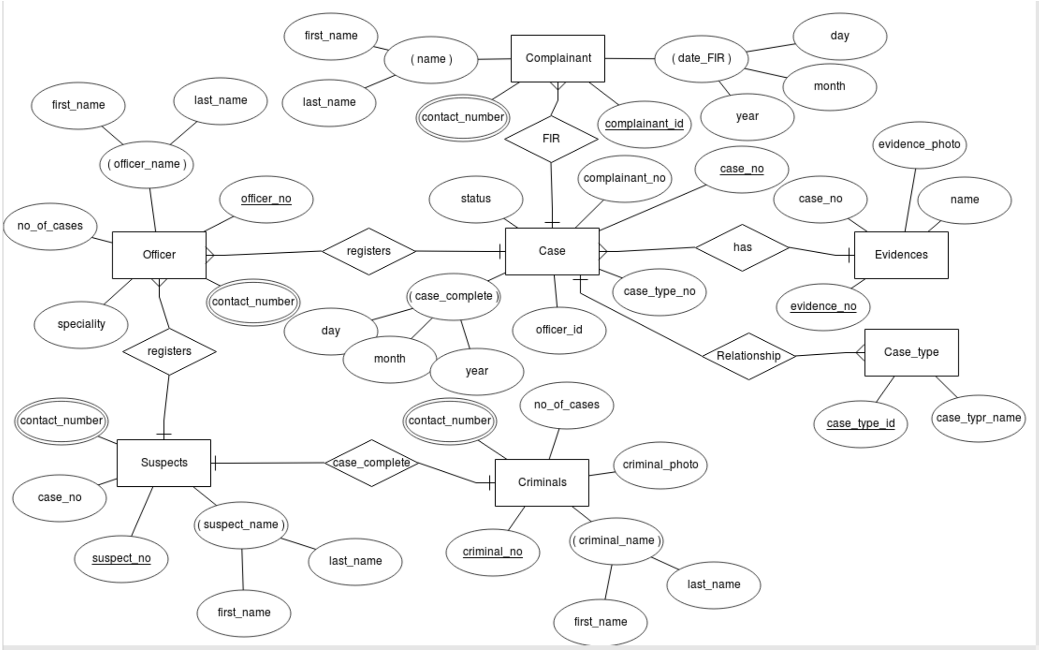


Figure 10.7: ER Diagram

is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database. Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

Security Requirements

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

Software Quality Attributes

- **AVAILABILITY:** The officer should be available for that particular case
- **CORRECTNESS:** The case must be solved correctly
- **MAINTAINABILITY:** The administrators and officers in chargers should maintain correct status and history of cases.
- **USABILITY:** Should be maximum

Chapter 11

Software Design

The software development phase consists of design, coding and testing.The development phase takes 75 % or more of the total software development cost. The aim of the design activity is to transform the requirements specified in the SRS(Software Requirements Specification) document in to a format that can be easily implemented using a suitable programming language or any implementation tool.

11.1 Data Design

Data design is the first and most important design activity.Here the main issue is to select the appropriate data structure. That is the Data design focuses on the definition of data structures. The data design describes structures that reside within the software. Attributes and relationships between data objects dictate the choice of data structures.
PRIMARY DATA:
SECONDARY DATA:

11.2 Architectural Design

The architecture design uses information flowing characteristics, and maps them into the program structure.

Attribute	DataType	Constraint	Description
Caseid	int	Primary	Case Id no
Casetype	varchar	Not null	The type of case to be registered
Officerid	int	Primary	Officer Id no
Complainant name	varchar	Not null	Name of the person who has lodged the FIR
Description	varchar	Not null	Contains the description of the case
Case status	varchar	Not null	Describes the case status

Table 11.1: Case

Attribute	DataType	Constraint	Description
Username	varchar	Unique	Stores the username of the admin
Password	varchar	Not null	Stores the password in hashformat

Table 11.2: Login_details

Attribute	DataType	Constraint	Description
Caseid	varchar	foreign	Stores the case id no
name	varchar	Not null	Stores the name of the suspect
contact	varchar	Not null	Stores the contact number of the suspect
address	varchar	Not null	Stores the address of the suspect
relation	varchar	Not null	Stores the relation of the suspect with the victim
note	varchar	Not null	Stores the important description of the suspect useful for the investigation
image$_suspect$	blob	Not null	Stores the photo of the suspect

Table 11.3: Suspect

Attribute	DataType	Constraint	Description
Caseid	varchar	foreign	Stores the case id no
Evidence$_type$	varchar	Not null	Stores the type of the evidence physical or logical
Evidence	varchar	Not null	Stores the evidence received at the crime site
suspect	varchar	Not null	stores the person who is more closely related to this evidence
image$_evidence$	blob	Not null	Stores the image of the evidence
note	varchar	Not null	Stores the important description of the evidence useful for the investigation
points	varchar	Not null	Stores the points for a particular evidence used to predict the suspect.

