COSC510 – SOFTWARE PROJECT MANAGEMENT

Assessment 4 – Group Project

Technical Report (Project Management Plan) for Online Voting System

Anil Khattri

akhattri@myune.edu.au

220276347

Bikash Neupane

bneupan2@myune.edu.au

220245756

Sabin Dhital

sdhital@myune.edu.au

220264523

Table of Contents

Table of Figures

Technical Report (Project Management Plan) for Online Voting System

1. Introduction

1.1. Project Overview

The online voting system (OVS) aim to create a secure, user-friendly, and efficient platform for enabling users to vote online, will be developed. The system should manage many visitors, protect voters' privacy, prevent cheating, and count votes correctly as well as instantly.

1.2. Goals of the Project

Goals of the online voting system are:

1. Develop a secure platform for online voting.
2. Ensure the system is user-friendly and accessible to all voters.
3. Implement robust mechanisms to prevent fraud and ensure vote integrity.
4. Provide real-time results and analytics for election officials.
5. Ensure compliance with legal and regulatory requirements.

1.3. Project Scope

The Online Voting System will be designed, developed, evaluated, and deployed as part of the project. The main user functionalities are user registration, authentication, ballot generation, vote casting, vote counting, and result display. It is important to note that post-deployment support and maintenance will be offered too.

1.4. Key Milestones

1. Project Initiation and Planning: Week 1
2. Requirements Gathering and Analysis: Weeks 2-4
3. System Design: Weeks 5-8
4. Development Phase 1 (Core Features): Weeks 9-13
5. Development Phase 2 (Advanced Features): Weeks 13-17
6. Testing and QA: Weeks 17-20
7. User Training and Documentation: Weeks 20-23
8. Deployment and Go-Live: Week 24

1.5. Identification of Stakeholders

A stakeholder is any individual, group, or company, that is directly or indirectly involved in the project and who may affect or get affected by the outcome of the project (Geeks for Geeks, 2024). The stakeholder of online voting system is listed below:

1. Client: Government and Election Commissions.
2. End Users: Registered voters.
3. Administrators: Election officials and administrators.
4. Development Team: Software developers, testers, project managers.
5. Security Experts: Specialists in cybersecurity and data protection.
6. Support Team: Technical support and maintenance personnel.

2. Technical Solution

2.1. Overview of Solution

The Online Voting System will be a web-based application which is accessible through browsers on desktops and mobile devices. It will use modern web technologies and robust security protocols to ensure a secure and smooth voting experience to the end user.

2.2. Major Components

Major components of the online voting system are:

- **Frontend**: User interfaces for voters and administrators.
- **Backend**: Server-side logic, APIs, and databases.
- **Security Layer**: Encryption, authentication, and fraud detection mechanisms.
- **Analytics Engine**: Real-time data processing and reporting.
- **Deployment Infrastructure**: Cloud-based hosting and scaling solutions.

2.3. Logical System Design

2.3.1. Use Case Diagram

Use Case is vital tool in system design which provide visual representation of user interaction with system. It acts as a blueprint for understanding the functional requirement of a system from an end user's side (Geeks for Geeks, 2024).

The use case diagram of Online voting system is given below:

