**Introduction**

* 1. **Company Profile**

WinSol Solutions Pvt. Ltd. strive to lead in the Innovation, development and maintenance of the industry's most advanced, state-of-the-art information technology solutions in the arena of software-based web & client-server solutions. We translate these technologies into value for our clients through creative, tailored solutions that best meet their needs. We have a team of Professional & Expert Web Designers, Web Developers and SEO in Pune, India, who provides you appropriate web services based on latest technology. The prime goal for WinSol Solutions is to provide dependable, high quality, and cost effective solutions using the state-of-the-art technology that consistently meet the client’s needs and expectations. All the products provide you a safer path to achieve the organizations objectives.

WinSol Solutions is an exciting, fast growing Software company with a family-friendly atmosphere, a strong dedication to customer satisfaction, and a drive to provide positive technical contributions to the nation’s most difficult technological problems. Our business operations and culture are anchored in our core values and characteristics.”

WinSol Solutions has been a client-focused organization since our earliest days. By placing clients at the heart of everything we do, we are able to deliver high value-added products and services that meet the needs of our clients and local communities, thereby contributing to social development and economic growth.

**1.2 Introduction of Project:**

Voter registration is the requirement in some democracies for citizens and residents to check in with some central registry specifically for the purpose of being allowed to vote in elections. An effort to get people to register is known as a voter registration drive.

The Government of India conducts an intensive revision of the voters list every 5 years. An additional summary revision is conducted every year. This overall system is having manual processing.

In order to resolve the above problem, we are implementing Electoral Roll Management System online which will reduce many manual efforts of people who are applying for voter id card through form 6.

* 1. **Existing System and Need for System**

**Existing System:**

* Difficult to maintain the information of enquiry.
* As it is manual system so it is very time consuming process.
* Very difficult to modify or update record.
* Not possible to generate quick report.
* Less security due to use of paper work.
* User never gets proper information.
* More physical interaction needed.

**Drawback of Existing System:**

* Lot of paper work adding to extra cost.
* Manual procedures slowing up the process.
* Too difficult to maintain all the details.
* No validations for data coming in and going out.
* Procedures to retrieve old data were very slow and difficult.

**Needs for System:**

To overcome all the above mentioned overheads a new system is needed. The system must provide:

* Due to online registration customer place online Gas Order
* Completely Automated Process.
* Greater Transparency.
* Home Delivery of voting card.
* Easy reports generation.
* Minimize Human errors.
* Minimal Storage Spaces.
  1. **Operating Environment**

**For Web Server –**

**Software Requirement:-**

Plat Form : Windows

Front End : JAVA (Struts)

Back End : MySQL

Web Server : Apache Tomcat 7.0

Report Tool : iReport

**Hardware Requirement: -**

Processor : P 4 & Above (1.3 GHz) / Higher

Hard Disc : 80 GB

RAM : 512 MB / Higher

**For Client –**

**Software Requirement:-**

Web Browser : IE 8.0, Mozilla, Firefox, Chrome

Operating System : Windows, Linux, Ubuntu

**Hardware Requirement: -**

Processor : P 4 & Above (1.3 GHz) / Higher

Hard Disc : 80 GB

RAM : 512 MB / Higher

* 1. **Detail Description of Technology used**

**Java**

Java is a general-purpose, concurrent, class-based, object-oriented computer programming language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to byte code (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture.

**Java Server Pages**

Java Server Pages (JSP) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. Released in 1999 by Sun Microsystems, JSP is similar to PHP, but it uses the Java programming language.

JSP may be viewed as a high-level abstraction of Java servlets. JSPs are translated into servlets at runtime; each JSP's servlet is cached and re-used until the original JSP is modified.

**Struts**

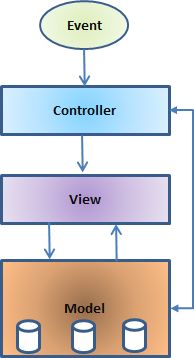
The Struts Framework is a standard for developing well-architected Web applications. Based on the Model-View-Controller (MVC) design paradigm, it distinctly separates all three levels (Model, View, and Control). It also offers the following features

**MVC Architecture**

Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts:

* **Model** - The lowest level of the pattern which is responsible for maintaining data.
* **View** - This is responsible for displaying all or a portion of the data to the user.
* **Controller** - Software Code that controls the interactions between the Model and View.

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows.



**The model**

The model is responsible for managing the data of the application. It responds to the request from the view and it also responds to instructions from the controller to update itself.

**The view**

A presentation of data in a particular format, triggered by a controller's decision to present the data. They are script based templating systems like JSP, ASP, PHP and very easy to integrate with AJAX technology.

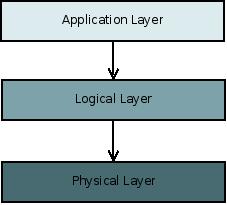
**The controller**

The controller is responsible for responding to user input and perform interactions on the data model objects. The controller receives the input, it validates the input and then performs the business operation that modifies the state of the data model.

**Introduction of MySql**

* MySQL is a database management system.
* MySQL is a relational database management system.
* MySQL software is Open Source.
* MySQL Server works in client/server or embedded systems.
* A large amount of contributed MySQL software is available.
* Widely used by web a3pplication developers, together with PHP and APACHE
* Multiple OS support

**MySQL Architecture**



* The application layer contains common network services for connection handling, authentication and security. This layer is where different clients interact with MySQL these clients can written in different API's:.NET, Java, C, C++, PHP, Python, Ruby, Tcl, Eiffel, etc...
* The Logical Layer is where the MySQL intelligence resides, it includes functionality for query parsing, analysis, caching and all built-in functions (math, date...). This layer also provides functionality common across the storage engines.
* The Physical Layer is responsible for storing and retrieving all data stored in “MySQL”. Associated with this layer are the storage engines, which MySQL interacts with very basic standard API's. Each storage engine has it strengths and weakness, some of this engines are MyISAM, InnoDB, CSV, NDB Cluster, Falcon, etc...

**Chapter 2: PROPOSED SYSTEM**

**2.1 Proposed System**

Voter registration is the requirement in some [democracies](http://en.wikipedia.org/wiki/Democracy) for citizens and residents to check in with some central registry specifically for the purpose of being allowed to [vote](http://en.wikipedia.org/wiki/Vote) in elections. An effort to get people to register is known as a [voter registration drive](http://en.wikipedia.org/wiki/Voter_registration_drive).

In democracies where [resident registration](http://en.wikipedia.org/wiki/Resident_registration) is in effect, voter registration is generally not required.

The Government of India conducts an intensive revision of the voters list every 5 years. An additional summary revision is conducted every year. Apart from this, citizens can request their inclusion in the Voters list by applying through form 6. If the application is valid, the applicant's name will get included in the list.

* Once, you open the website, you will find a new webpage displaying some information such as user name , password, officer contact list, online application , new user registration and forgot password.
* Now, you need to select new user registration option in order to apply voter ID card online.
* Once, you click on new user registration option, you will find a new webpage asking your mobile or contact number and email address or ID.
* Now, enter your active mobile number and email address and click on register option.
* Having entered your mobile number and email id, you will be sent a verification code on your given mobile number.
* Now, you need to submit a verification code on displaying webpage.
* Once, you submit the verification code, you need to choose precede option.
* Having selected proceed option, you will be given a form 6 application.
* Now, you need to fill this online application form with required information such as your name, date of birth, city or district name, place of birth, language and sex.
* Remember, having filled correctly verification code, you will find a new form 6 application webpage.
* Now, fill this application form with required certain information. Moreover, you also need to upload some documents such as your color passport-sized photograph, proof of id, proof of address and proof and age.
* Once, the application form is filled correctly, you need to submit it with attached required documents.
* Having submitted form 6, you will be provided an application id number. Remember, you can use this application id number for tracking your application status.

