

LAB REPORT

Submitted by

JAYKUMAR PATEL

[RA2111003011242]

Under the Guidance of

Mr. Vinoth N.A.S

Assistant Professor, C.Tech

In partial satisfaction of the requirements for the degree of

**BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING**



SCHOOL OF COMPUTING

**COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

KATTANKULATHUR - 603203

MAY 2023



COLLEGE OF ENGINEERING & TECHNOLOGY
SRM INSTITUTE OF SCIENCE & TECHNOLOGY
S.R.M. NAGAR, KATTANKULATHUE – 603 203
Chengalpattu District

BONAFIDE CERTIFICATE

Register No. RA2111003011218 Certified to be the bonafide
work done by JAYKUMAR PATEL of II Year/IV
Sem B.Tech Degree Course in the Practical Software Software Engineering and
Project Management 18CSC206J in SRM INSTITUTE OF SCIENCE AND
TECHNOLOGY, Kattankulathur during the academic year 2022 – 2023.

A handwritten signature in black ink, appearing to read 'Vinoth' or 'Vinoth N.A.S.' followed by the date '21/5/2023'.

LAB INCHARGE

Mr. Vinoth N.A.S

Assistant professor

Department of Computing Technologies

SRMIST – KTR.

A handwritten signature in blue ink, appearing to read 'M. Pushpalatha'.

Head of the Department

Date: 4/05/2023

ABSTRACT

In the present climate, emotional well-being is a main pressing issue. The emotional well-being circumstance has weakened because of a major number of individuals telecommuting and being isolated from their friends and family. Thus, it's basic to monitor and resolve any issues before they get excessively intense. Thus, this venture attempts to accomplish these things for client. This application carries yourself to a condition of quiet and wellbeing. Client will figure out how to oversee everyday pressure and work on their emotional well-being. Remembering that a few clients might be experiencing psychological instability and may not wish to communicate with an application as much as others. Thus, the application would be exceptionally easy to understand and inviting. To create the application, I will utilize Python and Machine learning. This multitude of dialects will be utilized to foster the application which will be easy to use and have basic UI. The application will initially let the client to signed in and after that it will show a few arrangements of questionaries which client need to reply. As indicated by answers given by the client, the condition of mental prosperity is shown and based on which legitimate idea is given. What's more, in the extreme case, much appropriate medication and specialist is proposed to the client which will assist them with defeating their current state.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO 1
	ABSTRACT	
	LIST OF FIGURES	
	LIST OF ABBREVIATIONS	
1	PROBLEM STATEMENT	1
2	STAKEHOLDERS & PROCESS MODELS	5
3	IDENTIFYING REQUIREMENTS	7
4	PROJECT PLAN & EFFORT	10
5	WORK BREAKDOWN STRUCTURE & RISK ANALYSIS	16
6	SYSTEM ARCHITECTURE, USE CASE & CLASS DIAGRAM	22
7	ENTITY RELATIONSHIP DIAGRAM	26
8	DATA FLOW DIAGRAM	32
9	SEQUENCE & COLLABORATION DIAGRAM	36
10	DEVELOPMENT OF TESTING FRAMEWORK/USER INTERFACE	39
11	TEST CASES & REPORTING	44
12	ARCHITECTURE/DESIGN/FRAMEWORK/IMPLEMENTATION	54
	CONCLUSION	
	REFERENCES	
	APPENDIX (CODE)	

LIST OF FIGURES

1. The Waterfall method
2. Project Management plan
3. Effort and cost estimation
4. Infrastructure/Resource cost
5. Maintenance and support cost
6. Identification team members
7. WBS for a mental health app
8. SWOT analysis
9. Risk management framework
10. Timeline- Gantt chart
11. System Architecture
12. Use case diagram
13. Class diagram
14. ER diagram
15. DFD level 0
16. DFD level 1
17. Sequence diagram
18. Collaboration diagram
19. Types of testing
20. Functional test cases
21. Non-functional test cases
22. Functional requirement
23. Current status of testing
24. Present obstacles
25. Recommendation to remove obstacles
26. Distribution and density by age
27. Total distribution by treated or not
28. Probability of mental health condition
29. Confusion matrix
30. Histogram of predicted probabilities
31. ROC curve
32. Success of methods

LIST OF ABBREVIATIONS

- 1. WBS – Work Breakdown Structure**
- 2. SWOT – Strengths , Weaknesses , Opportunities , Threats**
- 3. ER Diagram – Entity Relationship Diagram**
- 4. DFD – Data Flow Diagram**
- 5. UML - Unified Modelling Language**
- 6. ROC Curve – Receiver Operating Characteristic Curve**



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	1
Title of Experiment	To identify the Software Project, Create Business Case, Arrive at a Problem Statement
Name of the candidate	SHRAVANI S. MANE
Team Members	1.JAYKUMAR PATEL (RA2111003011218) 2.DEEPTHI AKKIPALI (RA2111003011247)
Register Number	RA2111003011223
Date of Experiment	

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To Frame a project team, analyze and identify a Software project. To create a business case and Arrive at a Problem Statement for the <title of the project>

Team Members:

S. No	Register No	Name	Role
1	RA2111003011223	SHRAVANI S. MANE	Lead/Rep
2	RA2111003011247	DEEPTHI AKKIPALLI	Member
3	RA2111003011218	JAYKUMAR PATEL	Member

Project Title: ANDROID VOTING SYSTEM.

Problem Statement: Because of this pandemic to avoid, crowd and maintaining social distancing E-Voting system is much easier to vote. It offers the facility of online voting and saves individuals time while standing in the queue.

ONE PAGE BUSINESS CASE TEMPLATE

DATE	
SUBMITTED BY	SHRAVANI S MANE (RA2111003011223) JAYKUMAR PATEL (RA2111003011218) DEEPTHI AKKIPALLI (RA2111003011247)
TITLE / ROLE	

1. The android voting system is an android application that provides a new technique of casting votes using mobile phones. This application is especially developed for organizations, corporations and commercial businesses to get employees' opinions whenever there is any new policy implemented or any issue being investigated or during controversies.
2. The project provides an effective solution in resolving all the arguments that occur in organizations by considering all employees' opinions.
3. The system can be used anytime and from anywhere by the employees.
4. It excludes the use of manual voting process.

5.no one can cast the votes on behalf of others and multiple times.

6.saves time and reduces human intervention.

7.It makeS employees happy as their opinions are considered for the matter in the organization.

8. Admin can get instant results.

9. The system is flexible and secured to be used.

HISTORY

. The history of electronic voting and the use of mobile devices in elections dates back to the mid-20th century. In the 1950s and 1960s, the first electronic voting machines were developed, and by the 1980s, electronic voting systems were in widespread use in many countries.

In the early 2000s, the widespread adoption of smartphones and mobile devices created new opportunities for the use of technology in elections. In recent years, many countries have explored the use of mobile voting as a way to increase voter participation and improve the efficiency of the voting process.

While the use of mobile voting is still in its early stages, it has the potential to revolutionize the way elections are conducted by providing a convenient, secure, and accessible way for citizens to participate in elections. However, the development of mobile voting systems has also been met with concerns about security, reliability, and the potential for fraud and tampering.

LIMITATIONS

1. Technical limitations: Mobile devices are prone to technical issues such as connectivity problems, software crashes, and security vulnerabilities, which could impact the integrity and security of the voting process.
2. Accessibility: Not all citizens may have access to a mobile device or the internet, which could limit the accessibility of the system and impact the fairness of the election.
3. Security risks: Online voting systems are vulnerable to security risks such as hacking, tampering, and fraud, which could compromise the confidentiality and integrity of the voting process.
4. Public trust and confidence: There may be concerns among the public about the security and reliability of online voting systems, which could impact the trust and confidence in the election process.
5. Integration with existing systems: The Android voting system may need to be integrated with existing voting systems, such as electronic voting machines and vote counting systems, which could pose technical and logistical challenges.
6. Implementation costs: The development and deployment of the Android voting system can be expensive and may require significant investment in technology, infrastructure, and personnel.

7. Maintenance and support: The system will require ongoing maintenance and support to ensure it continues to function effectively and efficiently, which could increase the costs associated with the system over time.

APPROACH

1. Requirements gathering and analysis: This phase involves gathering requirements from stakeholders such as election organizations and voters, and analyzing the requirements to determine the scope and goals of the project.
2. System Design: In this phase, the system architecture and design are created, including the user interface, database, and communication protocols. The design should also consider security measures to ensure the confidentiality and integrity of the voting process.
3. Implementation: In this phase, the code is written and the system is built, tested, and deployed. This phase may involve multiple iterations of testing and refinement to ensure the system meets the requirements and functions as expected.
4. Deployment: The Android voting system is deployed and made available to voters. This phase also involves training election organizations on how to use the system and providing support as needed.
5. Maintenance: The system is monitored and maintained to ensure it continues to function effectively and efficiently. This may involve fixing bugs, updating software, and making improvements as needed.

BENEFITS

- Increased voter turnout: The convenience and accessibility of the Android voting system is expected to increase voter participation and encourage more citizens to participate in elections.
- Improved security: The use of encrypted communication protocols and robust security measures will reduce the risk of fraud and tampering, ensuring the integrity of the voting process.
- Reduced costs: By eliminating the need for paper ballots and reducing the costs associated with traditional voting methods, the Android voting system will provide a more cost-effective solution for election organizations.
- Greater transparency: The real-time results and tracking capabilities of the system will provide greater transparency and accountability in the election process.

Result

Thus, the project team formed, the project is described, the business case was prepared and the problem statement was arrived.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	2
Title of Experiment	Identification of Process Methodology and Stakeholder Description
Name of the candidate	JAYKUMAR PATEL(RA2111003011218)
Team Members	SHRAVANI MANE(RA2111003011223) DEEPTHI AKKIPALLI(RA2111003011247)
Register Number	RA2111003011218
Date of Experiment	2/02/2023

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

effectively and efficiently over time. This may include regular software updates, bug fixes, and technical support.

Functional Requirements:-

1. Voter registration: The system should allow voters to register to vote and verify their eligibility to participate in the election.
2. Ballot presentation: The system should present the ballot to the voter in a clear and easy-to-understand format, including the candidates and any propositions or issues being voted on.
3. Vote casting: The system should allow voters to cast their votes and ensure that the votes are recorded accurately and securely.
4. Vote counting: The system should accurately count the votes and provide real-time results during the election.
5. Reporting and analysis: The system should generate reports and provide analysis of the election results, including vote counts by candidate, geographic region, and other factors.
6. Security: The system should have robust security measures in place to ensure the confidentiality and integrity of the voting process, including measures such as encryption, authentication, and secure communication protocols.

Non-Functional Requirements:-

1. Usability: The system should have a user-friendly interface and be easy to use for voters of all ages and abilities.
2. Performance: The system should be able to handle a large volume of traffic during peak periods and provide real-time results during the election.
3. Scalability: The system should be able to scale up or down as needed to accommodate changing voter populations and election requirements.
4. Reliability: The system should have high availability and uptime to ensure that it is always available when voters need it.
5. Security: The system should be secure and protect against security risks such as hacking, tampering, and fraud.
6. Privacy: The system should protect the privacy of voters and their personal information.

Result:-

Thus the requirements were identified and accordingly described.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	3
Title of Experiment	System, Functional and Non-Functional Requirements of the Project
Name of the candidate	JAYKUMAR PATEL (RA2111003011218)
Team Members	1.SHRAVANI S. MANE (RA2111003011223) 2.DEEPTHI AKKIPALI (RA2111003011247)
Register Number	RA2111003011218
Date of Experiment	31/01/2023

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To identify the system, functional and non-functional requirements for the project.

Team Members:

S No	Register No	Name	Role
1	RA2111003011223	SHRAVANI S. MANE	Rep/Member
2	RA2111003011247	DEEPTHI AKKIPALLI	Member
3	RA2111003011218	JAYKUMAR PATEL	Member

Project Title: ANDROID VOTING SYSTEM.

System Requirements:-

1. Technical requirements: The system should be able to run on a range of Android devices and be compatible with the latest versions of the Android operating system. It should also have a user-friendly interface and be easy to use.
 2. Security requirements: The system should have robust security measures in place to ensure the confidentiality and integrity of the voting process, including measures such as encryption, authentication, and secure communication protocols.
 3. Database requirements: The system should have a secure and scalable database to store voter information, vote counts, and other data related to the election.
 4. Network requirements: The system should be able to communicate securely over the internet and be able to handle a large volume of traffic during peak periods.
 5. Accessibility requirements: The system should be accessible to a wide range of voters, including those with disabilities. It should also be available in multiple languages to ensure accessibility for a diverse population.
 6. Integration requirements: The system should be able to integrate with existing voting systems, such as electronic voting machines and vote counting systems, to ensure a seamless and efficient election process.
- Maintenance and support requirements: The system should have a clear plan for maintenance and support to ensure it continues to function

Staff Signature with date

Aim

To identify the appropriate Process Model for the project and prepare Stakeholder and User Description.