Table 11.4: Evidence

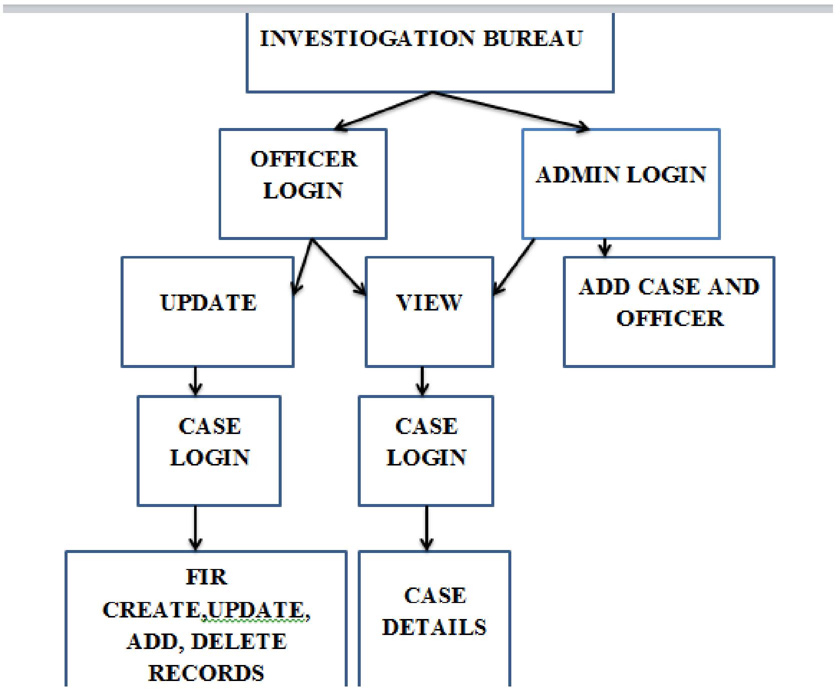


Figure 11.1: Architectural Design

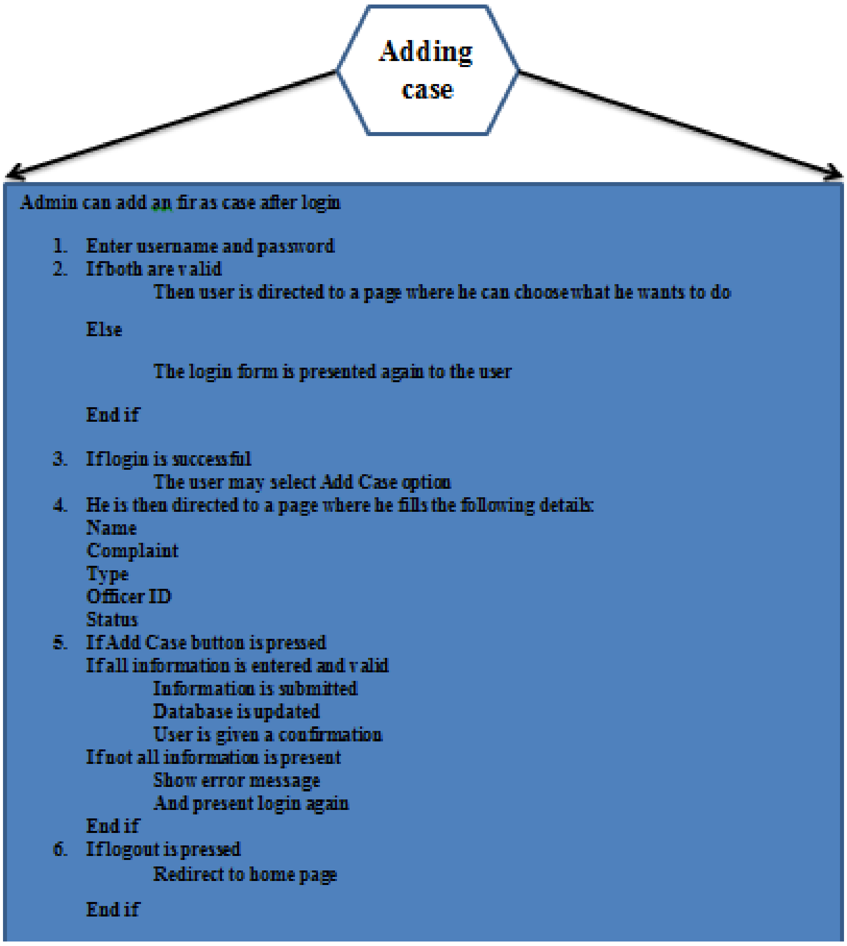
11.3 User Interface Design

User interface design (UID) or user interface engineering is the design of websites, computers, appliances, machines, mobile communication devices, and software applications with the focus on the user's experience and interaction. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals—what is often called user-centered design.

Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to itself. Graphic design may be utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface.

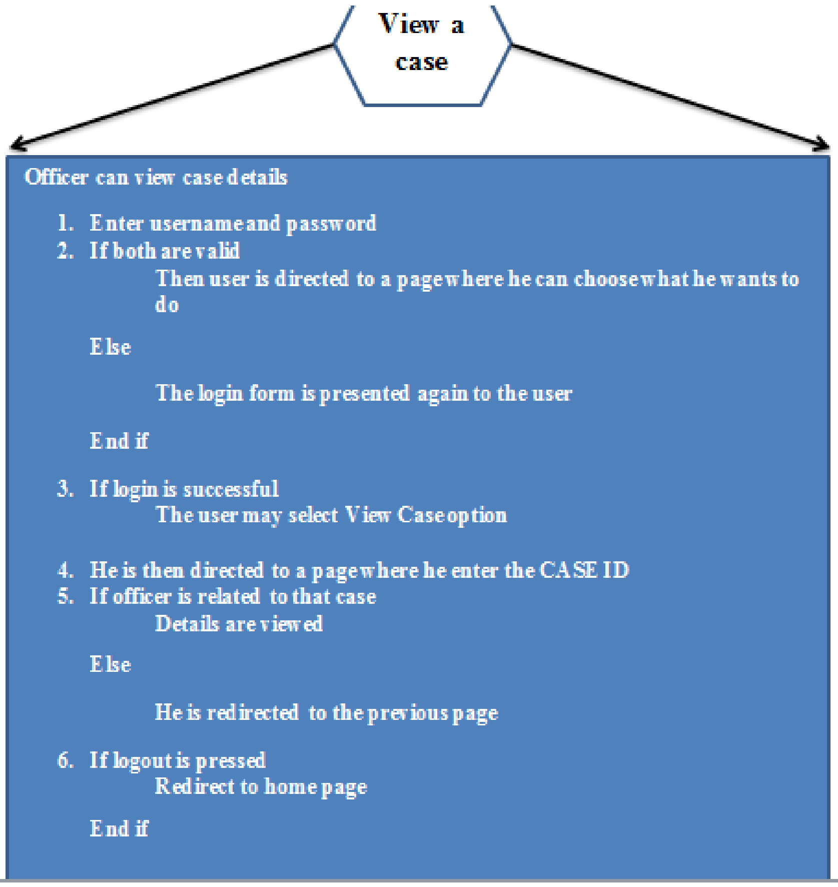
11.4 Procedural Design

The procedural design describes structured programming concepts using graphical, tabular and textual notations. These design mediums enable the de-