**2.2 Additional information**

However, there are some other important points that you need to remember while using Voter ID card online registration process. Stated below are points that you need to remember when you apply voter ID card online.

* If you have already got a voter id card for a particular constituency then you are not suppose to obtain a new card as having two cards is illegal in India.
* In case of changing address, you need to get a new voter id card for your new constituency.
* In case of any kind of error in your voter id card such as misspelled of name, wrong date of birth, wrong residence address etc, you can apply for correction by using a particular application form.

**Feasibility Study:**

The Feasibility Study is a process of evaluating the deciding factors to check whether proposed system is feasible or not. Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success. In its simplest term, the two criteria to judge feasibility are cost required and value to be attained. As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation.

The feasibility study is carried out in the following aspects:

1. Technical Feasibility

2. Economical Feasibility

3. Operational Feasibility

**Technical Feasibility:**

* The hardware and software required for the proposed system is present and available to the programmer.
* The proposed equipment has the capacity to hold the data required for using the new system.
* Proposed system will take care of reliability, data security, accuracy and ease of access.
* The proposed system is capable of incorporating new requirements of the client. The system can be expanded if required.

**Economical Feasibility:**

* Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action.
* Cost-based study: It is important to identify cost and benefit factors, which can be categorized as follows: 1. Development costs; and 2. Operating costs. This is an analysis of the costs to be incurred in the system and the benefits derivable out of the system.
* Time-based study: This is an analysis of the time required to achieve a return on investments. The future value of a project is also a factor.

**Operational Feasibility:**

* There are two aspects of operational feasibility of a system. First is that of the technical performance and other is acceptance.
* The proposed system is providing an attractive user interface to the Operator / end user, so he feels very easy to work onto it.
* Response to operator/end user is very fast and very good. The proposed system will be operationally feasible because of system works in specific way.

**FACT FINDING TECHNIQUE**

To study any system the analyst needs to do collect facts and all relevant information. The facts when expressed in quantitative form are termed as data. The success of any project is depended upon the accuracy of available data. Accurate information can be collected with help of certain methods/ techniques. These specific methods for finding information of the system are termed as fact finding techniques. Interview, Questionnaire, Record View and Observations are the different fact finding techniques used by the analyst. The analyst may use more than one technique for investigation.

**Interview**

This method is used to collect the information from groups or individuals. Analyst selects the people who are related with the system for the interview. In this method the analyst sits face to face with the people and records their responses. The interviewer must plan in advance the type of questions he/ she is going to ask and should be ready to answer any type of question. He should also choose a suitable place and time which will be comfortable for the respondent .   
The information collected is quite accurate and reliable as the interviewer can clear and cross check the doubts there itself. This method also helps gap the areas of misunderstandings and help to discuss about the future problems. Structured and unstructured are the two sub categories of Interview. Structured interview is more formal interview where fixed questions are asked and specific information is collected whereas unstructured interview is more or less like a casual conversation where in-depth areas topics are covered and other information apart from the topic may also be obtained.

**Questionnaire**

It is the technique used to extract information from number of people. This method can be adopted and used only by an skillful analyst. The Questionnaire consists of series of questions framed together in logical manner. The questions are simple, clear and to the point. This method is very useful for attaining information from people who are concerned with the usage of the system and who are living in different countries. The questionnaire can be mailed or send to people by post. This is the cheapest source of fact finding.

**Record View**

The information related to the system is published in the sources like newspapers, magazines, journals, documents etc. This record review helps the analyst to get valuable information about the system and the organization.

**Observation**

Unlike the other fact finding techniques, in this method the analyst himself visits the organization and observes and understand the flow of documents, working of the existing system, the users of the system etc. For this method to be adopted it takes an analyst to perform this job as he knows which points should be noticed and highlighted. In analyst may observe the unwanted things as well and simply cause delay in the development of the new system.

**2.3 User requirements:**

The requirements of the user are as follows:

**1.Authentications:** The Citizens initially is required to enter his/her user ID and password, based on which he will be authenticated for further rights.

**2. Data Management:** Officer must be able to maintain data about their polling.

**3. Report Generation:** The report will be generated on date, monthly wise.

**4. Search:** The Citizens can also search the voter detail and status of voter card. It would be improving performance of the system.

**5. Time compatibility:** System should not be time consuming

**6. Paper work:** System should lessen or remove paper work.

**7. Validation:** System must be properly validated. Wrong inputs must not be allowed within the system

**Chapter 3: ANALYSIS & DESIGN**

**3.1 DATA FLOW DIAGRAMS**

**Context Level Diagram**

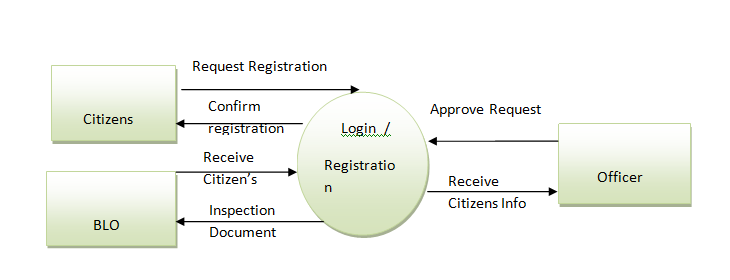
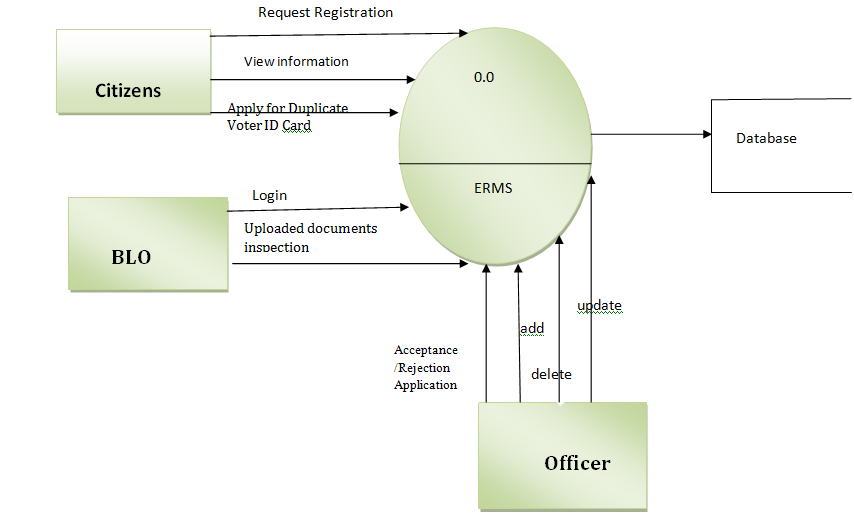
The context level diagrams are initially drawn followed by the levels of DFD’s. A context diagram is a top level (also known as 0) data flow diagram.