Team Members:

Sl No	Register No	Name	Role
1	RA2111003011218	JAYKUMAR PATEL	Member
2	RA2111003011223	SHRAVANI MANE	Member
3	RA2111003011247	DEEPTHI AKKIPALLI	Member

Project Title: ANDROID VOTING SYSTEM.

Selection of Methodology :

Two of the most commonly used methodologies for software development projects are Agile and Waterfall.

Agile methodology: Agile is a flexible and iterative approach that emphasizes collaboration, adaptability, and continuous delivery. In an Agile approach, the project is divided into smaller, manageable pieces and is delivered in short iterations, with regular feedback and continuous improvement.

Waterfall methodology: Waterfall is a sequential and linear approach that emphasizes planning, design, and testing before deployment. In a Waterfall approach, each phase of the project is completed before moving on to the next phase, and there is less opportunity for changes and adaptations once the project is underway.

Result

Thus the Project Methodology was identified and the stakeholders were described.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	4
Title of Experiment	Prepare Project Plan based on scope, Calculate Project effort based on resources and Job roles and responsibilities
Name of the candidate	JAYKUMAR PATEL
Team Members	1.SHRAVANI S. MANE 2.Deepthi akkipalli
Register Number	RA2111003011218
Date of Experiment	18/2/23

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To Prepare Project Plan based on scope, Calculate Project effort based on resources, Find Job roles and responsibilities

Team Members:

Sl No	Register No	Name	Role
1	RA2111003011223	SHRAVANI MANE	Lead
2	RA2111003011247	DEEPTHI AKKIPALLI	Member
3	RA2111003011218	JAYKUMAR PATEL	Member

Requirements

<Incorporate the Project plan template>

Result:

Thus, the Project Plan was documented successfully.

1. Project Management Plan

Describe the key issues driving the project. [Min 3 Focus Areas]

Focus Area	Details
Project Description	Develop an Android Voting System that allows users to vote remotely from their mobile devices. The system will provide a secure and easy-to-use interface for voters and administrators.
Scope Management	<ol style="list-style-type: none">1. Develop a user-friendly mobile app for voters2. Develop an administrative portal for administrators3. Implement secure login and authentication system4. Allow for the creation of elections and voting events5. Implement an efficient and secure vote counting system6. Provide real-time result updates7. Ensure system security and data privacy8. Test the system thoroughly
Project Timeline	<ol style="list-style-type: none">1. Planning and Design Phase: 2 weeks2. Development Phase: 12 weeks3. Testing Phase: 4 weeks4. Deployment and Maintenance Phase: 2 weeks
Resources Required:	<ol style="list-style-type: none">1. Project Manager2. Android App Developer3. Backend Developer4. UX/UI Designer5. QA Tester6. Technical Writer
Job Roles and Responsibilities:	<ol style="list-style-type: none">1. Project Manager: Responsible for project planning, coordination, and communication with stakeholders. Will also manage the project budget and schedule.2. Android App Developer: Responsible for developing the Android voting app and ensuring it meets the required specifications.3. Backend Developer: Responsible for developing the backend system that will handle user authentication, election and voting event creation, vote counting, and result updates.4. UX/UI Designer: Responsible for designing the user interface of the Android app and administrative portal.5. QA Tester: Responsible for testing the system thoroughly to ensure it meets the required quality standards.

	6.Techical Writer: Responsible for creating user manuals, technical documentation, and other relevant materials.
Resource Management	Estimate and Manage the need People: People & Skills Required Finance: Budget Required Physical: Facilities, IT Infrastructure
Project Effort:	1.Project Manager: 150 hours 2.Android App Developer: 600 hours 3.Backend Developer: 400 hours 4.UX/UI Designer: 200 hours 5.QA Tester: 200 hours 6.Techical Writer: 100 hours

2. Estimation

2.1. Effort and Cost Estimation

Activity Description	Sub-Task	Sub-Task Description	Effort (in hours)	Cost in INR
Planning and Design	Android App Developer	Responsible for developing the Android voting app and ensuring it meets the required specifications.	600 hours	200000
	UX/UI Designer	Responsible for designing the user interface of the Android app and administrative portal.	200 hours	150000

Effort (hr)	Cost (INR)
1	500

2.2. Infrastructure/Resource Cost [CapEx]

< OneTime Infra requirements >

Infrastructure Requirement	Qty	Cost per qty	Cost per item

App developer		350000	1000000
website maintenance		50000	3000000

2.3 Maintenance and Support Cost [OpEx]

Category	Details	Qty	Cost per qty per annum	Cost per item
People	Network, System, Middleware and DB admin	3	2,000,000	6,000,000
	Developer , Support Consultant			
License	Operating System Database Middleware IDE	10	10000	100,000
Infrastructures	Server, Storage and Network	20	20000	400,000

3. Project Team Formation

3.1. Identification Team members

Name	Role	Responsibilities
Shravani Mane	Key Business User (Product Owner)	Provide clear business and user requirements
Shravani Mane	Project Manager	Manage the project
Jay Patel	Business Analyst	Discuss and Document Requirements
Jay Patel	Technical Lead	Design the end-to-end architecture
Deepthi Akkipalli	UX Designer	Design the user experience
Shravani Mane	Frontend Developer	Develop user interface
Jay Patel	Backend Developer	Design, Develop and Unit Test Services/API/DB
Deepthi Akkipalli	Cloud Architect	Design the cost effective, highly available and scalable architecture
Shravani Mane	Cloud Operations	Provision required Services
Jay Patel	Tester	Define Test Cases and Perform Testing

3.2. Responsibility Assignment Matrix

RACI Matrix	Team Members			
Activity	Name (BA)	Name (Developer)	Name (Project Manager)	Key Business User
User Requirement Documentation	Shravani Mane(A)	jay Patel(R) Shravani Mane(R) Deepthi Akkipalli(R)	Shravani (I)	Shravani Mane(R) jay Patel(R) Deepthi Akkipalli(R)

A	Accountable
R	Responsible
C	Consult
I	Inform

Result:

Thus, the Project Plan was documented successfully.

Reference

1. <https://www.pmi.org/>
2. <https://www.projectmanagement.com/>
3. <https://www.tpsgc-pwgsc.gc.ca/biens-property/snnp-npms/ti-it/ervcpgrm-dsfvpmppt-eng.html>



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	5
Title of Experiment	Prepare Work breakdown structure, Timeline chart, Risk identification table
Name of the candidate	JAY KUMAR PATEL
Team Members	SHRAVANI S. MANE DEEPTHI AKKIPALI
Register Number	RA2111003011218
Date of Experiment	18/2/23

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

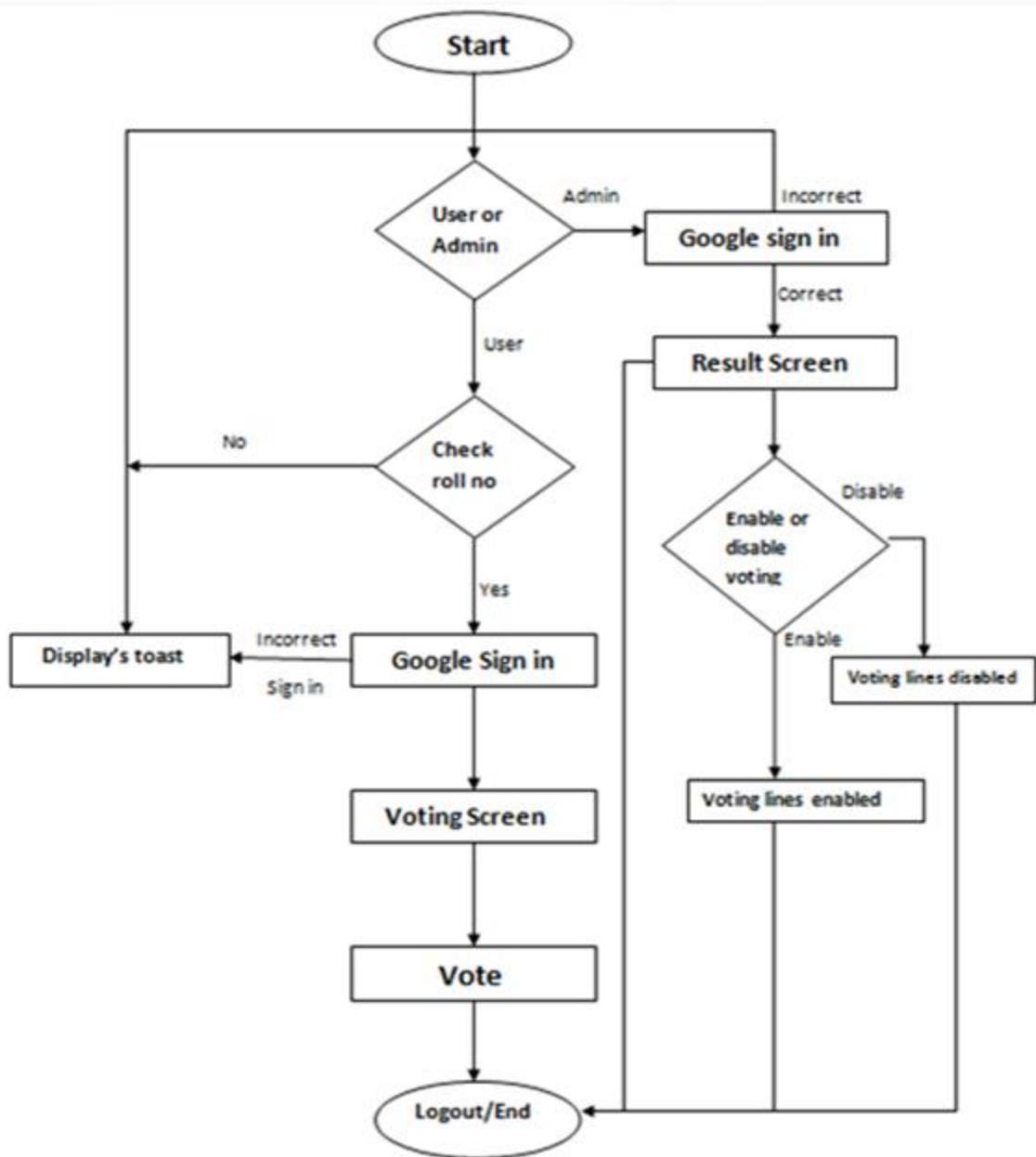
Aim

To Prepare Work breakdown structure, Timeline chart and Risk identification table

Team Members:

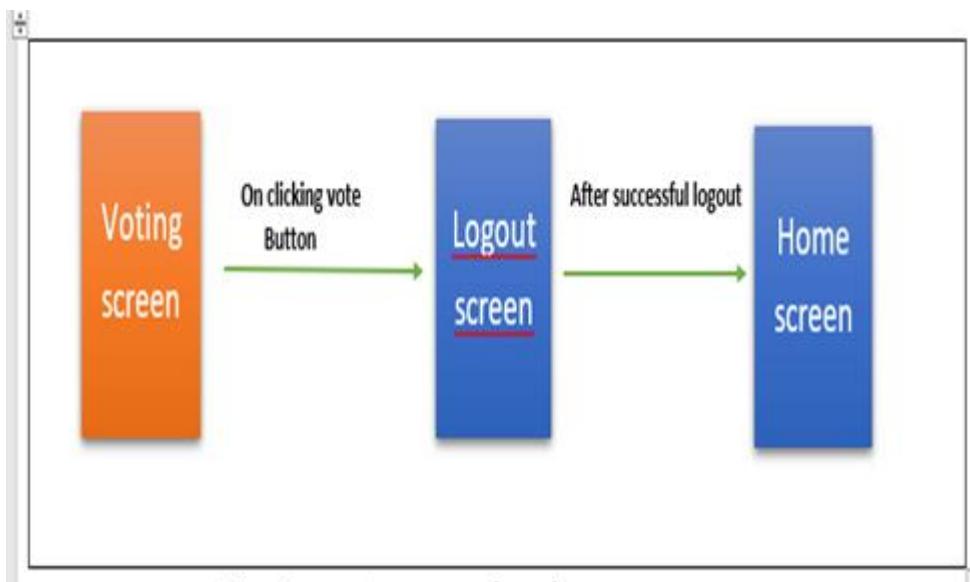
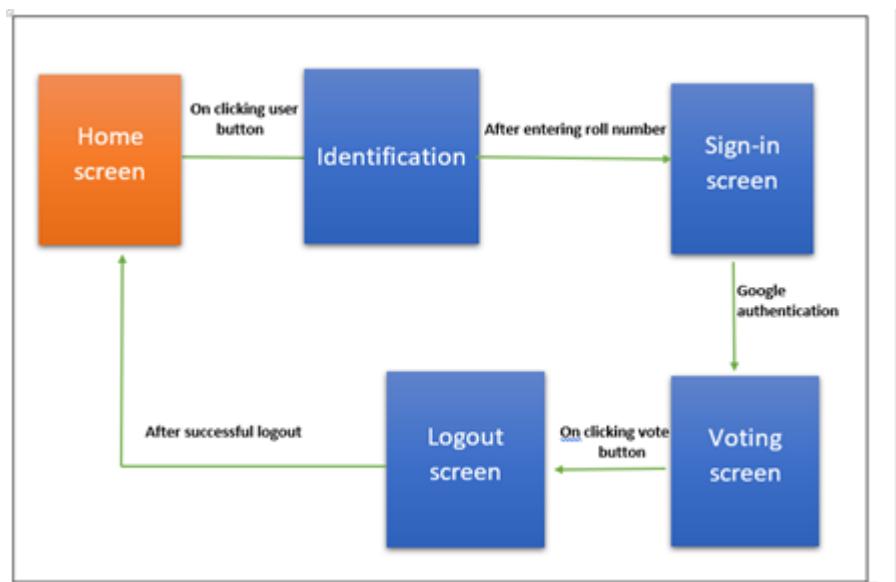
Sl No	Register No	Name	Role
1	RA2111003011223	SHRAVANI S. MANE	Rep
2	RA2111003011218	JAYKUMAR PATEL	Member
3	RA2111003011247	DEEPTHI AKKIPALLI	Member