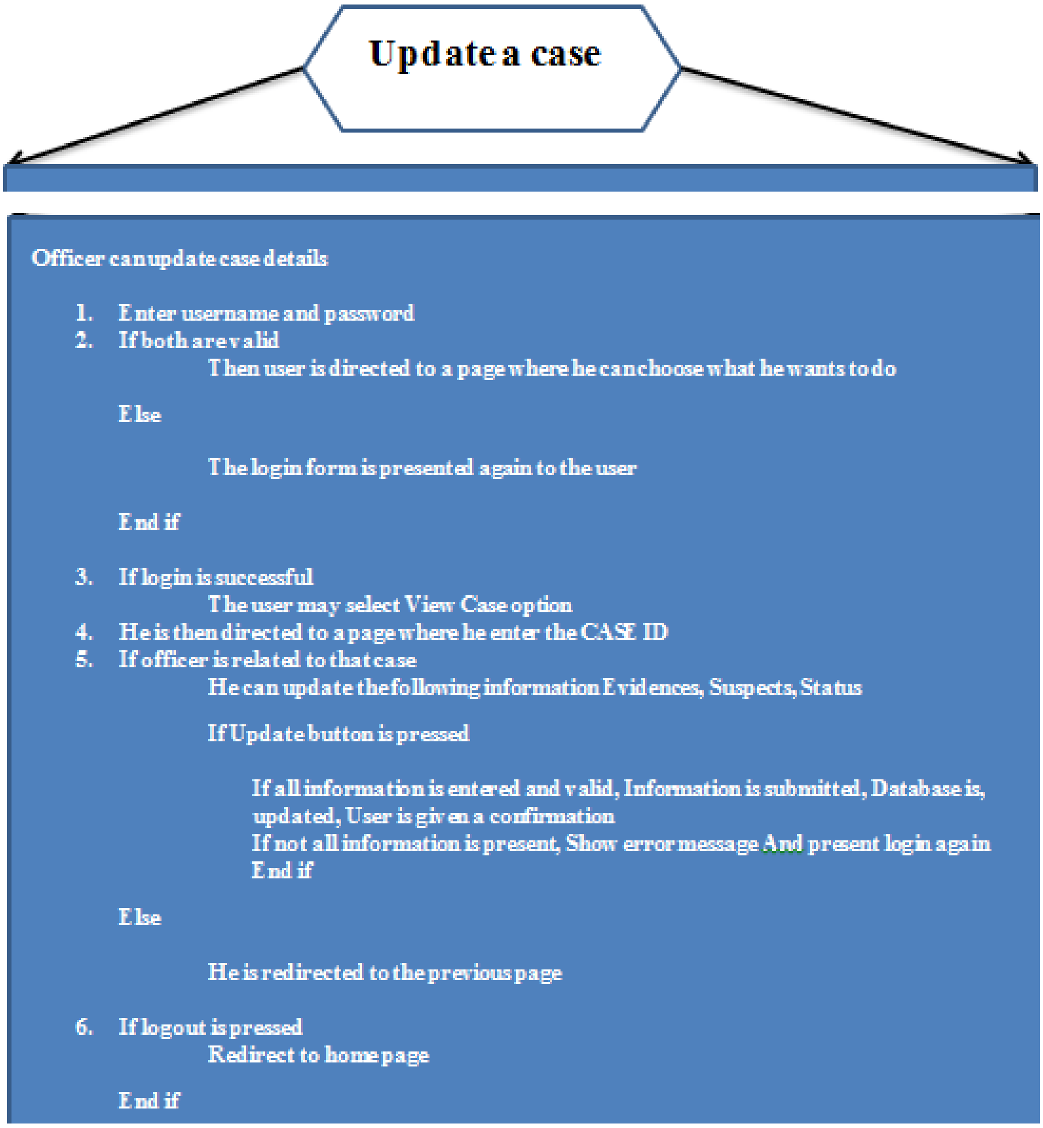
```
Admin can add an fir as case after login

  1. Enter username and password
  2. If both are valid
            Then user is directed to a page where he can choose what he wants to do

     Else

            The login form is presented again to the user

     End if

  3. If login is successful
            The user may select Add Case option
  4. He is then directed to a page where he fills the following details:
     Name
     Complaint
     Type
     Officer ID
     Status
  5. If Add Case button is pressed
     If all information is entered and valid
            Information is submitted
            Database is updated
            User is given a confirmation
     If not all information is present
            Show error message
            And present login again
     End if
  6. If logout is pressed
            Redirect to home page

     End if
```

Figure 11.2: PSPEC 1



```
Officer can view case details

    1. Enter username and password
    2. If both are valid
           Then user is directed to a page where he can choose what he wants to
           do
       Else
           The login form is presented again to the user
       End if
    3. If login is successful
           The user may select View Case option
    4. He is then directed to a page where he enter the CASE ID
    5. If officer is related to that case
           Details are viewed
       Else
           He is redirected to the previous page
    6. If logout is pressed
           Redirect to home page
       End if
```

Figure 11.3: PSPEC 2



```
Officer can update case details

    1.  Enter username and password
    2.  If both are valid
                Then user is directed to a page where he can choose what he wants to do
        Else
                The login form is presented again to the user
        End if
    3.  If login is successful
                The user may select View Case option
    4.  He is then directed to a page where he enter the CASE ID
    5.  If officer is related to that case
                He can update the following information Evidences, Suspects, Status
                If Update button is pressed
                    If all information is entered and valid, Information is submitted, Database is,
                    updated, User is given a confirmation
                    If not all information is present, Show error message And present login again
                    End if
        Else
                He is redirected to the previous page
    6.  If logout is pressed
                Redirect to home page
        End if
```