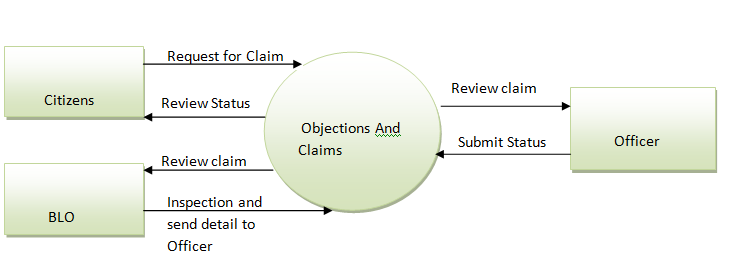
It only contains one process node (process 0) that generates the function of the entire system in relationship to external entities.

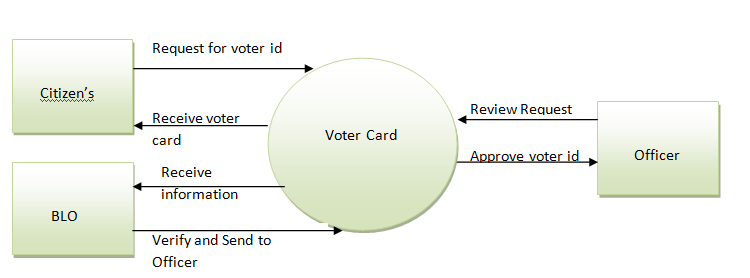
The first level DFD shows the main process can be broken into further processes until you reach pseudo code.

Symbols used for DFD’s:

|  |  |  |
| --- | --- | --- |
| Symbols | Meaning | Description |
|  | External Agent | An external entity is a source or destination of a data flow which is outside the area of study. |
|  | Data Flow | A data flow shows the flow of information from its source to its destination. A data flow is represented by a line, with arrowheads showing the direction of flow. Information always flows to or from a process and may be written, verbal or electronic. |
|  | Process | A process shows a transformation or manipulation of data flows within the system. |
|  | Data Store | A data store is a holding place for information within the system. It is represented by an open ended narrow rectangle. |

****

****

****

**3.2 Entity Relation Diagram:**

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

There are three basic elements in ER models:

Entities are the "things" about which we seek information.

Attributes are the data we collect about the entities.

Relationships provide the structure needed to draw information from multiple entities.

Lines linking attribute to entity sets and entity sets to relationship sets

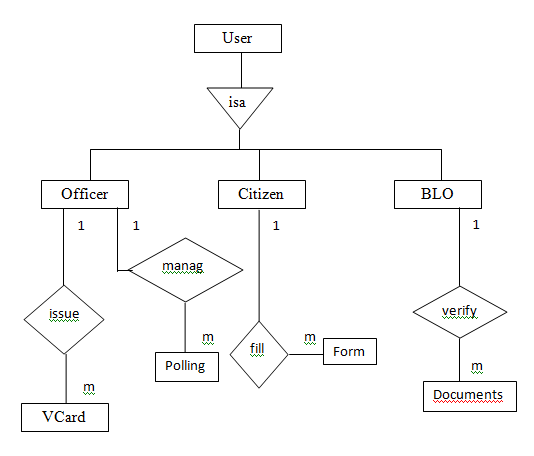
**Entities** : these are usually nouns used in descriptions of the system, in the discussion of business rules, or in documentation; identified in the narrative

**Relationships**: these are usually verbs used in descriptions of the system or in discussion of the business rules (entity \_\_\_\_\_\_ entity); identified in the narrative

**Attributes to the relations**: these are determined by the queries, and may also suggest new entities, e.g. grade; or they may suggest the need for keys or identifiers.

The *cardinality* defines the relationship between the entities in terms of numbers. An entity may be *optional*: for example, a sales rep could have no customers or could have one or many customers; or *mandatory*: for example, there must be at least one product listed in an order. There are several different types of cardinality notation; *crow's foot notation*, used here, is a common one. In crow's foot notation, a single bar indicates *one*, a double bar indicates *one and only one* (for example, a single instance of a product can only be stored in one warehouse), a *circle* indicates zero, and a *crow's foot* indicates many. The three main cardinal relationships are: one-to-one, expressed as 1:1; one-to-many, expressed as 1: M; and many-to-many, expressed as M: N.

**Entity Relation Diagram**:



Manage

**3.3 UML Diagrams**

**3.3.1 Use Case diagram:**

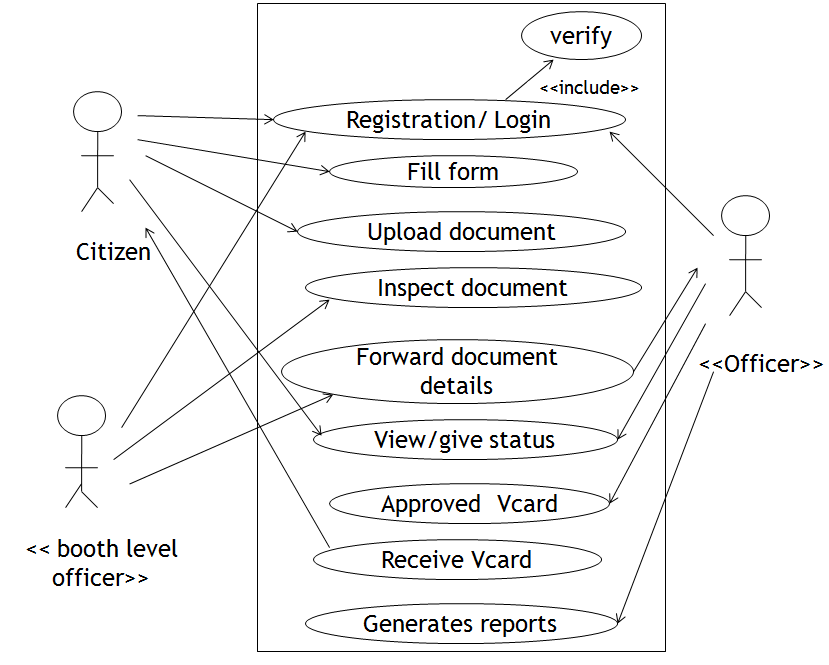
A use case diagram depicts participation relationships between actors and use cases. This diagram contains three basic components:

* System
* Actor
* Use case

**Notations used in Use Case diagram:**

|  |  |  |
| --- | --- | --- |
| Tool Name | Notation | Description |
| System |  | System boundary where all use cases will reside. |
| Actor |  | Actor which uses the system. |
| Use Case |  | It is generally function or process in the system. |
| Association |  | It shows direct association between actor and use cases. |
| Dependency |  | It shows dependency between two use cases. |
| Includes |  | It means one use case includes another use case. |

**Use Case for Complete System:-**



**Use Case Diagram for Citizens:**

Use case for citizen

Citizen

Registration

Login

Fill form

Upload Documents

Check status

verify

Received VCard

«

i

n

c

l

u

d

e

»

**Use Case Diagram for BLO:**

Use case for BLO

BLO

Login

Verify Documents

Verify Forms

Send

Mail/Message

verify

System

Door to door

verification

«

i

n

c

l

u

d

e

»

Forward

Documents

details

**Use Case Diagram for Officer:**

Use case for officer

Officer

Login

Add Booth

Verify Documents

details

Assign BLO

verify

System

Approved VCard

«

i

n

c

l

u

d

e

»

Delete Booth

Generate Reports

**3.2 Class Diagram:**

A **class diagram** in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.