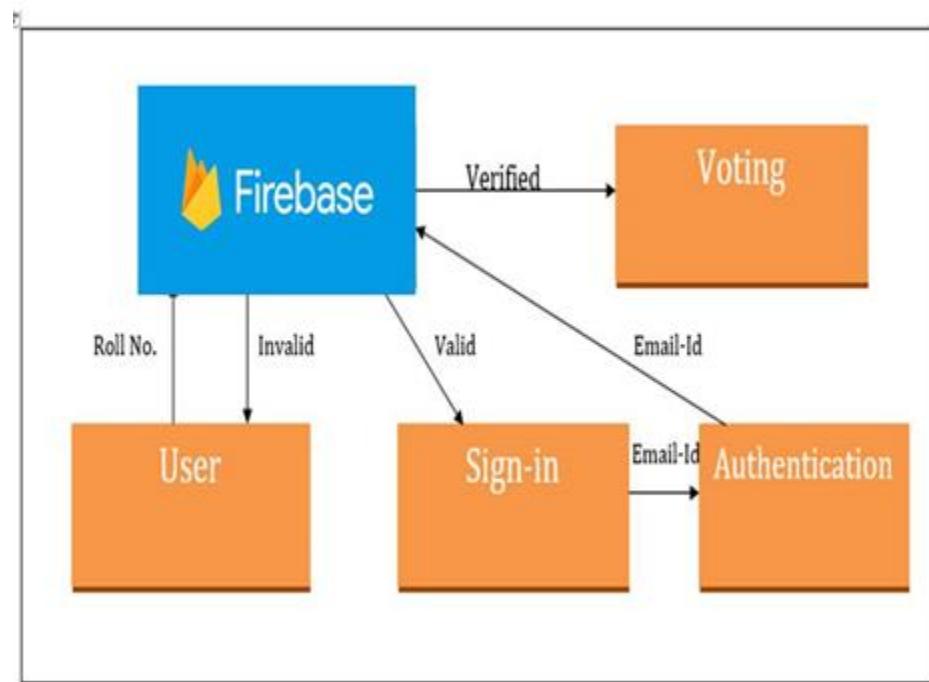
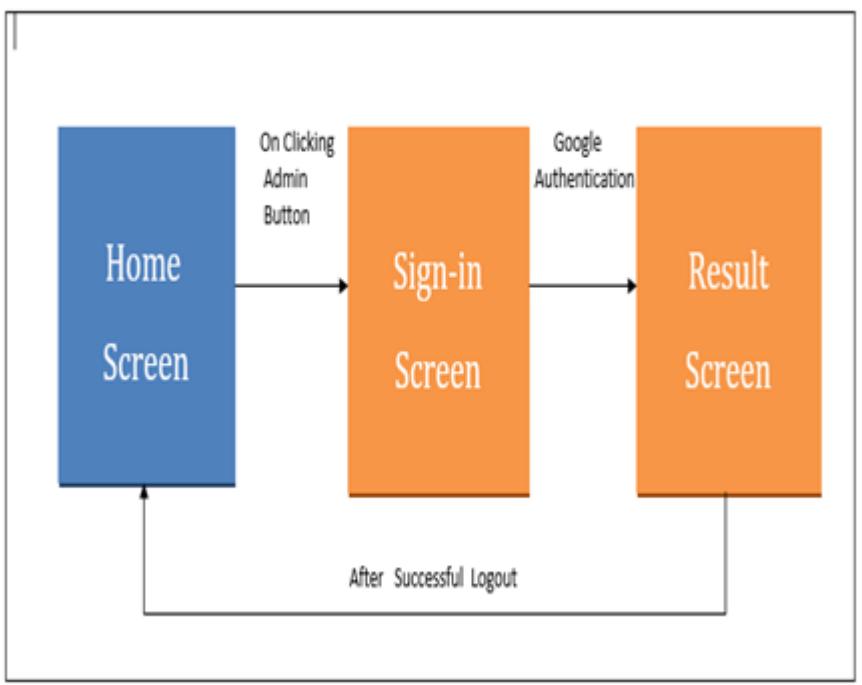
WBS – Examples



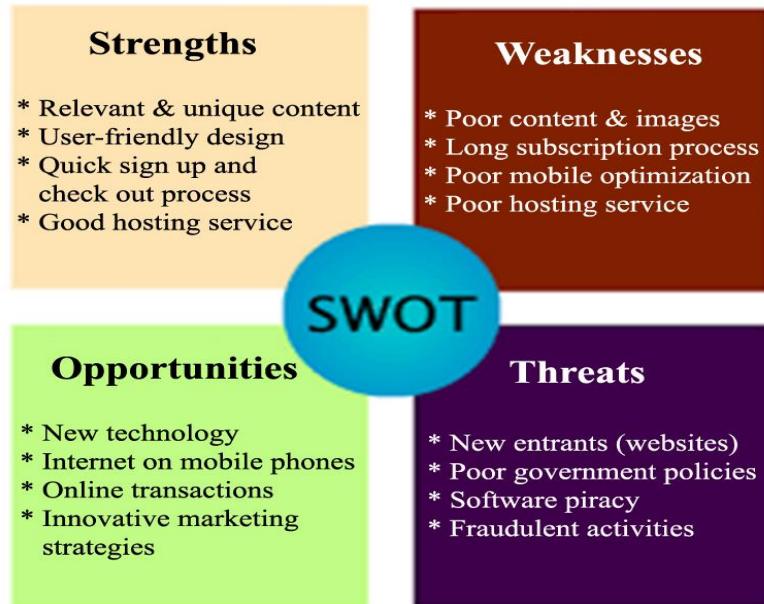
- 0.0 Retail Web Site
- 1.0 Project Management
- 2.0 Requirements Gathering
- 3.0 Analysis & Design
- 4.0 Site Software Development
 - 4.1 HTML Design and Creation
 - 4.2 Backend Software
 - 4.2.1 Database Implementation
 - 4.2.2 Middleware Development
 - 4.2.3 Security Subsystems
 - 4.2.4 Catalog Engine
 - 4.2.5 Transaction Processing
 - 4.3 Graphics and Interface
 - 4.4 Content Creation

5.0 Testing and Production





RISK ANALYSIS – SWOT & RMMM



 Risk Management Framework- Risks And Mitigation ...

Response	Strategy	Examples
Avoid	Risk avoidance is a strategy where the project team takes action to remove the threat of the risk or protect from the impact	<ul style="list-style-type: none"> ▪ Extending the schedule ▪ Reducing/removing scope ▪ Change the execution strategy
Transfer	Risk transference involves shifting or transferring the risk threat and impact to a third party. Rather transfer the responsibility and ownership	<ul style="list-style-type: none"> ▪ Purchasing insurance ▪ Performance bonds ▪ Warranties ▪ Contract issuance (lump sum)
Mitigate	Risk mitigation is a strategy where the project team takes action to reduce the probability of the risk occurring. This does not risk or potential impact , but rather reduces the likelihood of it becoming real.	<ul style="list-style-type: none"> ▪ Increasing testing ▪ Changing suppliers to a more stable one ▪ Reducing process complexity
Accept	Risk acceptance means the team acknowledges the risk and its potential impact, but decides not to take any preemptive action to prevent it. It is dealt with only if it occurs.	<ul style="list-style-type: none"> ▪ Contingency reserve budgets ▪ Management schedule float ▪ Event contingency

Slide 1 of 5

Result:

Thus, the work breakdown structure with timeline chart and risk table were formulated successfully.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	6
Title of Experiment	Design a System Architecture, Use Case and Class Diagram
Name of the candidate	JAYKUMAR PATEL (RA2111003011218)
Team Members	1. SHRAVANI S. MANE (RA2111003011223) 2. DEEPTHI AKKIPALI (RA2111003011247)
Register Number	RA2111003011218
Date of Experiment	22/02/2023

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To Design a System Architecture, Use case and Class Diagram

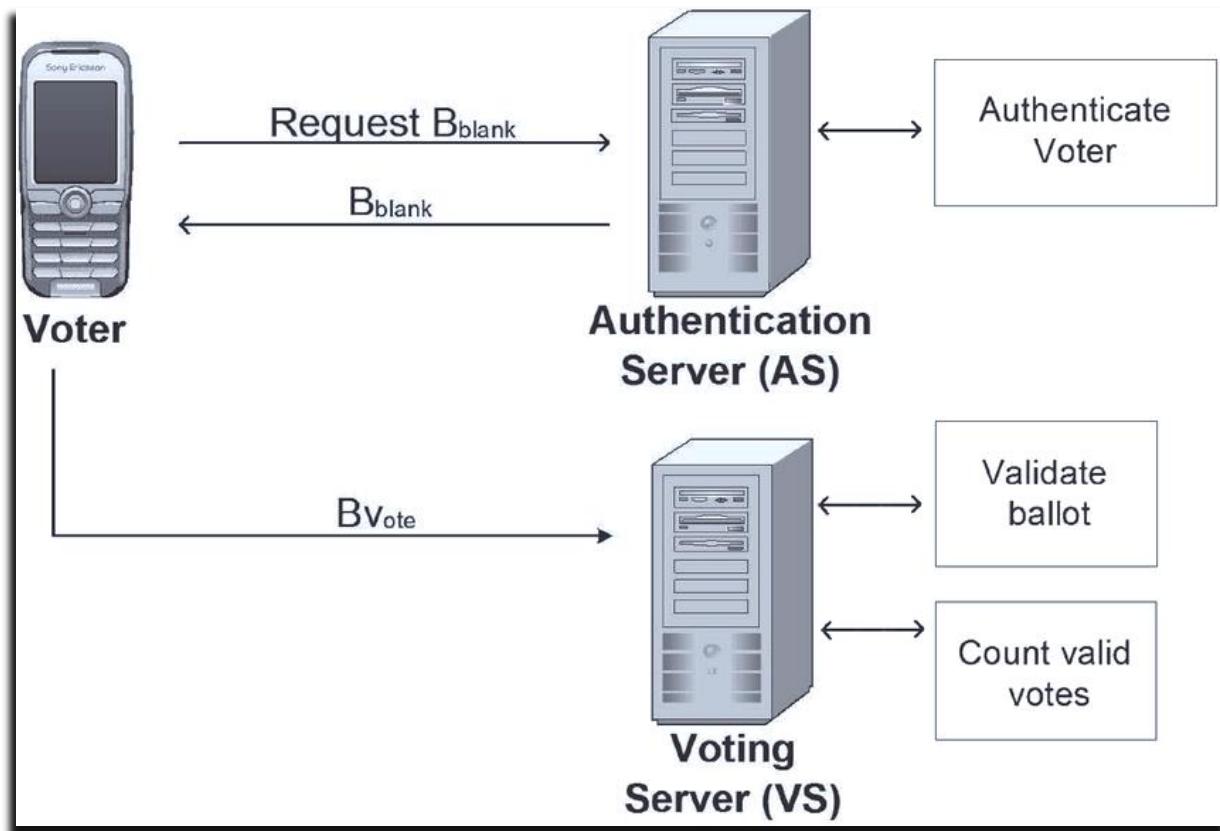
Team Members:

Sl No	Register No	Name	Role
1	RA2111003011223	SHRAVANI S. MANE	Rep
2	RA2111003011247	DEEPTHI AKKIPALLI	Member
3	RA2111003011218	JAYKUMAR PATEL	Member

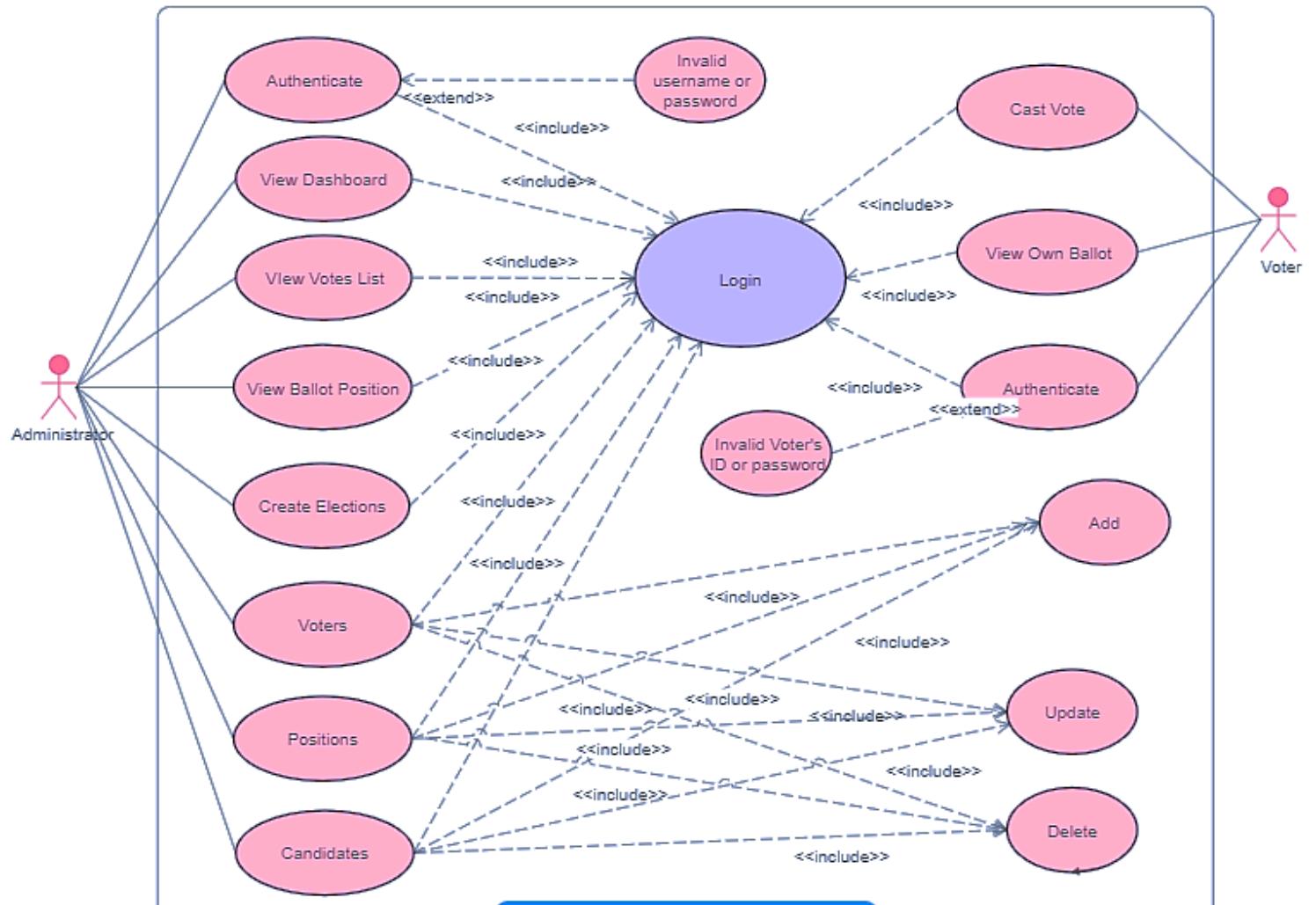
Requirements

<System Architecture, Use Case and Class Diagram>

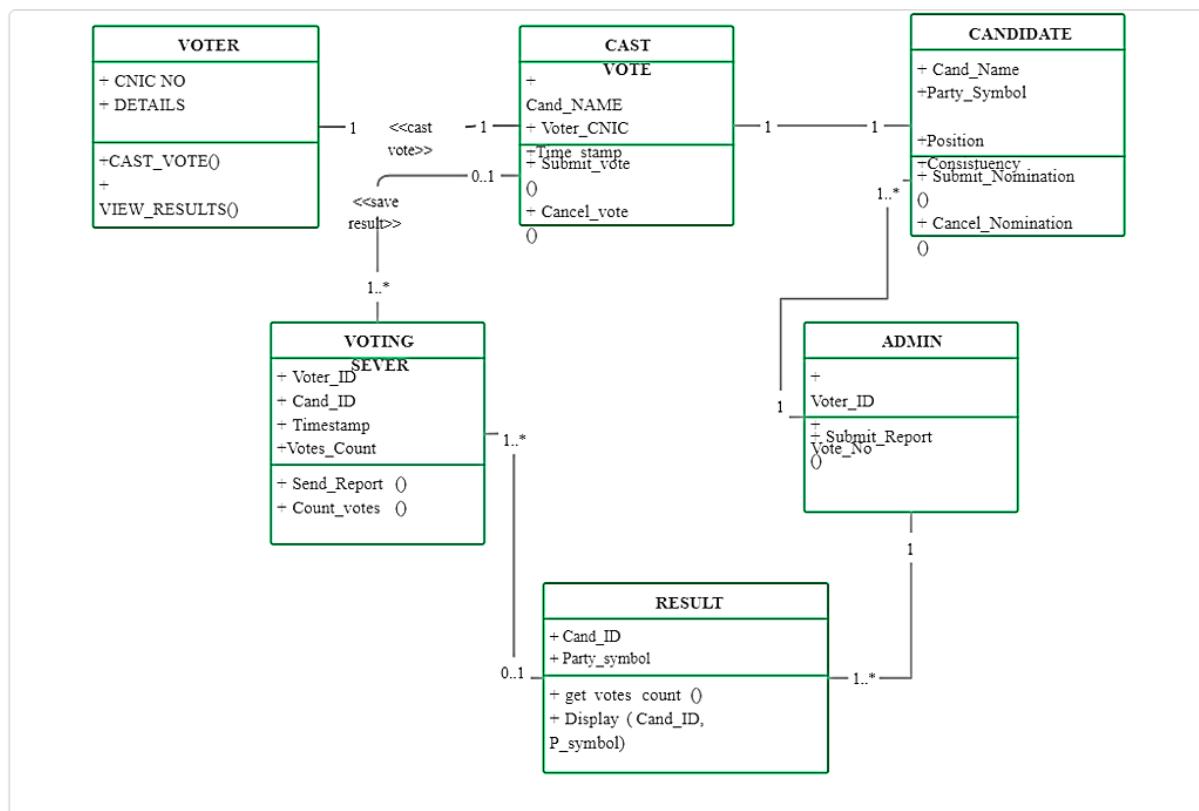
SYSTEM ARCHITECTURE – Example



USE CASE DIAGRAM – Example



CLASS DIAGRAM – Example



Result:

Thus, the system architecture, use case and class diagram created successfully.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	7
Title of Experiment	Design a Entity relationship diagram
Name of the candidate	JAYKUMAR PATEL (RA2111003011218)
Team Members	1. SHRAVANI S. MANE (RA2111003011223) 2. DEEPTHI AKKIPALI (RA2111003011247)
Register Number	RA2111003011218
Date of Experiment	09/03/2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To create the Entity Relationship Diagram

Team Members:

S No	Register No	Name	Role
1	RA2111003011223	SHRAVANI S. MANE	Rep
2	RA2111003011247	DEEPTHI AKKIPALLI	Member
3	RA2111003011218	JAYKUMAR PATEL	Member

<ER Diagram >

*/ ER Diagram, Notation and Example

What is ER Diagram?

- ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.
- ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.
- At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

What is ER Model?

- ER Model stands for Entity Relationship Model is a high-level conceptual data model diagram. ER model helps to systematically analyze data requirements to produce a well-designed database.
- ER Model represents real-world entities and the relationships between them. Creating an ER Model in DBMS is considered as a best practice before implementing your database.
- ER Modeling helps you to analyze data requirements systematically to produce a well-designed database. So, it is considered a best practice to complete ER modeling before implementing your database.

Why use ER Diagrams?

Here, are prime reasons for using the ER Diagram

- Helps you to define terms related to entity relationship modeling
- Provide a preview of how all your tables should connect, what fields are going to be on each table
- Helps to describe entities, attributes, relationships

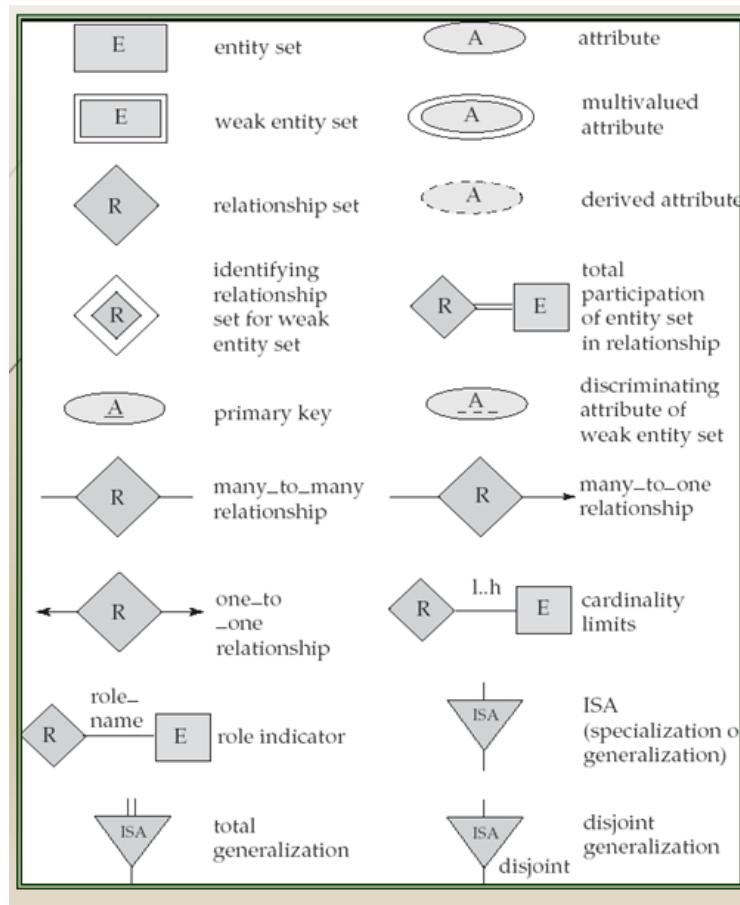
- ER diagrams are translatable into relational tables which allows you to build databases quickly
- ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications
- The database designer gains a better understanding of the information to be contained in the database with the help of ERP diagram
- ERD Diagram allows you to communicate with the logical structure of the database to users