Figure 11.4: PSPEC 3

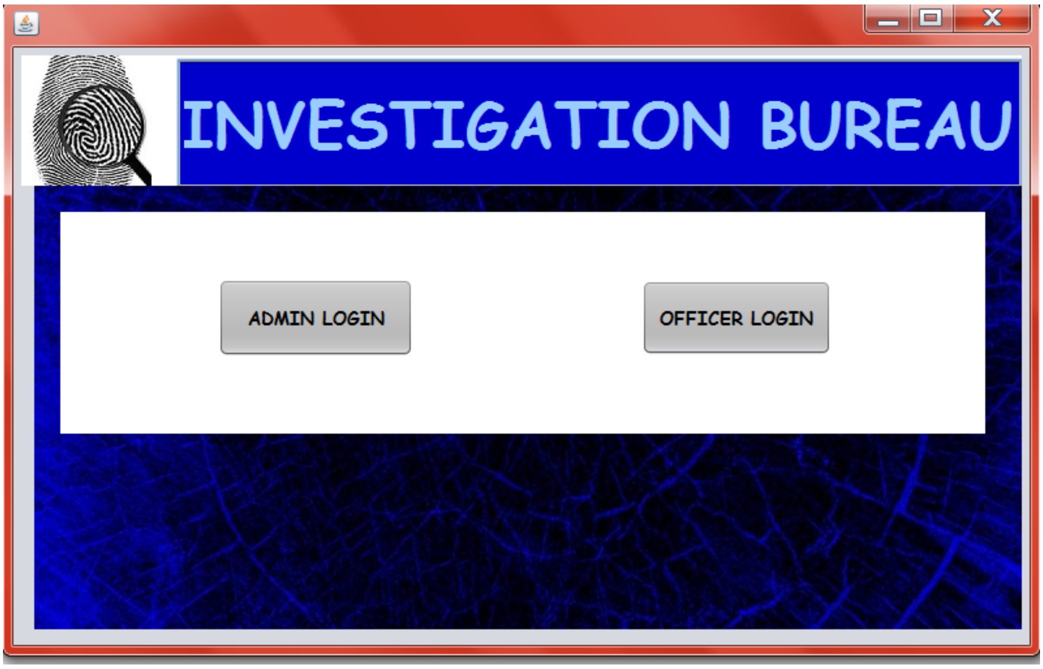
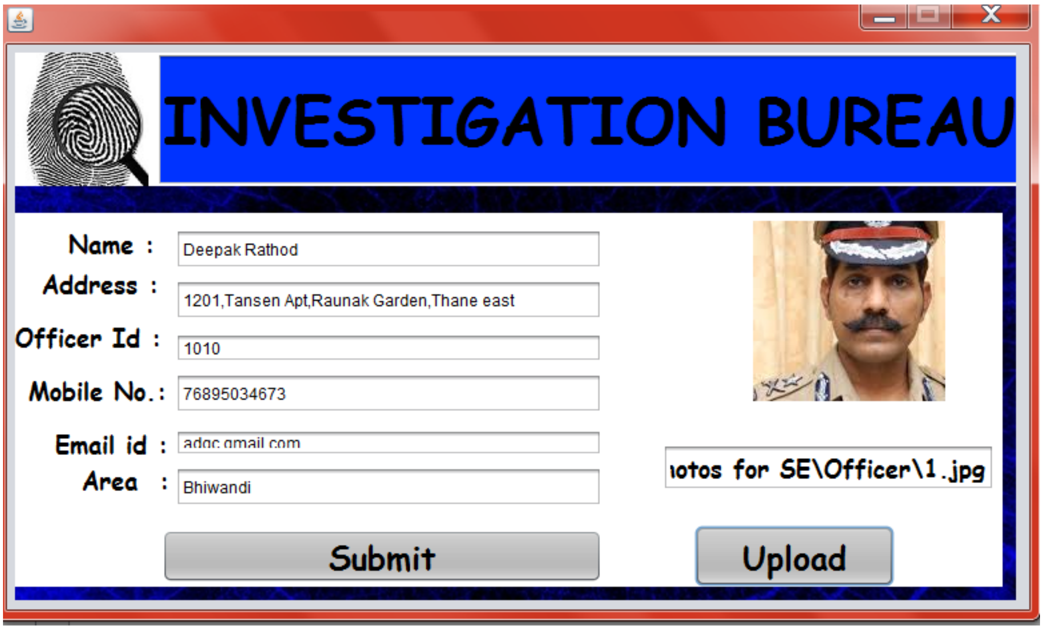