**Notations used in Class Diagram are as follows:**

|  |  |  |
| --- | --- | --- |
| Tool Name | Notation | Description |
| Class |  | A *class* defines a collection of similar instances. It exists at compilation time and serves as a type. It defines the interface and implementation of its instances. |
| Object |  | An object is a particular instance of a class. Each object represents a particular instance of something in the problem or solution domain and is created as needed. |
| Association |  | An association is a relationship between classes in a class diagram. |
| Aggregation |  | The aggregation relationship is a special form of association used to model the whole-part or part-of relationship. |
| Specialization & Generalization |  | Specialization is a relationship between classes. Specialization is the is-a-kind-of relationship, in which the specialization is the subclass, or subtype and the generalization is the super class or super type. |

**Class diagram:**

User

user\_id : string

password :string

name : string

address : string

Ph\_No : number

email\_id : string

create\_user()

del\_user()

Citizen

citizen\_id: string

app\_no : number

dob : date

gender : string

register()

chk\_status()

upload\_doc()

fill\_form()

submit\_form()

Officer

officor\_id: string

address : string

ph\_no : number

add\_booth()

add\_blo()

assign\_blo()

issue\_card()

reject\_card()

BLO

blo\_id : string

address:string

area : string

verify\_form()

verify\_doc()

give\_dtl()

reject\_app()

A

\*

\*

Form

form\_no : number

type : string

add()

delete()

modify()

transfer()

VCard

v\_id : string

add()

delete()

modify()

transfer()

Booth

booth\_no : number

part\_no : number

area : string

add()

delete()

1

1

\*

1

1

\*

1

1

\*

1

\*

apply

fill

\*

allocate

assign

issue

**3.3 Sequence Diagram:**

An interaction diagram is graphical representation of how objects interact with one another in a scenario. Objects communicate in an interaction diagram by sending messages. A Sequence diagram is organized temporally with the focus on the order in which messages are sent between objects.

**Notations used in Sequence diagram are as follows:**

|  |  |  |
| --- | --- | --- |
| Tool Name | Notation | Description |
| Object |  | Object can be an actor, class or physical entity. |
| Synchronous message |  | When one object sends this message & waits for the response or completion of process. |
| Asynchronous message |  | When one object sends this message & does not wait for response or completion of process. |
| Return message |  | It is used to give response for a message call. |
| Found message |  | It is used when the sender object is not known or not significant. |
| Lost message |  | It is used when the receiver object is not known or not significant. |
| Create message | <<create>> | When new object is created in the process. |
| Destroy message | <<destroy>> | When an object is destroyed in the process. |
| Conditional message | [if book<2]issuebook | Message along with the condition. |

**Sequence Diagram:**

Citizen

Registration

BLO

Officer

System

Regis Successful

Login

Login successful

Select Form

Fill Form

Upload Document

Login

login successful

Verify form

Verify documents

Send Details

l

o

g

i

n

l

o

g

i

n

s

u

c

c

e

s

s

f

u

l

C

h

e

c

k

D

o

c

Issue VCard

**3.4 Activity Diagram:**

An activity diagram is a diagram that shows activities and actions to describe workflows. In the Unified Modeling Language an activity diagram represents the business and operational step-by-step workflows of components in the system. An activity diagram shows the overall flow of control. Activity diagram are typically used for business process modeling. They consist of:

* Initial node
* Activity final node
* Activities

The starting point of the diagram is the initial node and the activity final node is the ending. An activity diagram can have zero or more activity final nodes. In between activities are represented by rounded rectangles.

**Notations used in Activity diagram:**

|  |  |  |
| --- | --- | --- |
| Tool Name | Notation | Description |
| Initial State |  | The initial state of the activity. |
| Action State |  | Represents the activity. |
| Decision |  | One activity conditionally follows another activity. |
| Synchronization |  | Multiple activities either follow or precedes synchronization bar. |
| Transition |  | Shows the flow between activities. |
| Signal send state |  | Used to send the signals. |
| Flow final |  | Shows the final flow of activity. |
| Final state |  | Shows the final state i.e. end of activity. |

**Activity Diagram for BLO:**

Login

Receive

Documents

Verify

Documents

Send details

to Officer

Activity for BLO

accept

reject

Invalid

valid

**Activity Diagram for Officer:**

Login

Receive

Details

Add booth

Delete booth

Assign BLO

Give status

Approve VCard

Receive VCard

Activity for Officer

Accept

Reject

**Activity Diagram for Citizen:**

Registration

Login

Apply for

VCard

Apply for

Transformation

Apply for

Modify

Apply for

Deletion

Fill Form

Upload

Documents

Check status

Receive VCard

accept

reject

Activity for Citizen

accept

reject

**3.5 Component Diagram:**

A component diagram describes the organization of the physical components in a system

**Notations used in Component diagram are as follows:**

|  |  |  |
| --- | --- | --- |
| Tool Name | Notation | Description |
| Component | Component | A component is a physical building block of the system. It is represented as a rectangle with tabs.. |
| Associations Roles | Interface | An interface describes a group of operations used or created by components |
| Dependencies | Dependencies | Draw dependencies among components using dashed arrows. |

**Component Diagram:**

Registration Schema

« Struts »

« Database »

« Registration »

Registration

Java

Lib.xml

Registration

Jar

Strut.dtd

« File »

« Library »

« File »

« File »

**3.6 Deployment Diagram:**

The deployment diagram shows how a system will be physically deployed in the hardware environment.

Its purpose is to show where the different components of the system will physically run and how they will communicate with each other. Since the diagram models the physical

runtime, a system's production staff will make considerable use of this diagram.

**Notations used in Deployment diagram are as follows:**

|  |  |  |
| --- | --- | --- |
| Tool Name | Notation | Description |
| Component | Node | A node is a physical resource that executes code components. |
| Components and Nodes | Association | Association refers to a physical connection between nodes, such as Ethernet. |
| Dependencies | Components and Nodes | Place components inside the node that deploys them. |

**Deployment Diagram:**

<<client WorkStation>>

<<Virtual

Machine sun

jre1.5>>

<<Web

browser

IE6,Firefox>>

<<Server o.s:Windows>>

<<Web Server

Apache>>

<<Database

>>

<<Database Server>>

{MYSQL}

H

T

T

P

S

Q

L

Deployment Diagram for

Electoral Roll Management

**3.7 Web Sitemap Diagram**

A sitemap is a list or diagram which represents the hierarchical structure of the html pages in a website.

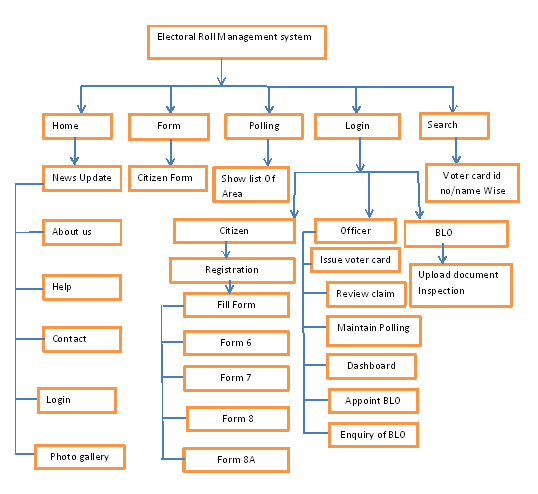
**Purpose of a Sitemap:**

A sitemap is a website design planning tool. It is used to:

Map out the site architecture\*

* + - Structure
    - Navigation
    - Page hierarchy;
  + Categorise the site content into logical groups, which will have meaning for the user;

Organise the order of the pages of the site, to create logical paths (so that targeted users achieve the purpose of their visit.