Components of the ER Diagram

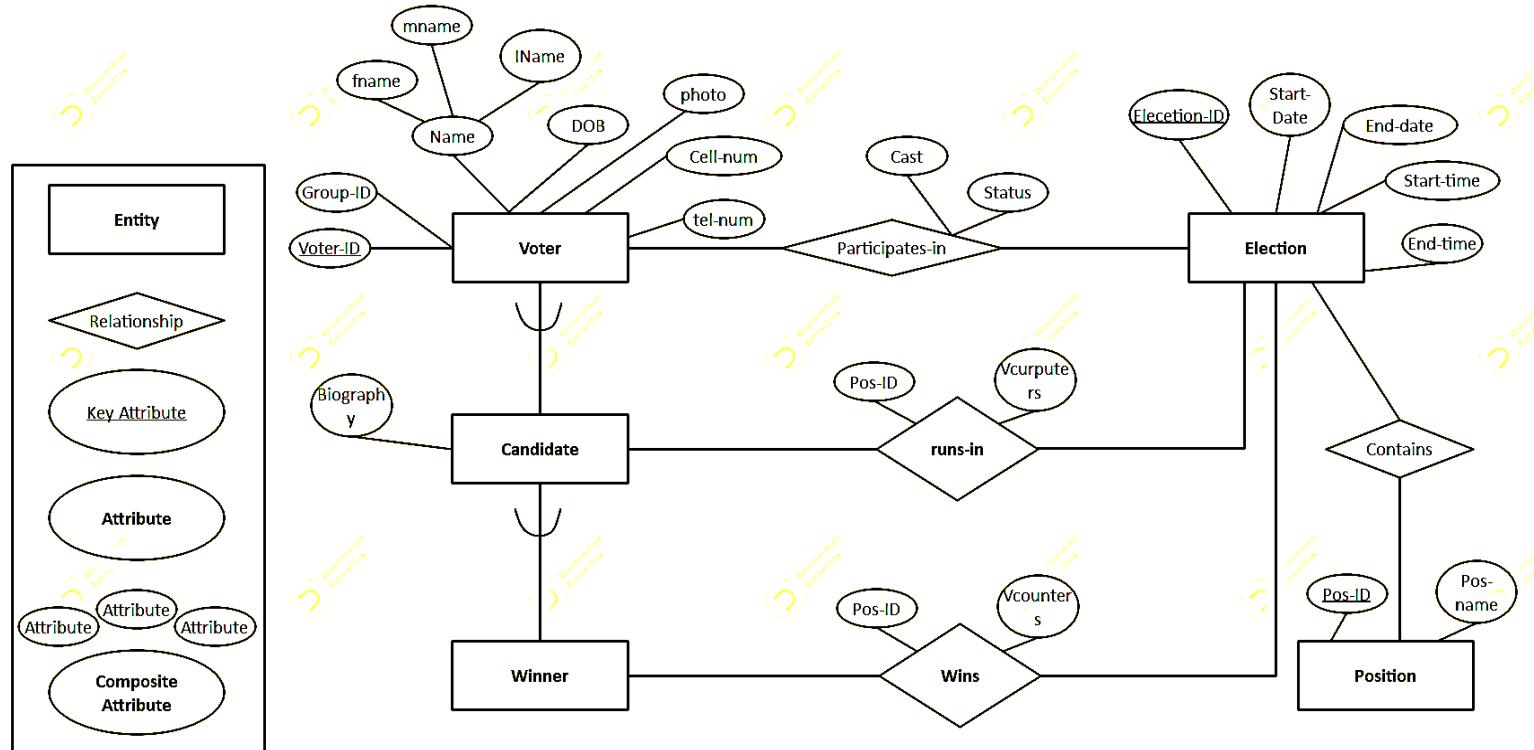
This model is based on three basic concepts: Entities, Attributes, Relationships

ER Diagram – Notations

- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Lines link attributes to entity sets and entity sets to relationship sets.
- Ellipses represent attributes
- Double ellipses represent multivalued attributes.
- Dashed ellipses denote derived attributes.
- Underline indicates primary key attributes



ER Diagram of University Database



ADDITIONAL NOTES

- A database can be modeled as a collection of entities, relationship among entities.
- An entity is an object that exists and is distinguishable from other objects.

Example: specific person, company, event, plant

- Entities have attributes.

Example: people have names and addresses

- An entity set is a set of entities of the same type that share the same properties.

Example: set of all persons, companies, trees, holidays

- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing binary relationship sets.
- We express cardinality constraints by drawing either a directed line (->), signifying “one,” or an undirected line (—), signifying “many,” between the relationship set and the entity set.

- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.

Example: customer = (customer-id, customer-name, customer-street, customer-city)
loan = (loan-number, amount)

- Domain – the set of permitted values for each attribute

- Attribute types:

1. Simple and composite attributes.
 2. Single-valued and multi-valued attributes
- E.g. multivalued attribute: phone-numbers

3. Derived attributes-Can be computed from other attributes

E.g. age, given date of birth

Cardinality

- For a binary relationship set the mapping cardinality must be one of the following types:

1. One to one

A customer is associated with at most one loan via the relationship borrower. A loan is associated with at most one customer via borrower

2. One to many

A loan is associated with at most one customer via borrower, a customer is associated with several (including 0) loans via borrower

3. Many to one

A loan is associated with several (including 0) customers via borrower, a customer is associated with at most one loan via borrower

4. Many to many

A loan is associated with several (including 0) customers via borrower, a customer is associated with several loans (including 0) via borrower

Weak Entity Set

- An entity set that does not have a primary key is referred to as a weak entity set and represented by double outlined box in E-R diagram.

Example : Consider the entity set payment which got three attributes : payment_number, payment_date and payment_amount. Payment numbers are sequential starting from 1 generally separately for each loan. Although each payment entity is distinct, payments for different loans may share the same payment number. Thus this entity set does not have a primary key.

Discriminator

- The discriminator (or partial key) of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set

Example: discriminator of weak entity set payment is the attribute payment_number since for each loan a payment number uniquely identifies one single payment for that loan.

Specialization-Generalization-ISA

- E-R model provides means of representing these distinctive entity groupings

- Process of designating subgroupings within an entity set is called specialization depicted by triangle component labelled ISA ("is a")

- Bottom up design process in which multiple entity sets are synthesized into higher level entity set - Generalization

- ISA relationship may also be referred to as superclass-subclass relationship

- Higher and lower level entity sets are designated by the terms superclass and subclass.

- Specialization and generalization are simple inversions of each other; they are represented in an E-R diagram in the same way.

Total & Partial Participation

- Total participation (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set

E.g. participation of loan in borrower is total, every loan must have a customer associated to it via borrower

- Partial participation: some entities may not participate in any relationship in the relationship set

Example: participation of customer in borrower is partial

Cardinality limits

- Cardinality limits can also express participation constraints
- Minimum and maximum cardinality is expressed as l..h where l is the minimum and h is the maximum cardinality
- Minimum value of 1 indicates total participation of entity set in relationship set
- Maximum value of 1 indicates entity participates in atmost one relationship set.
- Maximum value of * indicates no limit

Role indicator

- Entity sets of a relationship need not be distinct
- The labels “manager” and “worker” are called roles; they specify how employee entities interact via the works-for relationship set.
- Roles are indicated in E-R diagrams by labeling the lines that connect diamonds to rectangles.
- Role labels are optional, and are used to clarify semantics of the relationship

Disjoint Generalization

- Disjointness constraint requires that an entity belong to more than one lower level entity set.
- Example: account entity can satisfy only one condition for account_type attribute ; entity can either be savings or chequing account but not both.

Result:

Thus, the entity relationship diagram was created successfully.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	8
Title of Experiment	Develop a Data Flow Diagram (Process-Up to Level 1)
Name of the candidate	JAY KUMAR PATEL(RA2111003011218)
Team Members	DEEPTHI AKKIPALLI(RA2111003011247) SHRAVANI S. MANE (RA2111003011223)
Register Number	(RA2111003011218)
Date of Experiment	

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To develop the data flow diagram up to level 1 for the <project name>

Team Members:

S No	Register No	Name	Role
1	SHRAVANI MANE	RA2111003011223	Rep
2	JAY KUMAR PATEL	RA2111003011218	Member
3	DEEPTHI AKKIPALLI	RA2111003011247	Member

<DFD >

Result:

Thus, the data flow diagrams have been created for the <project name>.

Data Flow Diagram

The DFD takes an input-process-output view of a system. That is, data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software. Data objects are represented by labeled arrows, and transformations are represented by circles (also called bubbles). The DFD is presented in a hierarchical fashion. That is, the first data flow model (sometimes called a level 0 DFD or context diagram) represents the system as a whole. Subsequent data flow diagrams refine the context diagram, providing increasing detail with each subsequent level.

The data flow diagram enables you to develop models of the information domain and functional domain. As the DFD is refined into greater levels of detail, you perform an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of data as it moves through the processes that embody the application.

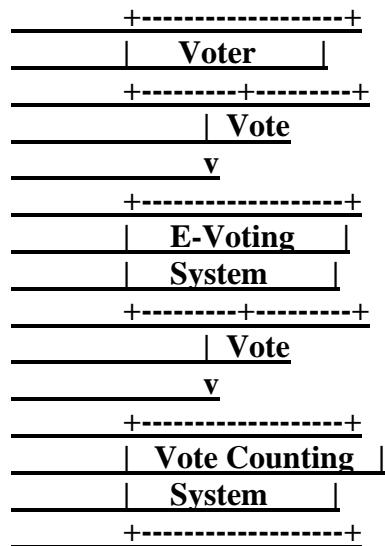
A few simple guidelines can aid immeasurably during the derivation of a data flow diagram:

- (1) Level 0 data flow diagram should depict the software/system as a single bubble;
- (2) Primary input and output should be carefully noted;
- (3) Refinement should begin by isolating candidate processes, data objects, and data stores to be represented at the next level;

- (4) All arrows and bubbles should be labeled with meaningful names;
- (5) Information flow continuity must be maintained from level to level and
- (6) One bubble at a time should be refined. There is a natural tendency to overcomplicate the data flow diagram. This occurs when you attempt to show too much detail too early or represent procedural aspects of the software in lieu of information flow.

***/ For Example**

DFD Level 0

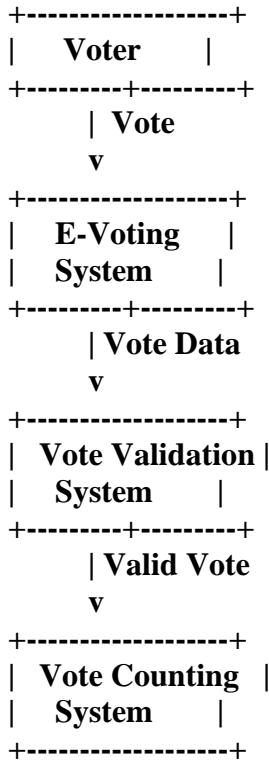


In this diagram, there are three main entities: the voter, the e-voting system, and the vote counting system.

The voter submits their vote to the e-voting system, which is responsible for verifying the voter's eligibility to vote and ensuring the integrity of the vote. Once the vote is verified, it is sent to the vote counting system for tabulation.

Note that this is a simplified example, and in a real e-voting system there would be many more details and checks involved at each stage of the process.

DFD Level 1



In this diagram, the e-voting system is broken down into three subsystems: the vote data validation system and the vote counting system, in addition to the voter entity.

After the voter submits their vote to the e-voting system, the vote data is transmitted to the vote data validation system. Here, the vote data is validated to ensure that it meets the necessary requirements for a valid vote. If the vote is valid, it is then sent to the vote counting system for tabulation. The vote counting system counts the votes and provides the final election results.

Note that in a real e-voting system, there may be additional subsystems and processes involved in ensuring the security and integrity of the voting process.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	9
Title of Experiment	Design a Sequence and Collaboration Diagram
Name of the candidate	JAYKUMAR PATEL(RA2111003011218)
Team Members	1. SHRAVANI S. MANE (RA2111003011223) 2.DEEPTHI AKKIPALI (RA2111003011247)
Register Number	RA2111003011218
Date of Experiment	21/04/2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