Use Case Diagram

Actors: Voter and Administrator

Use Cases: Registration, Authentication, Voting, Manage Candidates, Manage Election, Generate Reports, Error handling, etc.
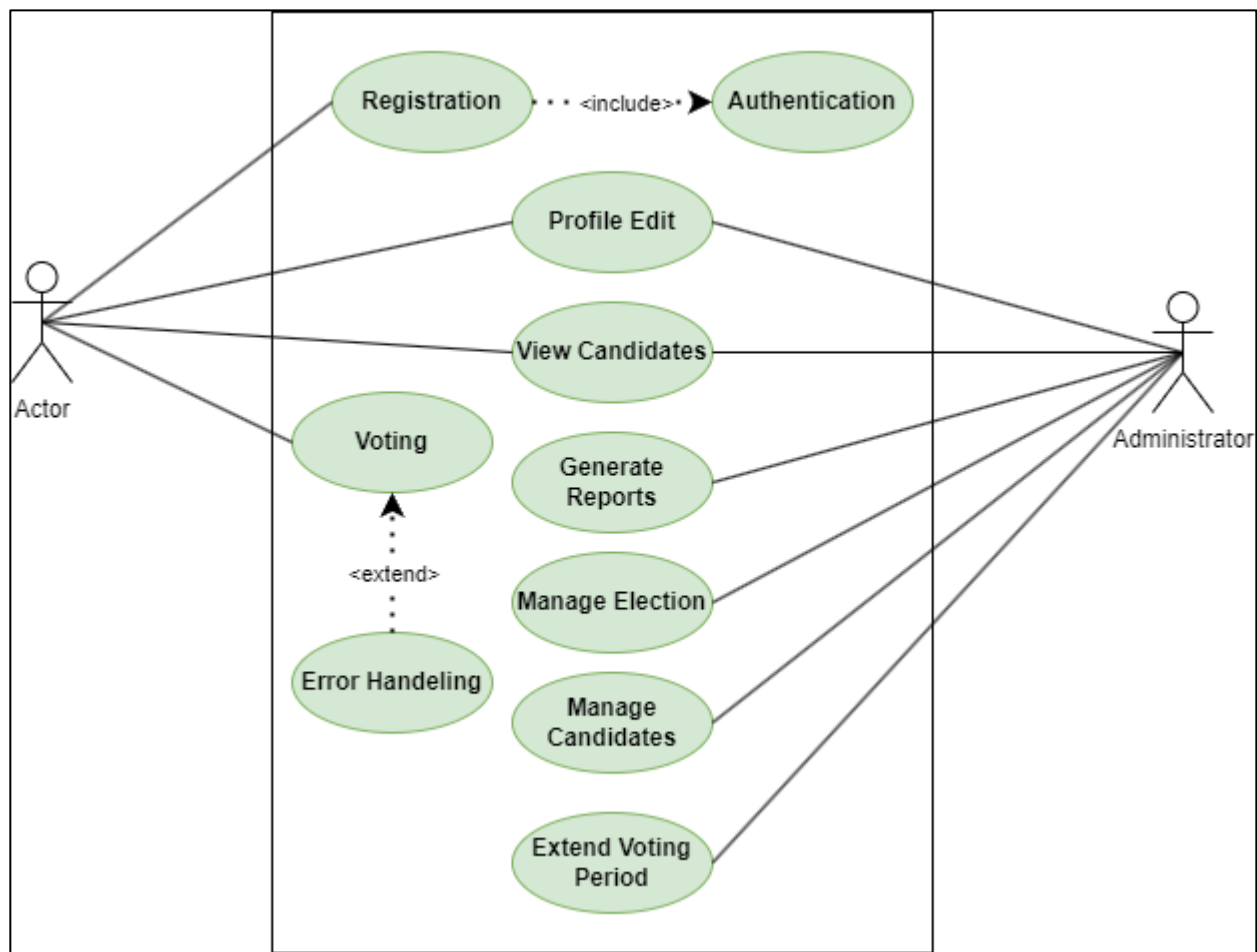
Figure 1: Use Case Diagram of Online Voting System

2.3.2. Sequence Diagram

Sequence Diagram is an interaction diagram which show the operational thing conducted in the system. It shows the interaction between objects of the system and focus on time and order of the transaction of the system (Visual Paradigm, 2024).