**3.8 Module Specifications**

**1) Admin Login**

Can read & write information about any center and can create, update, delete center as per the requirement and implementation plans.

* Add
* Update
* Delete
* Acceptance/ Rejection of Application

**2) User Login**

Can only read the various reports and can only get the information about the particular center and vote casting.

* Online Registration
* View Information
* Apply for [Duplicate Voter ID Card](http://www.voteridcard.net.in/duplicate-voterid-card.htm)

**3) Inspection Details**

* Uploaded Documents Inspection

**4) Application Status/Verification Details**

**5) Search By**

a) Name

b) Id Card

**6) CLAIMS AND OBJECTIONS**

* Addition
* Deletion
* Modification
* Transposition

**7) Reports Generation**

a) Polling Station Wise Report

b) Area /Village Wise Report

c)Daily, Monthly, Yearly Report

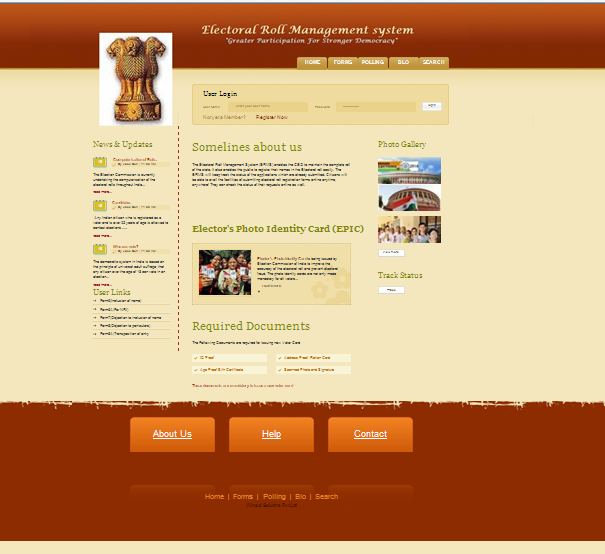
* Acceptance
* Rejection
* Pending

**8) *Mobile (SMS) Services***

* Know your Status (in Electoral Rolls) –
* Who is my BLO (Booth Level Officer)? –
* Where is my Polling Station? –