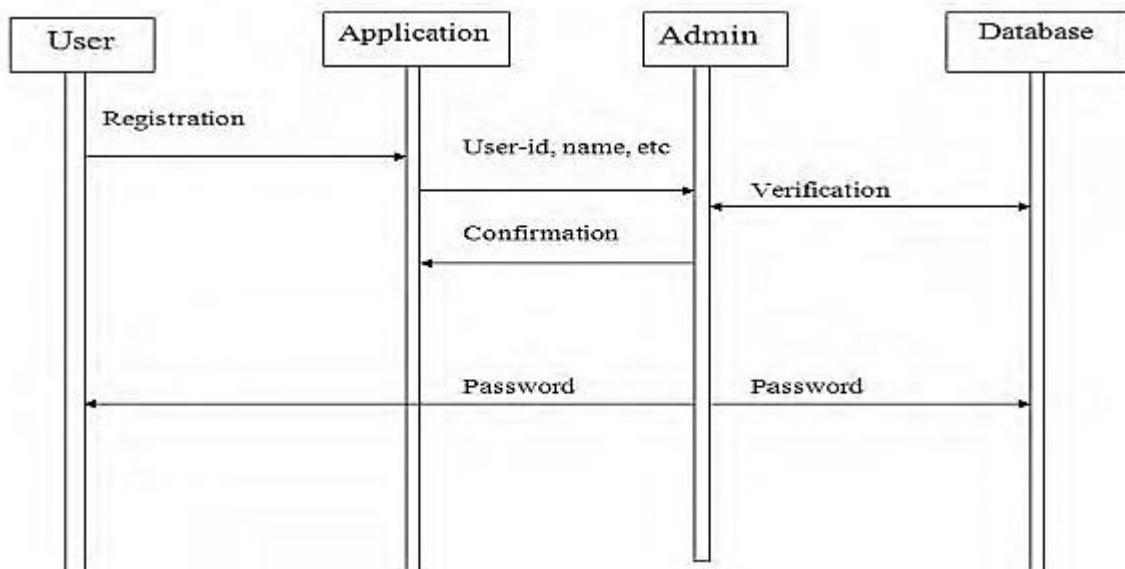
Aim

To create the sequence and collaboration diagram for the <project name>

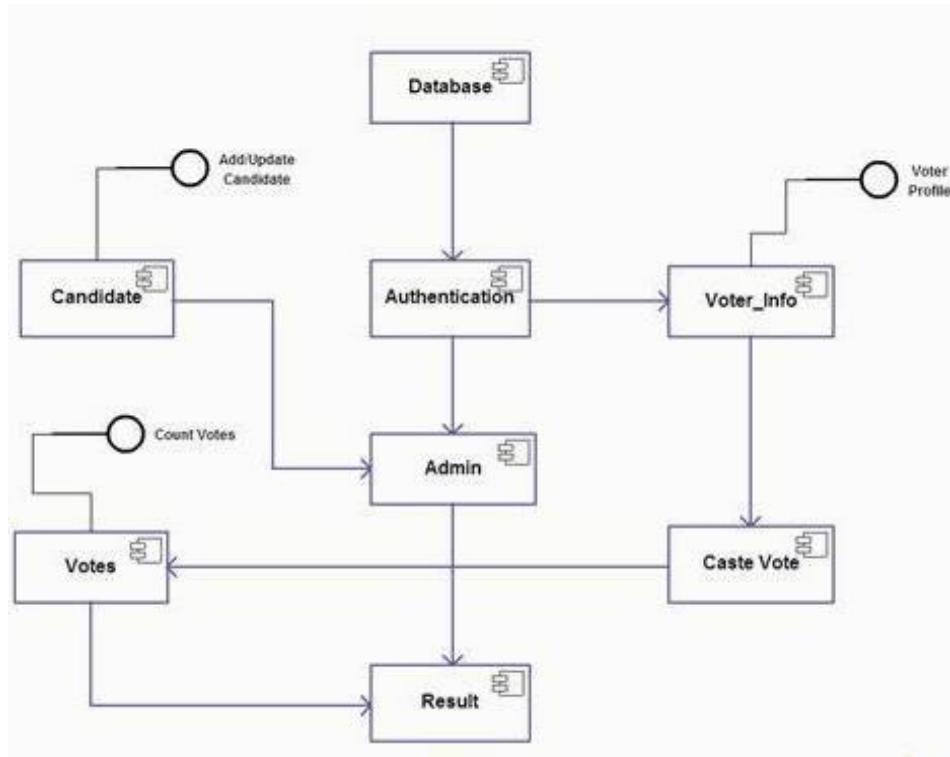
Team Members:

S No	Register No	Name	Role
1	RA2111003011223	SHRAVANI S. MANE	Rep/Member
2	RA2111003011247	DEEPTHI AKKIPALLI	Member
3	RA2111003011218	JAYKUMAR PATEL	Member

Sequence Diagram



Collaboration Diagram



Result:

Thus, the sequence and collaboration diagrams were created for the <project name>.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	10
Title of Experiment	Develop a Testing Framework/User Interface
Name of the candidate	JAYKUMAR PATEL(RA2111003011218)
Team Members	1. SHRAVANI S. MANE (RA2111003011223) 2. DEEPTHI AKKIPALI (RA2111003011247)
Register Number	RA2111003011218
Date of Experiment	22/04/2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To develop the testing framework and/or user interface framework for the android voting system

Team Members:

S No	Register No	Name	Role
1	RA2111003011223	SHRAVANI S. MANE	Rep/Member
2	RA2111003011247	DEEPTHI AKKIPALLI	Member
3	RA2111003011218	JAYKUMAR PATEL	Member

Executive Summary

Scope:

The scope of testing for the software application for Android voting systems includes verifying and validating the functionality, performance, security, and usability of the application. The testing process will ensure that the application meets the requirements and expectations of the stakeholders and is free from defects that may impact its integrity, reliability, and accuracy.

Objective:

The objective of testing the software application for Android voting systems is to identify and rectify any defects, vulnerabilities, or shortcomings in the application to ensure its proper functioning and reliability. The testing process will aim to verify if the application is working as intended, meets the specified requirements, and is user-friendly, secure, and efficient in handling the voting process.

Approach:

The approach to testing the software application for Android voting systems will follow a systematic and structured approach, which includes the following steps:

Requirements Analysis: Review and analyze the requirements of the software application, including functional and non-functional requirements, to understand the scope and expectations of the application.

Test Planning: Develop a comprehensive test plan that outlines the objectives, scope, resources, schedule, and deliverables of the testing process. Define the testing strategies, techniques, and tools to be used.

Test Design: Create detailed test cases and test scenarios based on the requirements and test objectives. Define the test data, expected results, and preconditions for each test case.

Test Environment Setup: Prepare the testing environment, including setting up the Android devices, emulators, network configurations, and necessary software and hardware components.

Test Plan

1. **Test Objectives:** Clearly define the objectives of the testing process, such as verifying the functionality, performance, security, and usability of the application, and ensuring it meets the requirements and expectations of the stakeholders.
2. **Scope:** Define the scope of testing, including the features, functionalities, and components of the application to be tested, as well as any exclusion or limitations.
3. **Test Schedule:** Develop a timeline for the testing process, including milestones, deadlines, and dependencies, to ensure timely completion of testing activities.
4. **Test Resources:** Identify and allocate the necessary resources for testing, including personnel, devices, emulators, testing tools, and test data.
5. **Test Techniques:** Specify the testing techniques to be used, such as black-box testing, white-box testing, grey-box testing, and any other relevant techniques based on the nature of the application and its requirements.
6. **Test Environment:** Define the testing environment, including the Android devices or emulators to be used, network configurations, and any other hardware or software requirements.
7. **Test Data:** Define the test data to be used for testing, including different scenarios, inputs, and expected outputs, to thoroughly test the application's functionality and performance.
8. **Test Cases:** Develop detailed test cases based on the requirements and objectives of the testing process. Each test case should include the test scenario, test data, expected results, and any preconditions.

Scope of Testing

The scope of testing for the software application for Android voting systems includes verifying the functionality, performance, security, and usability of the application. This encompasses testing the features, functionalities, and components of the application to ensure they meet the requirements and expectations of stakeholders. It may also include performance testing, security testing, and usability testing to thoroughly evaluate the application's performance, security measures, and user-friendliness. The scope of testing may be defined based on the project requirements and objectives, and may be subject to any limitations or exclusions specified in the test plan.

Functional:

In the functional testing scope for the software application for Android voting systems, all modules are typically covered to ensure that they are functioning as expected and meeting the defined requirements. However, there may be exceptions if certain modules are not applicable to the application or if they are excluded based on specific project requirements or limitations, which would be documented in the test plan.

Automation testing may be used to cover functional test cases, but it may not necessarily cover all test cases. The extent of automation coverage depends on various factors such as project timeline, available resources, and complexity of the application. Automation may focus on critical path test cases or high-impact test cases that are prone to regression, and repetitive testing tasks that can be automated for efficiency.

Regression testing, which aims to ensure that previously working functionalities are not affected by new changes or updates, may also be included in the scope of functional testing. This helps in identifying any regression issues that may arise due to changes in the application or its environment.

Non-Functional:

1. **Performance Testing:** Testing the application's performance, including response time, throughput, and scalability, to ensure it meets the defined performance requirements.
2. **Security Testing:** Testing the application's security measures, such as authentication, authorization, data encryption, and protection against vulnerabilities like SQL injection or cross-site scripting (XSS).
3. **Usability Testing:** Testing the application's user-friendliness, including ease of use, navigation, and overall user experience, to ensure it meets the needs of its intended users.
4. **Compatibility Testing:** Testing the application's compatibility with different devices, operating systems, browsers, or network configurations, to ensure it works well in various environments.
5. **Reliability and Availability Testing:** Testing the application's reliability, stability, and availability, including testing for error handling, fault tolerance, and disaster recovery, to ensure it operates consistently and is accessible when needed.
6. **Load and Stress Testing:** Testing the application's performance under load or stress conditions, such as high concurrent users, high data volume, or peak traffic, to identify any performance or scalability issues.

7. **Localization and Internationalization Testing:** Testing the application's localization and internationalization features, such as language translations, date/time formats, and currency conversions, to ensure it is suitable for different regions and cultures.

Types of Testing, Methodology, Tools

Category	Methodology	Tools Required
a. Functional Testing	Waterfall	a. Appium or Espresso for mobile application testing.
b. Performance Testing	agile	b. JUnit or TestNG for unit testing.
c. Security Testing	hybrid methodologies.	c. Selenium or Appium for web testing.
d. Usability Testing	Waterfall follows a sequential approach with predefined phases	d. JIRA or TestRail for test management.
e. Compatibility Testing:		e. LoadRunner or JMeter for performance testing.
f. Regression Testing	while agile emphasizes iterative and collaborative development with frequent feedback loops.	f. OWASP ZAP or Burp Suite for security testing.
g. Load and Stress Testing		g. Cucumber or SpecFlow for behavior-driven development (BDD) testing.
h. Localization and Internationalization Testing	A hybrid methodology may combine elements of both.	h. Postman or SOAPUI for API testing.

Result:

Thus, the testing framework/user interface framework has been created for the android voting system



H

School of Computing

SRM IST, Kattankulathur — 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	11
Title of Experiment	Test Cases
Name of the candidate	JAY PATEL
Team Members	SHRAVANI MANE (RA2111003011223) DEEPTHI AKKIPALL (RA2111003011247)
Register Number	RA2111003011218
Date of Experiment	

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To develop the test cases manual for the <project name>

Team Members:

S No	Register No	Name	Role
1	(RA2111003011223)	SHRAVANI MANE	Rep
2	(RA2111003011218)	JAY PATEL	Member
3			Member

<Utilize the templates below and incorporate the project's test cases - Manual Test case to be written for at least one module >

Result:

Thus, the test case manual has been created for the <project name>.

*/ For example

Test Case

Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks

	Verify User Registration from India	Accept Valid India Mobile Number on the Page#1	1. User clicks on User Registration link 2. Enter the mobile Number on the text box 3. Click Register button	User should be taken to the next page for entering more user details		Pass / Failure	success
	Verify User Registration from India	Don't Accept Non IndianMobile Number on the Page#1					

Non-Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
	To increase usability, efficiency, maintainability and portability of the product. To help in the reduction of production risk related with non-functional aspects of the product. To help in the reduction of cost related with non-functional aspects of the product. To optimize the installation, execution and monitoring	Compatibility testing: A type of testing to ensure that a software program or system is compatible with other software programs or systems. Compliance testing: A type of testing to ensure that a software program or system	1)Performance Testing: Test the response time of the application when multiple users try to vote simultaneously. 2)Security Testing: Verify that the user data is encrypted and secure during transmission. 3)Compatibility Testing:	Performance testing would aim to measure how the system responds under normal and peak loads. This could include testing the response time for users to submit their votes, measuring the throughput of the system, and monitoring the system's resource utilization to ensure it can handle the expected	The actual outcome of non-functional testing in an Android voting system will depend on the specific requirements and goals of the system, as well as the testing methodologies and tools used. The actual outcome of non-functional testing	If testing is ongoing, the status may indicate which execution qualities have been tested, which are still being tested, and what the results of the testing have been so far. For example, the status may indicate that performance testing has been completed and the system is 'able to	Testing should be 1) comprehensive: Non-functional testing should cover a wide range of scenarios and use cases to ensure that the system is tested under a variety of conditions. 2)Testing should be ongoing: Non-functional testing should be an ongoing process, with regular testing performed throughout the development lifecycle and