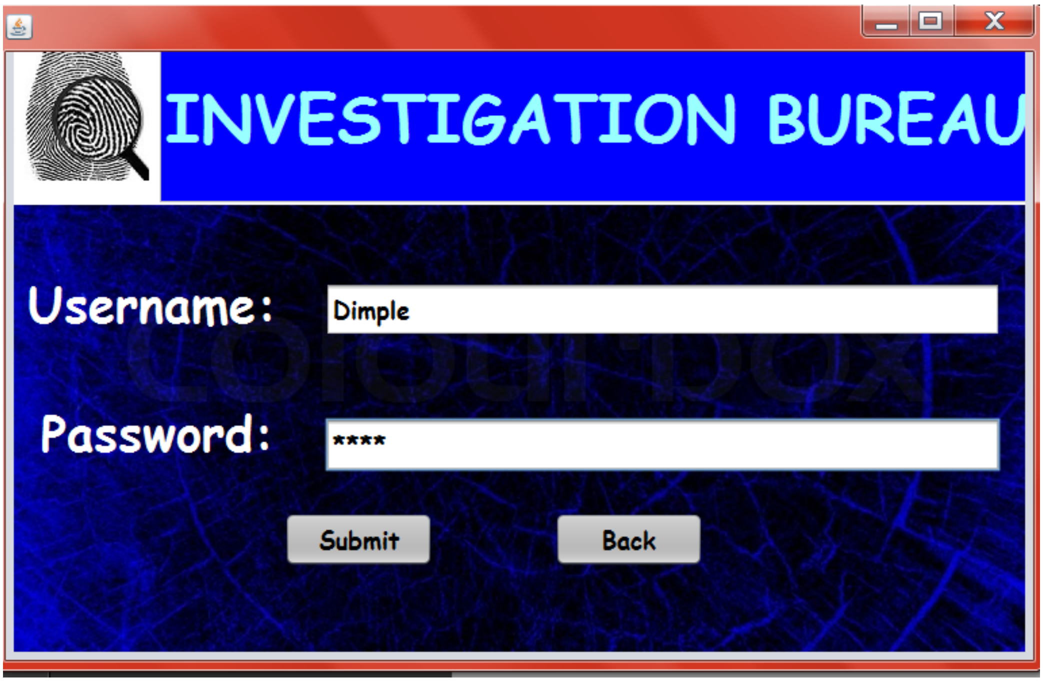
Figure 11.5: Welcome page



Name :	Deepak Rathod
Address :	1201,Tansen Apt,Raunak Garden,Thane east
Officer Id :	1010
Mobile No.:	76895034673
Email id :	adgc.gmail.com
Area :	Bhiwandi