**3.9 User Interface Design ( Screens )**

1. Home and Login Page



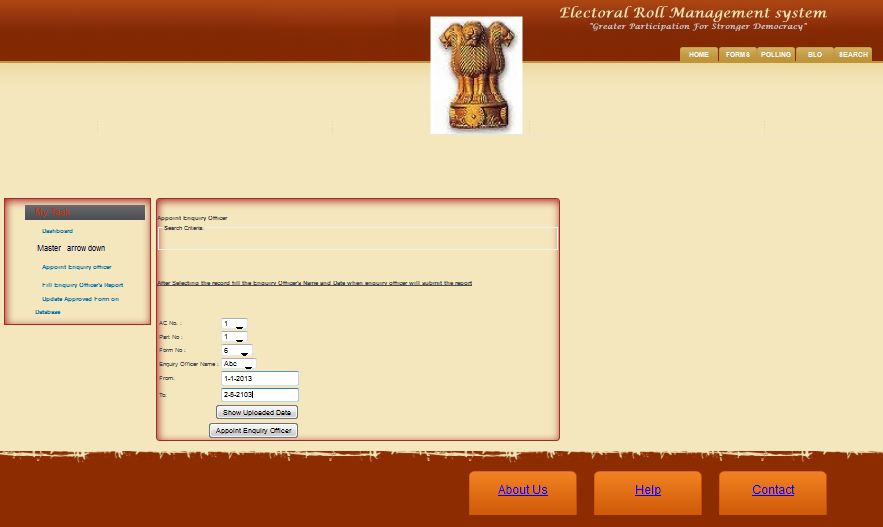
2.User Registration



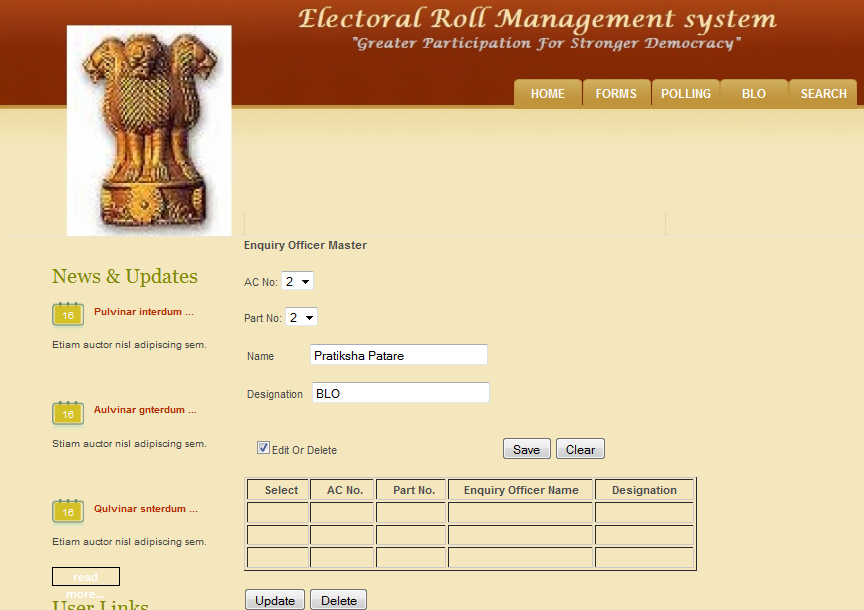
3. About Us Page



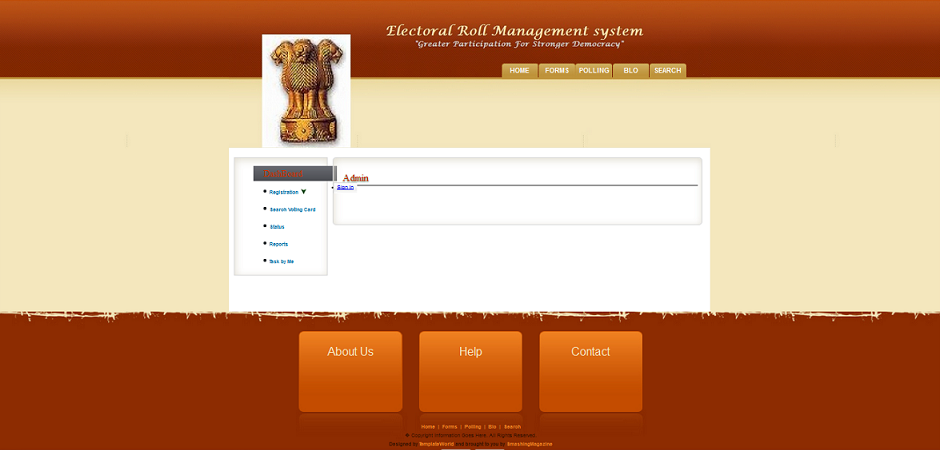
4 Appoint Enquiry Officer



5. Enquiry Officer Master



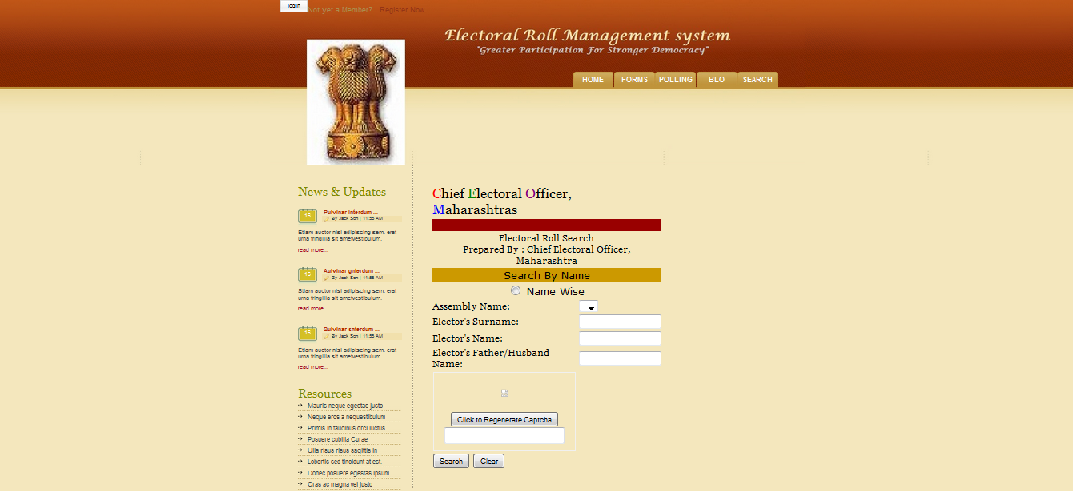
**6** . Admin Dashboard



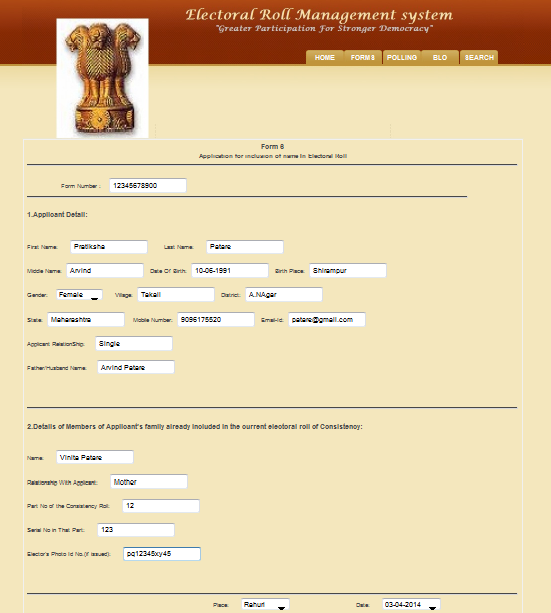
7. Electoral Roll search by ID



8. Electrol Roll Search by Name



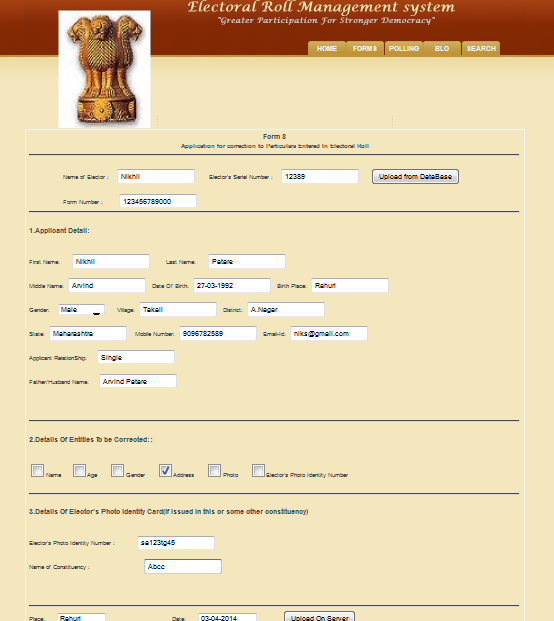
9 . Form No.6



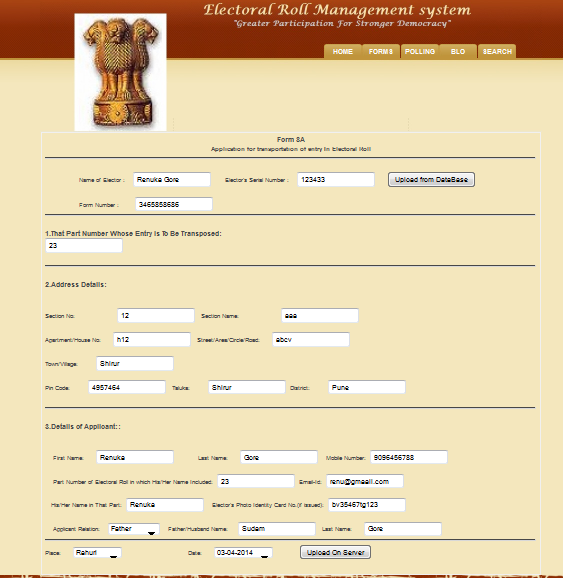
10. Form No 7



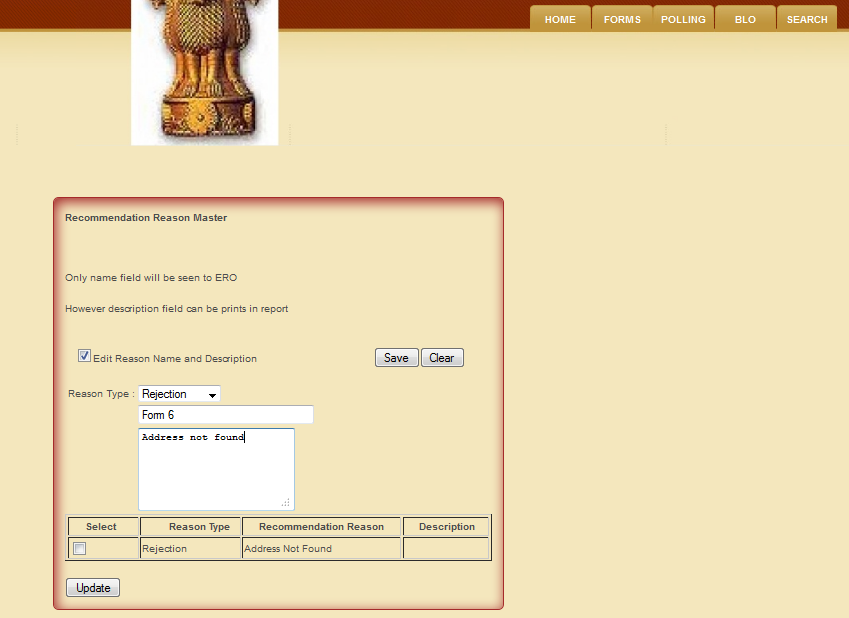
11 . Form No. 8



12 . Form No. 8A



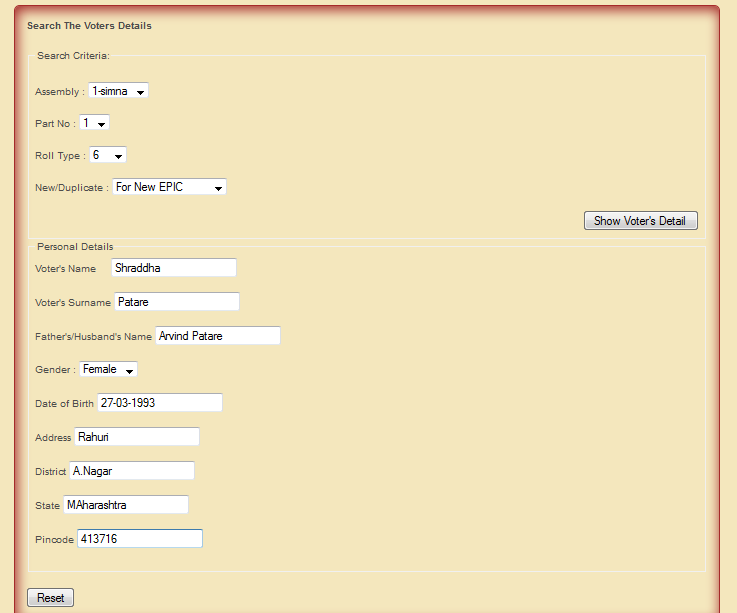
13 . Recommendation Reason Master



14 . Removal of incorrect approval



15. Search Voter Details



* 1. **Table Specification:**

1. **Table Name : citizen**

Description : The table contains all information about the citizen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** | **Constraints** |
| Reg\_id | Varchar | 30 | Applicant registration id | Foreign key |
| C\_fname | Varchar | 30 | Applicant first name | Null |
| C\_mname | Varchar | 30 | Applicant middle name | Null |
| C\_lname | Varchar | 30 | Applicant last name | Null |
| Username | Varchar | 30 | User name of citizen | Not null |
| Password | varchar | 30 | Password | Not null |
| C\_phone no | Integer | 10 | Citizen phone no | Null |
| DOB | Date | 30 | Citizen DOB | Null |
| Place\_of\_birth | Varchar | 50 | Citizen DOB place | Null |
| Village | Varchar | 50 | Citzen village | Null |
| District | Varchar | 50 | Citzen District | Null |
| State | Varchar | 50 | Citzen State | Null |
| Country | Varchar | 50 | Citizen country | Null |
| Language | Varchar | 50 | Citizen language | Null |
| Sex | Varchar | 10 | Sex of citizen | Null |
| Form No. | Integer | 20 | Form Number | Foreign key |
| Pincode | Integer | 40 | City pincode | Null |
| Email-id | Varchar | 50 | Citizen email id | Null |
| Occupation | Varchar | 50 | Citizen occupation | Null |
| Relationship | Varchar | 20 | Relationship | Null |
| Ac\_no | Integer | 20 | Assembly Constituency number |  |
| Sect\_no | Integer | 20 | Sector Number | Null |
| Sect\_name | Varchar | 20 | Sector Name | Null |
| House\_no | Integer | 20 | House/Apartment No | Null |
| Street\_ name | Varchar | 50 | Street/area/road name | Null |
| Passp\_no | Varchar | 20 | Passport Number | Null |
| P\_issue\_place | Varchar | 50 | Passport issue place | Null |
| P\_issue\_date | Date | 50 | Passport issue date | Null |
| P\_expiry\_date | Date | 50 | Passport expiry date | Null |
| Visa\_no | Varchar | 20 | Visa Number | Null |
| Visa\_type | Varchar | 50 | Visa type | Null |
| V\_issue\_date | Date | 30 | Visa issue date | Null |
| V\_expiry-date | Date | 30 | Visa expiry date | Null |
| V\_authority | Varchar | 30 | Visa authority | Null |

2**. Table Name : Officer**

Description : The table contains all information about the Electoral Roll Officer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | Description | **Constraints** |
| Officer\_id | Integer | 20 | Admin Id | Primary key |
| Officer name | varchar | 30 | Admin \_name | Null |
| Post\_id | Integer | 30 | Officer’s post Id | Null |
| Ac\_No | integer | 20 | Assembly Constituency no | Null |
| Constituency | varchar | 30 | Constituency name | Null |

3. **Table Name :BLO**

Description: The table contains all information about the Booth Level Officer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | Description | **Constraints** |