The sequence diagram of online voting system is show below:
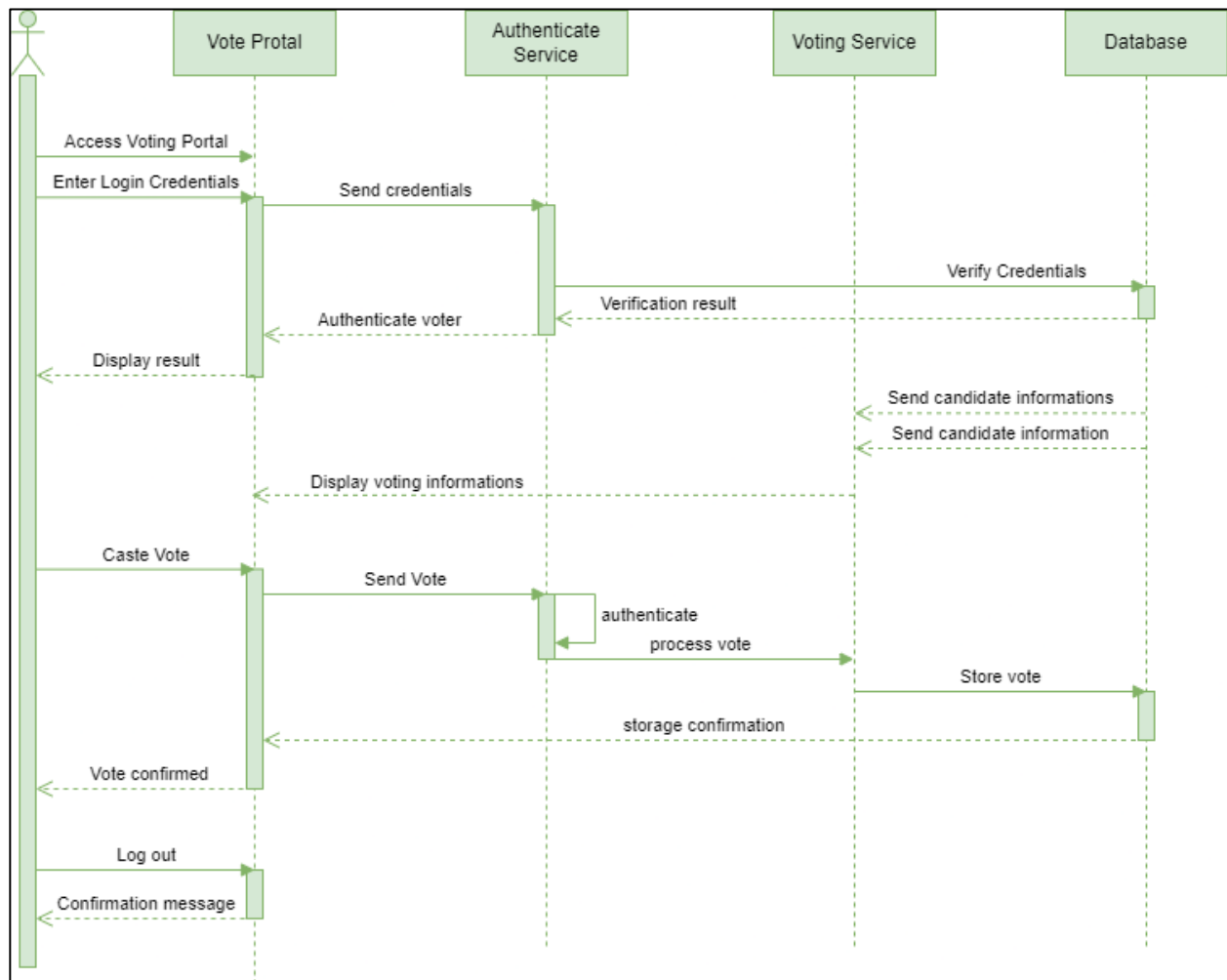
Figure 2: Sequence Diagram of Online Voting System

2.3.3. Class Diagram

Class Diagram is a type of static structure diagram that show the structure of a system by showing system's class, their attributes, methods, and the relationships among objects (Visual Paradigm, 2024).