ıotos for SE\Officer\1.jpg

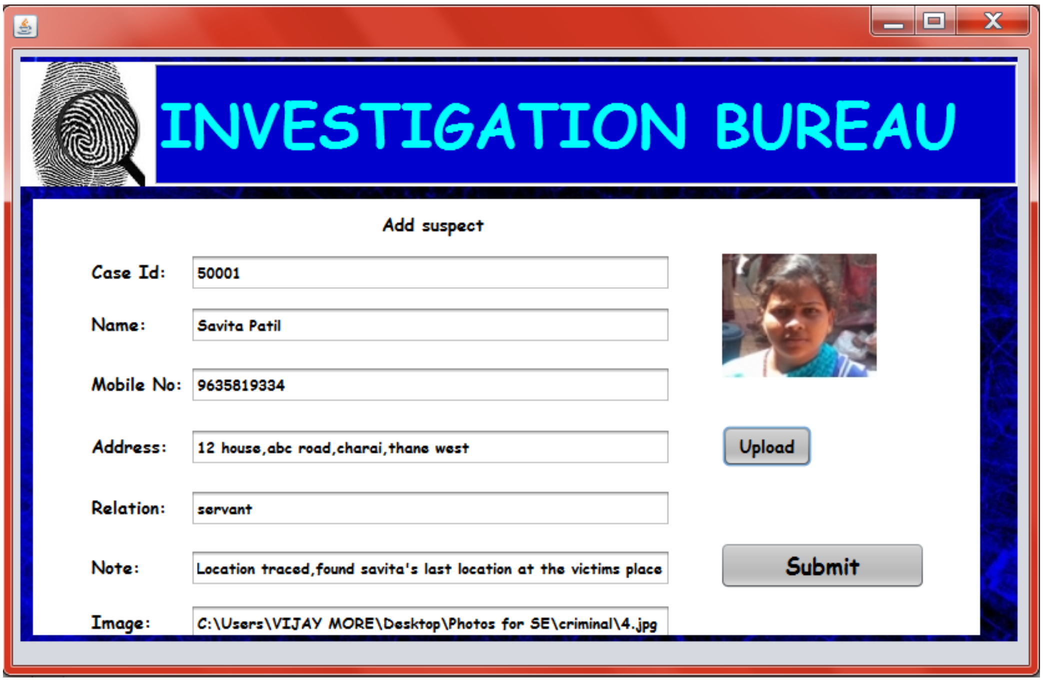
Figure 11.6: Add officer page



Username: Dimple

Password: ****

Figure 11.7: Admin Login Page



Add suspect

Case Id: 50001

Name: Savita Patil

Mobile No: 9635819334

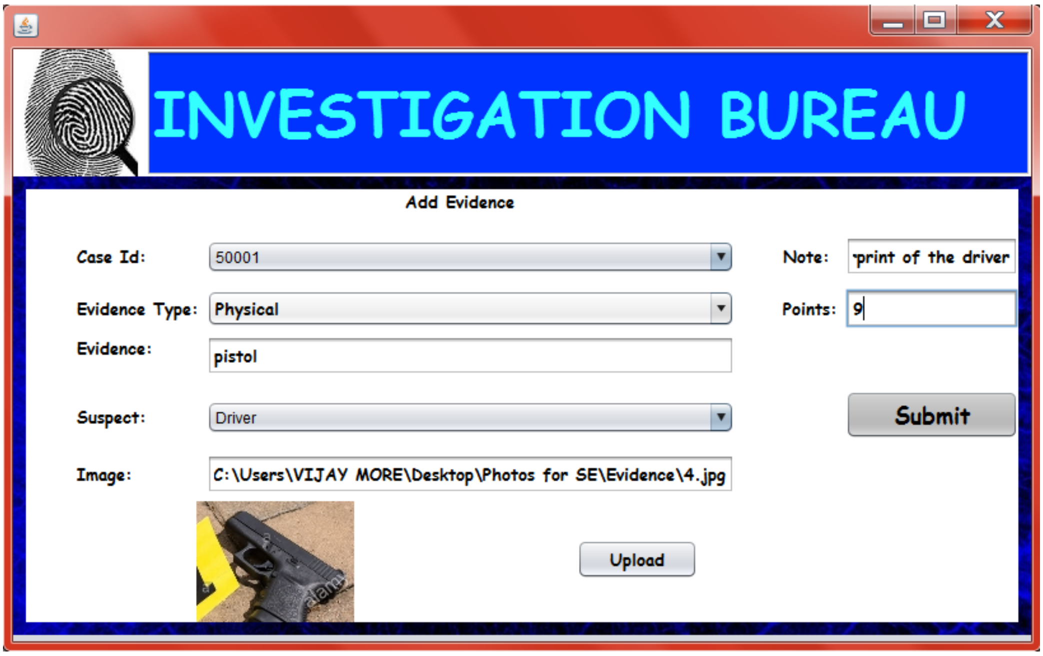
Address: 12 house,abc road,charai,thane west

Relation: servant

Note: Location traced,found savita's last location at the victims place

Image: C:\Users\VIJAY MORE\Desktop\Photos for SE\criminal\4.jpg

Figure 11.8: Add suspect page



Add Evidence

Case Id: 50001

Evidence Type: Physical

Evidence: pistol

Suspect: Driver

Image: C:\Users\VIJAY MORE\Desktop\Photos for SE\Evidence\4.jpg

Note: print of the driver

Points: 9

Figure 11.9: Add evidence page

signer to represent procedural detail, that facilitates translation to code. This blueprint for implementation forms the basis for all subsequent software engineering worked.

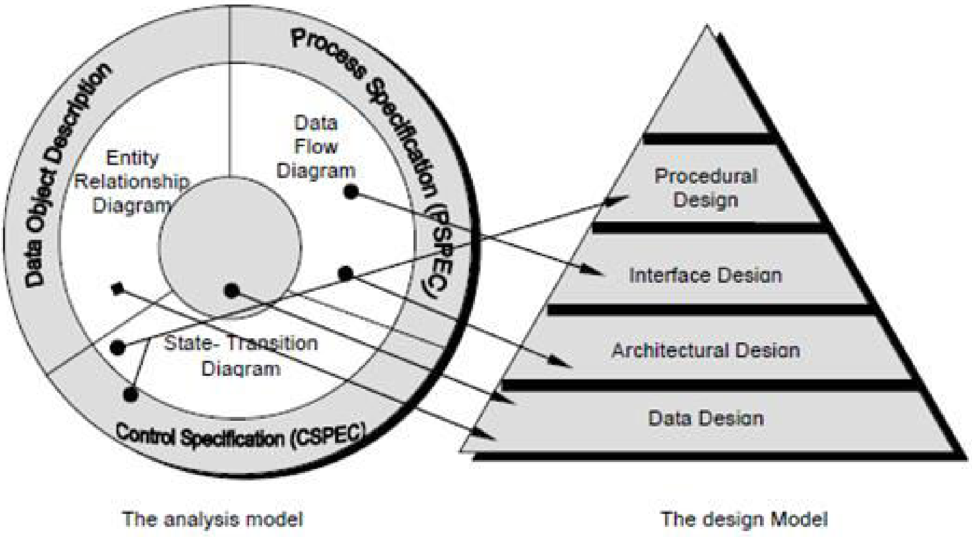


Figure Translating the analysis model into a software desing

Figure 11.10: Procedural Design

Chapter 12

Testing

12.1 White Box Testing:

White Box Testing is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box testing.