	<p>way of the product.</p> <p>To collect and produce measurements and metrics for internal research and development.</p> <p>To improve and enhance knowledge of the product behavior and technologies in use.</p>	<p>meets a specific compliance standard, such as HIPAA or Sarbanes-Oxley.</p> <p>Endurance testing: A type of testing to ensure that a software program or system can handle a long-term, continuous load.</p> <p>Load testing: A type of testing to ensure that a software program or system can handle a large number of users or transactions.</p> <p>Performance testing: A type of testing to ensure that a software program or system</p>	<p>4)Usability Testing: Evaluate the application's user interface and check if it is easy to use and navigate.</p> <p>5)Reliability Testing: Test the application's stability and ensure that it is reliable.</p> <p>6)Scalability: This refers to the ability of the system to handle increasing amounts of data or traffic without affecting performance or reliability.</p>	<p>number of users.</p> <p>Reliability testing would aim to measure how the system performs under different scenarios, such as when there are network outages or other disruptions. This could include testing the system's ability to recover from errors, checking the system's availability, and verifying that the system can continue to operate in the event of hardware or software failures.</p>	<p>will also depend on the scope and depth of the testing performed. For example, if testing is limited to only a few scenarios or use cases, the actual outcome may be less comprehensive than if testing is performed on a wide range of scenarios and use cases.</p> <p>Similarly, if testing is limited to only a few tools or methodologies, the actual outcome may be less robust than if multiple tools and methodologies are used</p>	<p>handle the expected load, but security testing is still ongoing and vulnerabilities are being identified and addressed.</p> <p>If testing has been completed, the status may indicate whether the system meets the necessary execution qualities, or if there are still issues that need to be addressed. The status may also provide a summary of the testing results, such as identifying areas of the system that performed well and areas that need improvement.</p>	<p>during operation of the system.</p> <p>3)Testing should be automated: Automated testing tools can help to improve the efficiency and accuracy of non-functional testing, and enable testing to be performed more frequently.</p> <p>4)Testing should be repeatable: Tests should be repeatable to ensure that the same results can be achieved consistently, and to enable regression testing to be performed when changes are made to the system.</p> <p>5)Testing should be based on industry standards: Testing should be based on established industry standards and best practices, such as those outlined by the Software Engineering Institute (SEI) or the International Organization for Standardization (ISO).</p>
--	---	---	--	--	---	---	---

RESULT: successfully done that To develop the test cases manual for the <project name>



SRM

INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	12
Title of Experiment	Manual Test Case Reporting
Name of the candidate	JAYKUMAR PATEL
Team Members	SHRAVANI MANE(RA2111003011223) DEEPTHI AKKIPALLI(RA2111003011247)
Register Number	RA2111003011218
Date of Experiment	

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To prepare the manual test case report for the <project name>

Team Members:

S No	Register No	Name	Role
1	(RA2111003011223)	SHRAVANI MANE	Rep/Member
2	(RA21113011218)	JAY PATEL KAMLESHBHAI	Member
3	(RA2111003011247)	DEEPTHI AKKIPALLI	Member

<Manual Test Case Report to be incorporated >

<< Summarize the current status of the Testing>>

<<present obstacles to proceed further >>

<< Seek help from stakeholders to remove obstacles/constraints>>

Category	Progress Against Plan	Status
Functional Testing : Functional testing in an Android voting system would involve testing the system's functionality to ensure that it is working as expected and meeting the requirements of the system. This would typically involve testing various aspects of the system, such as user interface, user authentication and authorization, data input and storage, processing of votes, and generation of results.	Green / Amber / Red Green - Progress is on track and meeting the plan. For example, if the development of the voting system is progressing smoothly and all the milestones are being met on time, the status could be marked as Green. Amber - There may be some concerns or risks that need to be addressed. For example, if there are some technical issues	Not-Started / In-Progress / Completed Not-Started - This status would apply to tasks or activities that have not yet been started. For example, if the development team has not yet begun working on the user interface design for the voting system, the status for that task could be marked as Not-Started. In-Progress - This status would apply to tasks or activities that

<p>User interface testing: This would involve testing the system's user interface to ensure that it is user-friendly and easy to use. This could involve testing the layout and design of the screens, buttons, menus, and other interface elements.</p> <p>Authentication and authorization testing: This would involve testing the system's authentication and authorization features to ensure that only authorized users can access the system and that they can only perform actions that they are authorized to perform.</p>	<p>that need to be resolved or some aspects of the voting system that are not meeting the requirements, the status could be marked as Amber.</p> <p>Red - There are significant issues or risks that need to be addressed immediately. For example, if the development of the voting system is significantly behind schedule or there are critical technical issues that need to be resolved before the system can be deployed, the status could be marked as Red.</p>	<p>are currently underway. For example, if the development team is currently working on the implementation of the vote counting algorithm for the system, the status for that task could be marked as In-Progress.</p> <p>Completed - This status would apply to tasks or activities that have been finished and are considered done. For example, if the testing team has completed testing the functionality of the voting system and has signed off on its readiness for deployment, the status for that task could be marked as Completed.</p>
<p>Non-Functional Testing : Non-functional testing in an Android voting system would involve testing the system's non-functional requirements, such as performance, security, reliability, usability, and scalability. Here are some examples of non-functional tests that could be performed on an Android voting system:</p> <p>Performance Testing: This involves testing the system's performance under various conditions such as different numbers of concurrent users or different loads. The goal is to ensure that the system can handle the expected number of voters without any significant degradation in performance.</p> <p>Security Testing: This involves testing the system's security features to ensure that they are robust enough to protect against potential threats, such as hacking attempts or</p>	<p>Green - Progress is on track and meeting the plan. For example, if the non-functional testing activities are progressing smoothly, all the required tests have been executed and are meeting the acceptance criteria, the status could be marked as Green.</p> <p>Amber - There may be some concerns or risks that need to be addressed. For example, if some of the non-functional tests are not meeting the acceptance criteria, or if there are delays in executing the tests, the status could be marked as Amber.</p> <p>Red - There are significant issues or risks that need to be addressed immediately. For example, if the system is not performing as expected, or if there are major security vulnerabilities that need to be addressed, the status could be marked as Red.</p>	<p>Not-Started - This status would apply to non-functional testing activities that have not yet been started. For example, if the team has not yet begun performance testing, the status for that activity could be marked as Not-Started.</p> <p>In-Progress - This status would apply to non-functional testing activities that are currently underway. For example, if the team is currently performing security testing on the voting system, the status for that activity could be marked as In-Progress.</p> <p>Completed - This status would apply to non-functional testing activities that have been finished and are considered done. For example, if the team has completed all the required scalability testing and all the tests have passed, the status for that activity could be marked as Completed.</p>

unauthorized access to voter data.		
Reliability Testing: This involves testing the system's ability to perform consistently and reliably over time. This could involve testing the system's ability to handle unexpected errors or faults without crashing or losing data.		

Functional	Test Case Coverage (%)	Status	
Module ID :1. User Authentication Module - Module ID: UA 2.Voter Registration Module - Module ID: VR 3.Candidate Registration Module - Module ID: CR 4.Vote Casting Module - Module ID: VC 5.Vote Counting Module - Module ID: VCO 6.Reporting Module - Module ID: RM 7.Admin Module - Module ID: AM 8.Data Management Module - Module ID: DM	30% A test case coverage of 30% in an Android voting system suggests that only 30% of the possible test cases have been created and executed. This means that there is a significant portion of the system that has not been adequately tested, and there may be potential issues or defects that have not yet been discovered. To improve the test case coverage, the testing team could review the existing test cases and identify any gaps in the testing coverage. They could also work with the development team to identify any additional features or functionality that needs to be tested and create new test cases accordingly. Increasing the test case coverage can	Not-Started / In-Progress / Completed : Not-Started - This status would apply to functional testing activities that have not yet been started. For example, if the team has not yet begun testing the user authentication module, the status for that activity could be marked as Not-Started. In-Progress - This status would apply to functional testing activities that are currently underway. For example, if the team is currently testing the candidate registration module, the status for that activity could be marked as In-Progress. Completed - This status would apply to functional testing activities that have	

	help ensure that the system is more thoroughly tested	been finished and are considered done. For example, if the team has completed testing the voter registration module and all the tests have passed, the status for that activity could be marked as Completed.

Result:

Thus, the test case report has been created for the <project name>.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	13
Title of Experiment	Provide the details of Architecture Design/Framework/Implementation
Name of the candidate	JAYKUMAR PATEL(RA2111003011218)
Team Members	SHRAVANI MANE(RA2111003011223) DEEPTHI AKKIPALLI(RA2111003011247)
Register Numbers	RA2111003011218
Date of Experiment	25/04/2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

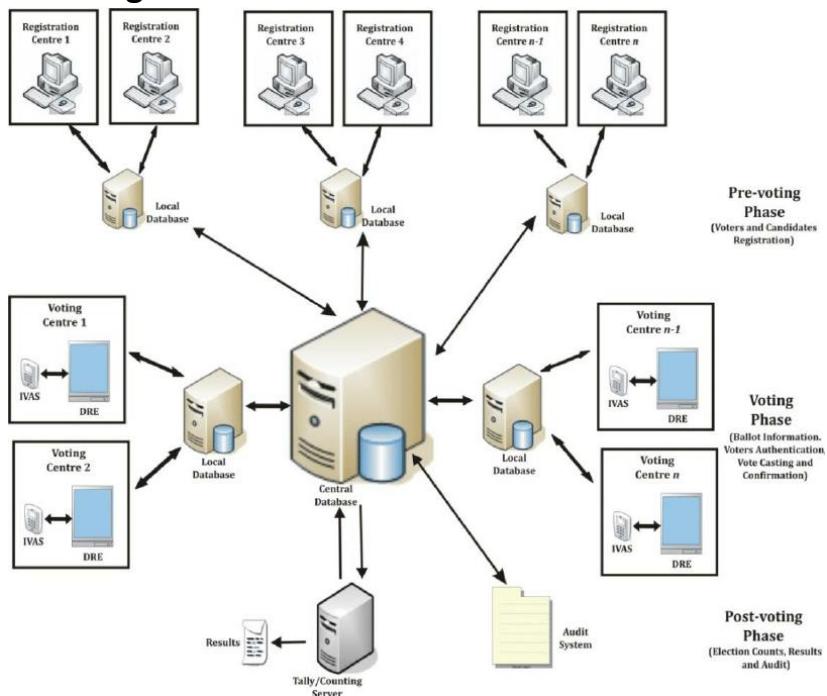
Aim

To provide the details of architectural design/framework/implementation

Team Members:

S No	Register No	Name	Role
	RA2111003011223	SHRAVANI MANE	Rep/Member
2	RA2111003011218	JAYKUMAR PATEL	Member
3	RA2111003011247	DEEPTHI AKKIPALLI	Member

architectural design

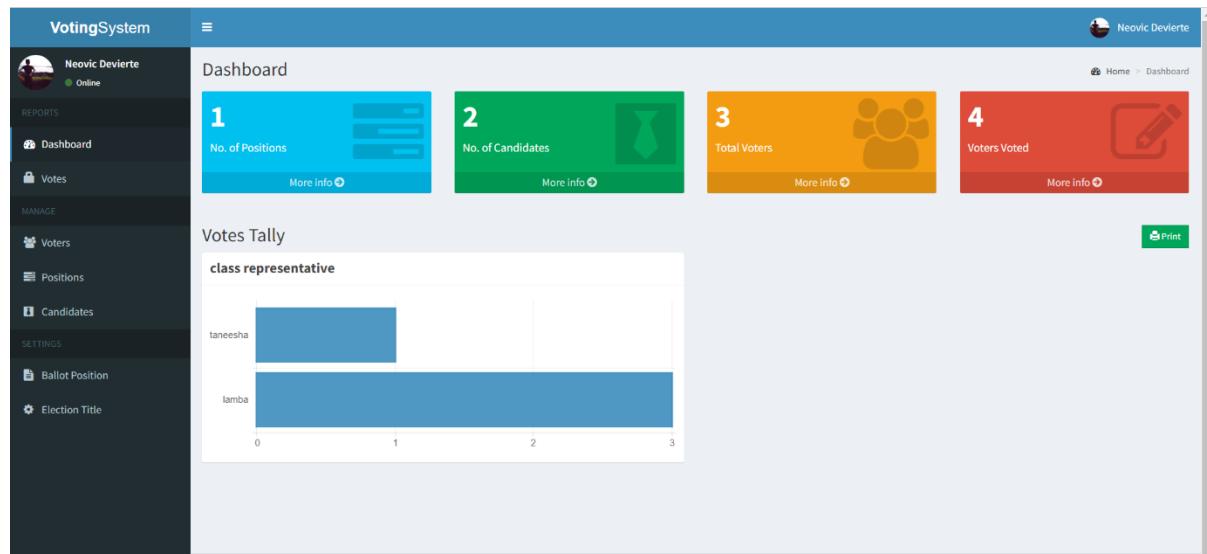


1. Front-end User Interface: This is the interface that voters use to access the online voting system. It typically includes a website or a mobile app that allows voters to register, log in, and cast their votes. It may also include features such as ballot selection, vote submission, and voter authentication.

2. Voter Registration and Authentication: This component handles the process of registering voters and verifying their identities. It may include features such as voter registration forms, identity verification methods (e.g., email or SMS verification), and voter database management.
3. Ballot Generation: This component is responsible for generating the ballots for each election. It may include features such as ballot creation, ballot customization (e.g., based on voter's location or eligibility), and ballot distribution.
4. Vote Casting: This component handles the process of voters casting their votes. It may include features such as ballot selection, vote submission, and voter confirmation.
5. Vote Counting and Tabulation: This component is responsible for counting and tabulating the votes. It may include features such as vote counting algorithms, data encryption for security, and audit trails for transparency.