The class diagram of online voting system is shown below:

Figure 3: Class diagram of Online Voting System

2.3.4. Activity Diagram

Activity Diagram is a behavioural diagram to describe dynamic aspect of the system which also can be looked like advanced version of flow chart that modelling the flow from one activity to another activity (Visual Paradigm, 2024).

The activity diagram of Online Voting System is shown below:

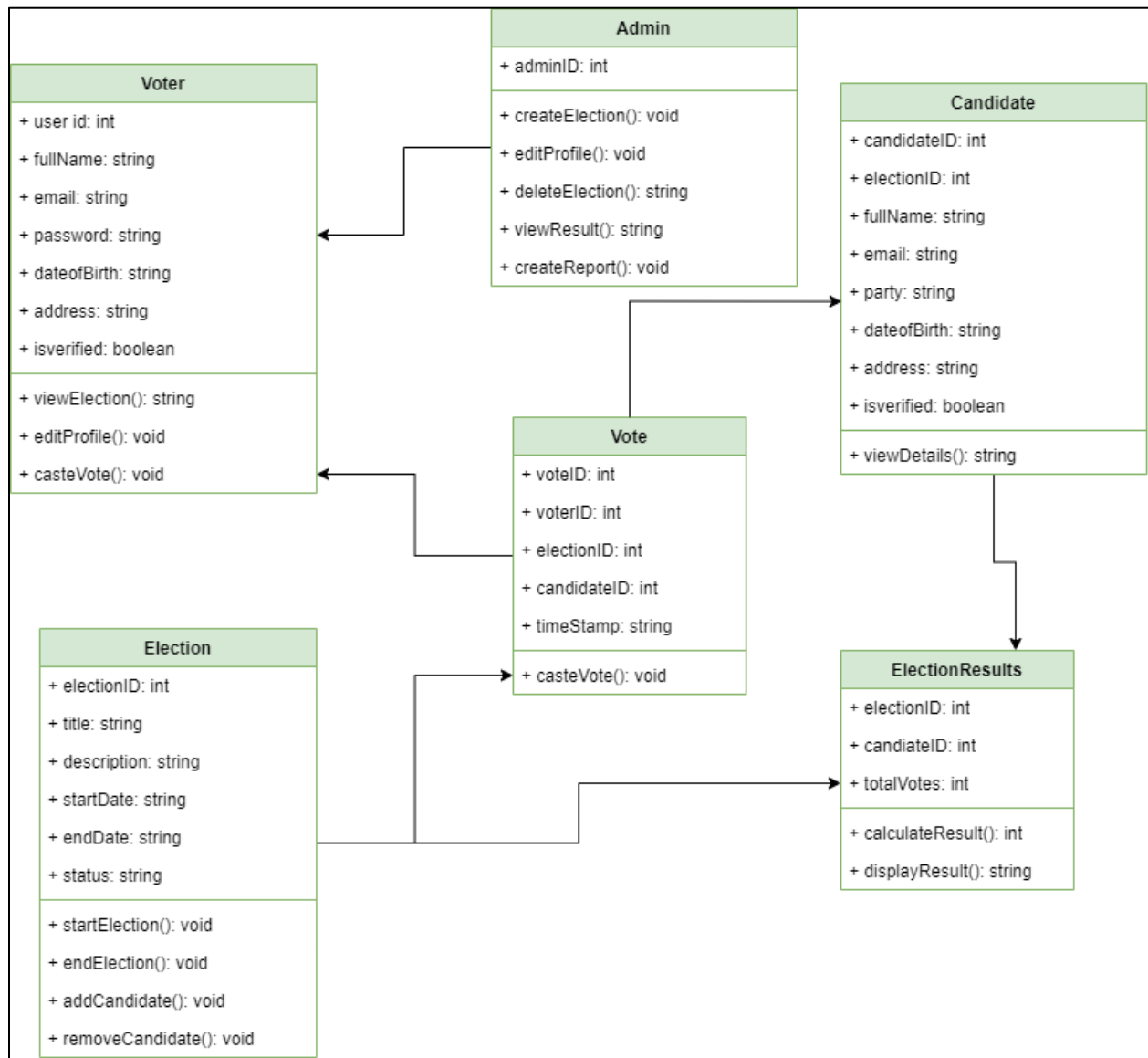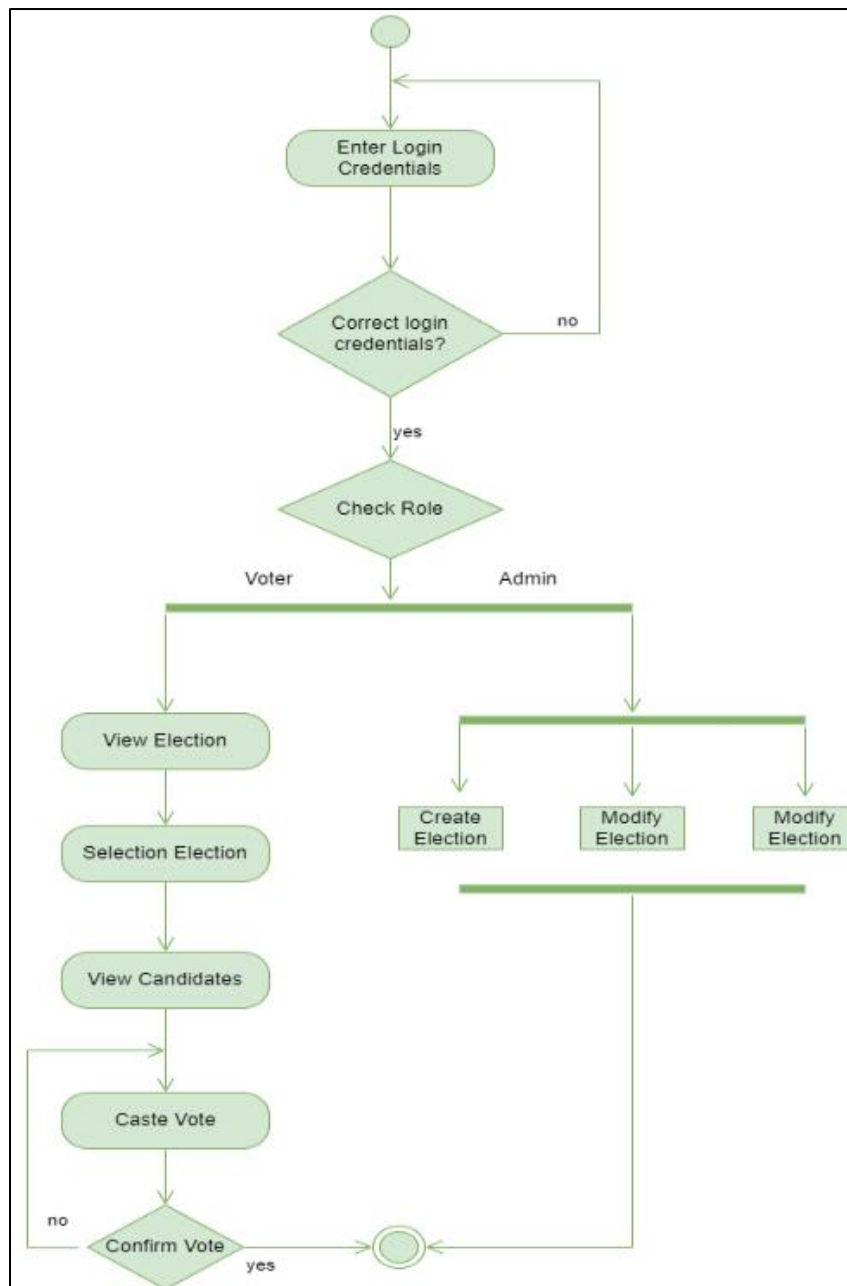*Figure 4: Activity Diagram of Online Voting System*

2.4. Justification of Technical Solution

Utilizing innovative encryption to ensure voter privacy, a scalable architecture that can withstand high traffic, user-friendly interfaces that encourage voter engagement, and robust authentication mechanisms prevent fraud during elections with an online voting system - proposed (so the choice is better than other digital alternatives as well as the traditional ones).