| blo\_id | Integer | 20 | BLO Id | Primary key |
| Blo\_name | varchar | 30 | BLO name | Null |
| M\_name | Varchar | 30 | Citizen Middle Name | Null |
| L\_name | Varchar | 30 | Citizen Last Name | Null |
| Ac\_no | Integer | 30 | BLO’s Ac no | Null |
| Gender | Varchar | 10 | Gender of BLO | Null |
| Ac\_No | integer | 20 | Assembly Constituency No | Null |
| Addr | Varchar | 100 | Citizen Address | Null |
| City | Varchar | 30 | Citizen City | Null |
| State | Varchar | 30 | Citizen State | Null |
| Pincode | Integer | 20 | Citizen Pincode | Null |
| Username | Varchar | 30 | User name | Null |
| Password | Varchar | 30 | Password | Null |

4**. Table Name : Form**

Description: The table contains all information about different type of forms

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Form** | **Data type** | **Size** | **Description** | **Constraint** |
| Form\_no | Integer | 20 | Form no to be filled | Primary key |
| Reg\_id | Varchar | 20 | Registration\_id | Foreign key |
| Description | Varchar | 100 | Detail about form | Null |

5. **Table Name : Polling**

Description : The table contains all information about the Polling station .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Data type** | **Size** | **Description** | **Constraint** |
| Polling\_no | Integer | 20 | Polling station no | Primary key |
| Polling\_name | varchar | 30 | Polling station name | Null |
| Ac\_no | Integer | 10 | Display Ac\_no | Null |
| Ac\_name | Varchar | 20 | Display Ac\_name | Null |
| Center\_addr | Varchar | 100 | Address of that center | Null |

6**. Table Name: Citizen Registration**

Description: The table contains all information about the citizen registration.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Data type** | **Size** | **Description** | **Constraint** |
| reg\_id | Varchar | 30 | Applcnt User\_id | Primary key |
| F\_name | Varchar | 30 | Citizen First Name | Null |
| M\_name | Varchar | 30 | Citizen Middle Name | Null |
| L\_name | Varchar | 30 | Citizen Last Name | Null |
| Addr | Varchar | 100 | Citizen Address | Null |
| City | Varchar | 30 | Citizen City | Null |
| State | Varchar | 30 | Citizen State | Null |
| Pincode | Integer | 20 | Citizen Pincode | Null |
| Username | Varchar | 30 |  | Null |
| Password | Varchar | 30 | Applicant password | Null |
| Mob\_no | Integer | 20 | Mobile number | Null |
| Email\_id | Varchar | 30 | Email id | Null |

7**. Table Name: vcard**

Description : The table contains the information about voting card details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Data type** | **Size** | **Description** | **Constraint** |
| V\_no | Integer | 10 | Appcnt voter\_no | Primary key |
| V\_name | Varchar | 50 | Applcnt name | Null |
| V\_middle name | Varchar | 30 | Father/husband name of applcnt. | Null |
| Gender | Varchar | 20 | Gender of applcnt | Null |
| DOB | Date | 30 | Apllcnt DOB | Null |
| App\_address | Varchar | 50 | Applcnt adrr | Null |

8. **Table Name : User**

Description : The table contains the information about Users.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Form** | **Data type** | **Size** | **Description** | **Constraint** |
| Username | Varchar | 20 | User Name | Null |
| Password | Varchar | 20 | Password | Null |

**Test Procedures and Implementation**

Software Testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs. It can also be stated as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development, so that it works as expected and can be implemented with the same characteristics.