Framework

An online voting system framework typically includes the legal and regulatory, technical, security, user interface, operational, audit and transparency, and governance and administration aspects of the system. It provides predefined rules, guidelines, and tools for building and implementing a secure, transparent, and user-friendly online voting system.



VotingSystem


Neovic Devierte
Online

REPORTS

- [Dashboard](#)
- [Votes](#)

MANAGE

- [Voters](#)
- [Positions](#)
- [Candidates](#)

SETTINGS

- [Ballot Position](#)
- [Election Title](#)

Positions

[+New](#)

Description	Maximum Vote	Tools
class representative	2	Edit Delete

Showing 1 to 1 of 1 entries

Previous 1 Next

VotingSystem


SHRAVANI mane [LOGOUT](#)

CLASS VOTING

class representative

You may select up to 2 candidates

[Reset](#)

<input type="checkbox"/>	Platform		deepanshu lamba
<input type="checkbox"/>	Platform		kavya taneesha

[Preview](#) [Submit](#)

VotingSystem


Neovic Devierte
Online

REPORTS

- [Dashboard](#)
- [Votes](#)

MANAGE

- [Voters](#)
- [Positions](#)
- [Candidates](#)

SETTINGS

- [Ballot Position](#)
- [Election Title](#)

Votes

[Reset](#)

Position	Candidate	Voter
class representative	deepanshu lamba	
class representative	deepanshu lamba	
class representative	kavya taneesha	DEEPHI.
class representative	deepanshu lamba	SHRAVANI mane

Showing 1 to 4 of 4 entries

Previous 1 Next

VotingSystem	
 Neovic Devierte Online	
 Dashboard	
 Votes	
MANAGE	
 Voters	
 Positions	
 Candidates	
SETTINGS	
 Ballot Position	
 Election Title	

Voters List

[+New](#)
Show: entries
Search:

Lastname	Firstname	Photo	Voters ID	Tools
DEEPTHI	SHRAVANI		qpRHnA3iDtu46QI	Edit Delete
mane	JAY		WkGVZ5Mu7GBjzEL	Edit Delete
patel			IgMS2sbpZNnE3BP	Edit Delete

Showing 1 to 3 of 3 entries

✔ Success!
 Ballot Submitted

CLASS VOTING

You have already voted for this election.

[View Ballot](#)

CODE

```
-- phpMyAdmin SQL Dump
-- version 4.7.9
-- https://www.phpmyadmin.net/
-- 
-- Host: 127.0.0.1
-- Generation Time: May 18, 2018 at 07:34 AM
-- Server version: 10.1.31-MariaDB
-- PHP Version: 7.1.15
```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
*/;
/*!40101 SET NAMES utf8mb4 */;

-- 
-- Database: `votesystem` 

-----


-- 
-- Table structure for table `admin` 

-- 

CREATE TABLE `admin` (
  `id` int(11) NOT NULL,
  `username` varchar(50) NOT NULL,
  `password` varchar(60) NOT NULL,
  `firstname` varchar(50) NOT NULL,
  `lastname` varchar(50) NOT NULL,
  `photo` varchar(150) NOT NULL,
  `created_on` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- 
```

```
-- Dumping data for table `admin`  
--  
  
INSERT INTO `admin` (`id`, `username`, `password`, `firstname`, `lastname`, `photo`,  
`created_on`) VALUES  
(1, 'nurhodelta',  
'$2y$10$fLK8s7ZDnM.1lE7XMP.J6OuPbQ.DPUVKBo7rENnQY7gYq0xAzsKJy', 'Neovic',  
'Devierte', 'facebook-profile-image.jpeg', '2018-04-02');
```

```
--  
-- Table structure for table `candidates`  
--
```

```
CREATE TABLE `candidates` (  
    `id` int(11) NOT NULL,  
    `position_id` int(11) NOT NULL,  
    `firstname` varchar(30) NOT NULL,  
    `lastname` varchar(30) NOT NULL,  
    `photo` varchar(150) NOT NULL,  
    `platform` text NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--  
-- Table structure for table `positions`  
--
```

```
CREATE TABLE `positions` (  
    `id` int(11) NOT NULL,  
    `description` varchar(50) NOT NULL,  
    `max_vote` int(11) NOT NULL,
```

```
 `priority` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--  
-- Table structure for table `voters`  
--
```

```
CREATE TABLE `voters` (  
 `id` int(11) NOT NULL,  
 `voters_id` varchar(15) NOT NULL,  
 `password` varchar(60) NOT NULL,  
 `firstname` varchar(30) NOT NULL,  
 `lastname` varchar(30) NOT NULL,  
 `photo` varchar(150) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--  
-- Table structure for table `votes`  
--
```

```
CREATE TABLE `votes` (  
 `id` int(11) NOT NULL,  
 `voters_id` int(11) NOT NULL,  
 `candidate_id` int(11) NOT NULL,  
 `position_id` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--  
-- Indexes for dumped tables  
--
```

```
--  
-- Indexes for table `admin`  
  
--  
ALTER TABLE `admin`  
    ADD PRIMARY KEY (id);  
  
--  
-- Indexes for table `candidates`  
  
--  
ALTER TABLE `candidates`  
    ADD PRIMARY KEY (id);  
  
--  
-- Indexes for table `positions`  
  
--  
ALTER TABLE `positions`  
    ADD PRIMARY KEY (id);  
  
--  
-- Indexes for table `voters`  
  
--  
ALTER TABLE `voters`  
    ADD PRIMARY KEY (id);  
  
--  
-- Indexes for table `votes`  
  
--  
ALTER TABLE `votes`  
    ADD PRIMARY KEY (id);  
  
--  
-- AUTO_INCREMENT for dumped tables  
  
--
```

```
--  
-- AUTO_INCREMENT for table `admin`  
  
--  
ALTER TABLE `admin`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;  
  
--  
-- AUTO_INCREMENT for table `candidates`  
  
--  
ALTER TABLE `candidates`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=18;  
  
--  
-- AUTO_INCREMENT for table `positions`  
  
--  
ALTER TABLE `positions`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;  
  
--  
-- AUTO_INCREMENT for table `voters`  
  
--  
ALTER TABLE `voters`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;  
  
--  
-- AUTO_INCREMENT for table `votes`  
  
--  
ALTER TABLE `votes`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=81;  
COMMIT;  
  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
```

```
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

<?php include 'includes/session.php'; ?>
<?php include 'includes/header.php'; ?>
<body class="hold-transition skin-blue layout-top-nav">
<div class="wrapper">

<?php include 'includes/navbar.php'; ?>

<div class="content-wrapper">
<div class="container">

<!-- Main content -->
<section class="content">
<?php
    $parse = parse_ini_file('admin/config.ini', FALSE, INI_SCANNER_RAW);
    $title = $parse['election_title'];
?
<h1 class="page-header text-center title"><b><?php echo strtoupper($title); ?></b></h1>
<div class="row">
    <div class="col-sm-10 col-sm-offset-1">
        <?php
            if(isset($_SESSION['error'])){
?
                <div class="alert alert-danger alert-dismissible">
                    <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
                    <ul>
                        <?php
                            foreach($_SESSION['error'] as $error){
                                echo "
                                <li>".$error."</li>
                            }
                        </ul>
                    </div>
                <?php
            }
        ?
    </div>
</div>
```

```

        ";
    }
?>
</ul>
</div>
<?php
unset($_SESSION['error']);

}

if(isset($_SESSION['success'])){
echo "
<div class='alert alert-success alert-dismissible'>
<button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
<h4><i class='icon fa fa-check'></i> Success!</h4>
".$_SESSION['success']."
</div>
";
unset($_SESSION['success']);
}

?>

<div class="alert alert-danger alert-dismissible" id="alert"
style="display:none;">
<button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button>
<span class="message"></span>
</div>

<?php
$sql = "SELECT * FROM votes WHERE voters_id = '".$voter['id']."' ";
$query = $conn->query($sql);
if($query->num_rows > 0){

```

```

?>
<div class="text-center">
    <h3>You have already voted for this election.</h3>
    <a href="#view" data-toggle="modal" class="btn btn-flat btn-primary
btn-lg">View Ballot</a>
</div>
<?php
}
else{
?>
<!-- Voting Ballot -->
<form method="POST" id="ballotForm" action="submit_ballot.php">
<?php
    include 'includes/slugify.php';

$candidate = "";
$sql = "SELECT * FROM positions ORDER BY priority ASC";
$query = $conn->query($sql);
while($row = $query->fetch_assoc()){
    $sql = "SELECT * FROM candidates WHERE
position_id='".$row['id']."'";

$cquery = $conn->query($sql);
while($crow = $cquery->fetch_assoc()){
        $slug = slugify($row['description']);
        $checked = "";
        if(isset($_SESSION['post'][$slug])){
            $value = $_SESSION['post'][$slug];

            if(is_array($value)){
                foreach($value as $val){
                    if($val == $crow['id']){
                        $checked = 'checked';
                    }
                }
            }
        }
    }
}

```

```

        }
        else{
            if($value == $crow['id']){
                $checked = 'checked';
            }
        }
    }

    $input = ($row['max_vote'] > 1) ? '<input type="checkbox"' . $row['id'] . $checked . ' : '<input type="radio" class="flat-red '$slug.'" name="'. $slug .'[]' . $row['id'] . $checked . '" value="'. $row['id'] . $checked . '" :>' : '<input type="radio" class="flat-red '$slug.'" name="'. $slugify($row['description']) .'" value="'. $row['id'] . $checked . '" :>';

    $image = (!empty($row['photo'])) ? 'images/' . $row['photo'] : 'images/profile.jpg';

    $candidate .= '
<li>
    '.$input.'<button type="button" class="btn btn-primary btn-sm btn-flat clist platform" data-platform="'. $row['platform'].'" data-fullname="'. $row['firstname'] . ' ' . $row['lastname'] . '"><i class="fa fa-search"></i> Platform</button><span class="cname clist">'. $row['firstname'] . ' ' . $row['lastname'] . '</span>
</li>
';
}

```

```

    $instruct = ($row['max_vote'] > 1) ? 'You may select up to ' . $row['max_vote'] . ' candidates' : 'Select only one candidate';

```

```

echo '
<div class="row">
    <div class="col-xs-12">
        <div class="box box-solid" id="'. $row['id'] .'">
            <div class="box-header with-border">
                <h3 class="box-
title"><b>'. $row['description'] . '</b></h3>

```

```

        </div>
        <div class="box-body">
            <p>'$.instruct.'
                <span class="pull-right">
                    <button type="button" class="btn btn-success btn-sm btn-flat reset" data-desc=".slugify($row['description']).'"><i class="fa fa-refresh"></i>
                    Reset</button>
                </span>
            </p>
            <div id="candidate_list">
                <ul>
                    '$candidate.'
                </ul>
            </div>
            </div>
            </div>
            </div>
            '$;
        }

    ?>
    <div class="text-center">
        <button type="button" class="btn btn-success btn-flat" id="preview"><i class="fa fa-file-text"></i> Preview</button>
        <button type="submit" class="btn btn-primary btn-flat" name="vote"><i class="fa fa-check-square-o"></i> Submit</button>
    </div>
    </form>
    <!-- End Voting Ballot -->
    <?php

```

```
    }

<?php include 'includes/footer.php'; ?>
<?php include 'includes/ballot_modal.php'; ?>
</div>

<?php include 'includes/scripts.php'; ?>
<script>
$(function(){
$('.content').iCheck({
  checkboxClass: 'icheckbox_flat-green',
  radioClass: 'iradio_flat-green'
});
$(document).on('click', '.reset', function(e){
  e.preventDefault();
  var desc = $(this).data('desc');
  $('.'+desc).iCheck('uncheck');
});

$(document).on('click', '.platform', function(e){
  e.preventDefault();
  $('#platform').modal('show');
  var platform = $(this).data('platform');
  var fullname = $(this).data('fullname');
  $('.candidate').html(fullname);
  $('#plat_view').html(platform);
});
$('#preview').click(function(e){
  e.preventDefault();
  var form = $('#ballotForm').serialize();
  if(form == ""){
    $('.message').html('You must vote atleast one candidate');
    $('#alert').show();
  }
})
```

```

else{
$.ajax({
    type: 'POST',
    url: 'preview.php',
    data: form,
    dataType: 'json',
    success: function(response){
        if(response.error){
            var errmsg = "";
            var messages = response.message;
            for (i in messages) {
                errmsg += messages[i];
            }
            $('.message').html(errmsg);
            $('#alert').show();
        }
        else{
            $('#preview_modal').modal('show');
            $('#preview_body').html(response.list);
        }
    }
});
</script>
</body>
</html>

```

Result:

Thus, the details of architectural design/framework/implementation along with the screenshots were provided.