It is one of two parts of the "box testing" approach of software testing. Its counter-part, blackbox testing, involves testing from an external or end-user type perspective. On the other hand, Whitebox testing is based on the inner workings of an application and revolves around internal testing.

12.2 Procedure

1. **Step I:** Draw Flow Graph The main function i.e registration and login of admin and officers

   We number the different points in the function as follows:

   (a) Fill credentials
   (b) Credentials format invalid
   (c) Credentials format valid
   (d) Prompt for re-entry of credentials
   (e) Authenticate credentials
   (f) Login/Registration failed

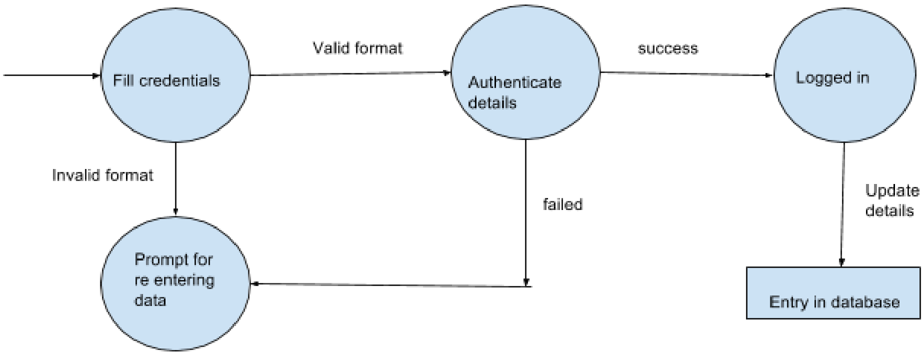


Figure 12.1: Testing Flow graph

(g) Login/Registration successful

(h) Store details in database

2. **Step II:** Determine independent paths
According to above points, we have following paths:
   - Path 1: 1 - 2 - 4
   - Path 2: 1 - 3 - 5 - 6
   - Path 3: 1 - 3 - 5 - 7 - 8

3. **Step III:** Prepare test case for each path

   (a) For path 1:
   One or more of the following conditions should be true:
   - ID is not 8 digit
   - ID contains non-numeric characters
   - Name contains non alphabetic characters
   - Password is less than 7 characters
   - All fields are not filled

   (b) For path 2:
   In this case, the ID or password must be incorrect so that the officer/admin will not be able to login. The credentials do not match with the database values.

   (c) For path 3:
   In this case,none of the conditions from case I and case II should be present.

Chapter 13

Conclusion

Investigation Bureau management is a very vital software which demands a lot of hard work and dynamism. As the name suggests, it means conceptualizing, planning, organizing an Investigation Bureau. As we know that with digitization comes a lot of data. The data in today's world has increased exponentially since the past decade. To manage this data in an ordered fashion we require some tools which compartmentalize data, making it easy to maintain. Our software does the task of helping investigation bureau officers to maintain their data in an ordered fashion and makes it easy for them to view, update and modify their data in a secure way.

Investigation Bureau management is the organizing and management of an investigation organization. This system simplifies the task of an officer in the bureau. An officer can register an FIR or update case semantics like updating evidences, looking up criminal records of suspects, etc.
In an existing system,
Whenever we implement new system it is developed to remove the shortcomings of an existing system. The computerized has more Edge over the manual system.As we are doing a project on "INVESTIGATION BUREAU MANAGEMENT". So firstly we will introduce the existing system, the existing system is based on manual system, which takes lot of time to get tasks done. It has the following disadvantages:
Wastage of time: -Adding of data in files or making documents takes up a lot of time. Also searching manually through records is time consuming and tiresome.
Inaccuracy: -Human error can lead to wrong data updation or missing of a significant detail leading to inaccuracy.
Loss of data: -Papers, documents, files can be lost. Evidences can be misplaced. All leading to loss of significant data.
While in our proposed system,

We have developed new system, which is based on computer in which the officer has to register on the computer by simply logging into the software. It is secure time saving system which . It has the following advantages:
Time Saving:-As all the data is documented, any thing can be looked for in a jiffy. For examples, a criminal can be looked for simply by searching by his name.
Accuracy:-The system stores the database so no information needs to be maintained. A secure mysql database is used, hence no inaccurate information is shown.