Software Testing, depending on the testing method employed, can be implemented at any time in the development process, however the most test effort is employed after the requirements have been defined and coding process has been completed.

**Test procedure:**

The project needs to be fully tested as it needs to be integrated to an internal existing system. Also the data that it possesses and its accuracy are equally important. Since the system works on the on the network it was thoroughly tested internally and can only then be implemented to the other modules.

**The methods adopted for testing are as follows:**

**Unit Testing:**

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect. Each unit is tested separately before integrating them into modules to test the interfaces between modules. Unit testing has proven its value in that a large percentage of defects are identified during its use.

Unit testing is a software verification and validation method where the programmer gains confidence that individual units of source code are fit for use. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual program, function, procedure, etc., while in object-oriented programming, the smallest unit is a class, which may belong to a base/super class, abstract class or derived/child class.

Ideally, each test case is independent from the others: substitutes like method stubs, mock objects, fakes and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended. Its implementation can vary from being very manual (pencil and paper) to being formalized as part of build automation.

The entire code that is used in the project has been tested thou roughly. The code is tested by subjecting it with various inputs and in various amounts, with various databases. The output is then tested against projects requirements, as stated above they were first locally.

**Integration Testing:**

Integration testing, also known as integration and testing (I&T), is a software development process which program units are combined and tested as groups in multiple ways. In this context, a unit is defined as the smallest testable part of an application. Integration testing can expose problems with the interfaces among program components before trouble occurs in real-world program execution. Integration testing is a component of Extreme Programming (XP), a pragmatic method of software development that takes a meticulous approach to building a product by means of continual testing and revision.

There are two major ways of carrying out an integration test, called the bottom-up method and the top-down method. Bottom-up integration testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds. In top-down integration testing, the highest-level modules are tested first and progressively lower-level modules are tested after that. In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing.

It is the systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that been dictated by design.

The system involves data transfer through TCP/IP this made it necessary to test whether the input to the output of one module properly enters as input of the other module. For this purpose integration testing has to be done.

**System Testing:**

In the system testing the primary objective was to fully exercise the computer based system.

**Recovery Testing:**

In recovery testing the system was forced to fall in variety of ways and verified that recovery is properly performed. If the recovery was atomic, re-initialization, data recovery and restart were evaluated for correctness

**Security Testing:**

As the system has to be operated on LAN, security testing determined to verify that protection mechanisms built in to the system protect it from improper penetration.

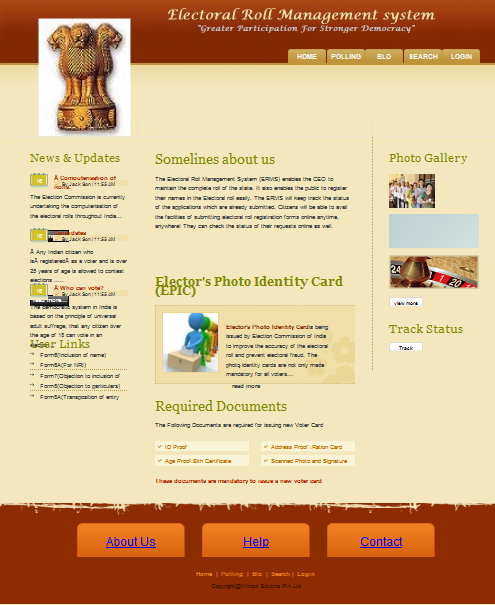
**Implementation:**

It is considered with those tasks leading immediately to a fully operational system. It involves programmers, users, operation management, but its planning and timing is a prime function of the system analysis. It includes the final testing of the complete system to user satisfaction and supervision of the initial operation of the new system.

**User Manual:**

**1 User Manual for Home Page**

Main Menu

****

Contact

us

Photo

galary

News

&

Updates

**2 User Manual for Home Page**

****

Forget Password

New Registration

User Login

The ‘Electoral Roll Management System’ site is basically built for software project management. The basic purpose behind to build this site is to effective and efficient project management with early project delivery. ERO (Admin), BLO, Citizen these are main users of this site. ERO is key user but his role is hide be hide the project he just manage and maintains all project which registered in system.

How to use the system?

There are three types of users:

1. ERO(Admin)
2. BLO
3. Citizen

The system start by typing the URL in the browser , which shows the home page that contains many links. There are following pages for Administrator.

1. Home Page:

This is the Welcome page that display the banner of website. The home contains the main menu bar of various menus like Home, About us, Login BLO Polling ,Search etc.

1. About Us:

Here user gets the all website related information and including project members and office address and contact.

1. Login:

By this menu ERO, BLO, Citizen can logged in system.

**Operational Manual**

1. **Registration:**

Step 1: Click on registration menu.

Step 2: Register as citizen.

Step 3: Fill all registration details click on save button.

After clicking save button successful message is will be generated.

After clicking on login button user is logged in to website.

1. **User Login:**

Step 1: Click on login menu.

Step 2: Select Form

Fill Form

Upload Documents

Step 3: Click on logout button.

1. **ERO Login:**
2. Step 1: Click on login menu.

Step 2: Add BLO, Polling.

Update BLO, Polling.

Remove BLO, Polling.

Step 3: Click on logout button.

1. **BLO Login:**

Step 1: Click on login menu.

Step 2: Receive Documents.

Verify Documents.

Send Status.

Step 3: Click on logout button.

**Drawbacks and Limitations**

Every System cannot be referred as complete or a perfect system in all aspects in spite of successful execution of testing plan. The problem arises, when the system enters in the real working environment, the problem which arises cannot be predetermined to be rectified.

The limitations are:

* Citizen can have only one userid & password, It is problematic if miss it.
* System is not fully developed to provide all the proposed functionality, some part is still under development

**PROPOSED ENHANCEMENTS**

**Proposed Enhancement:**

* Online Processing.
* Fast processing and less efforts.
* Security.

**Objectives and scope:**

* Maintain the elector’s voting\_id card information at district level.
* All the status of citizens and BLO can be updated and checked via SMS and E-mails.
* Final verification of the Applicant of voting card can be done by using web cam & digital signature by ERO.
* Home delivery of voting card.

**CONCLUSION**

* Automates all the activities taking place manually.
* This reduces the chances of errors.
* Lower the cost incurred in manual process.
* The system suits need of the users
* System is generalized to include may function and as per user requirements
* The system can be customized for specific needs
* The system is flexible enough to be integrated with others and there is scope for expanding system as well

**Bibliography:**

References for the Project Development were taken from the following Books and Web Sites.

Starting with Struts 2

- (Ian Roughley), Free PDF mini-book

* Web Work in Action book

-(Patrick Lightbody, Jason Carreira), print only. (Most of WW2 in Action applies to Struts 2. The Struts 2 package names changed, and we've added some features, but the fundamentals remain the same.)

* Struts 2 in Action

-(Don Brown, Chad Michael Davis, Scott Stanlick)

* Struts 2 Design and Programming

-(Bud Kurniawan).

**Books:**

1. Java Server Programming Black Book
2. Struts 2 Black Book
3. J2EE Professional by Shadab siddiqui
4. JAVA server pages by Larne Pekowsley
5. JAVA Server pages by Nick Todd

**Web Sites:**

[www.Wikipedia.com](http://www.Wikipedia.com)

[www.javatpoints.com](http://www.javatpoints.com)

[www.google.com](http://www.google.com)

[www.stackoverflow.com](http://www.stackoverflow.com)

[www.codeproject.com](http://www.codeproject.com)