Overall, the proposed online voting system not only meets but exceeds the project requirements, offering a dependable, secure, and efficient solution that aligns with contemporary needs and technological advancements.

3. Development and Project Management Plan

3.1. Project Methodology

Software development methodology is defined as a framework for developing information systems, focusing on planning and organization. It benefits both teams and customers by improving efficiency and adaptability to changes (Badkar, 2023).

For the development of Online Voting System, Agile Methodology will be adopted. Choosing Agile methodology for the development of an online voting system is advantageous due to its iterative and flexible nature, which ensures the delivery of a high-quality, user-centric product. Agile allows for continuous feedback and collaboration with stakeholders, enabling the team to quickly adapt to changes and address any emerging security concerns or usability issues. This iterative approach promotes incremental improvements, ensuring that each development cycle enhances the system's functionality, reliability, and user experience. Additionally, Agile's emphasis on regular testing and validation helps identify and mitigate risks early, ensuring that the final product meets the stringent requirements of security, scalability, and accessibility essential for a robust online voting system (Laoyan, 2024).

.

3.2. Programming Languages and Environments

For Online Voting System, different programming language and environment has been used in the process. They are listed below in categorised format:

- **Frontend**: React.js, HTML5, CSS3
- **Backend**: Node.js, Express.js
- **Database**: MySQL
- **Security**: OpenSSL, JWT (JSON Web Tokens)
- **Testing**: Jest, Selenium
- **Deployment**: AWS, Docker, Kubernetes
- **Project management**: Jira, Git
- **Text Editor**: VS CODE

3.3. Required Resources

Developing a comprehensive and secure Online Voting System requires a variety of resources to ensure the project is executed successfully. These resources include development tools, server infrastructure, security tools, and testing software. Each resource plays a crucial role in distinct stages of the project, from initial development to deployment and maintenance.

- Development Tool: VS Code, GitHub, Docker
- Servers: AWS EC2 Instances
- Security Tools: SSL Certificates, Firewall Services
- Testing Tools: Automated Testing Software

3.4. Development Team Composition and Roles

A well-structured development team is essential for the successful execution of the project. The team will comprise members with distinct roles and responsibilities to ensure that each aspect of the project is managed by experts in the field. The team composition includes project managers, developers, QA engineers, and security experts.

- **Project Manager**: Oversees project progress and coordinates between stakeholders.
- **Frontend Developers**: Design and implement the user interfaces.
- **Backend Developers**: Develop server-side logic and database interactions.
- **Security Experts**: Implement and monitor security protocols.
- **QA Engineers**: Conduct testing and ensure software quality.
- **Technical Support**: Provide post-deployment support and maintenance.

3.5. Project Effort/ Cost Estimation

Estimating the effort and cost for the project is crucial to ensure that we stay within budget and deliver on time. We will use the Function Point Analysis technique to estimate the effort required. This technique helps in quantifying the functionalities provided to the user based on the requirements specification.

Using this approach, we estimate the project to require approximately 2000 person-hours. This includes time for design, development, testing, and deployment. The detailed breakdown of the effort estimation helps in resource planning and setting realistic timelines.

4. Preliminary Execution Schedule

A preliminary execution schedule outlines the timeline for the project, indicating when each phase will start and end. This schedule helps in tracking progress and ensuring that the project stays on track. We will use a Gantt chart to visually represent the schedule, providing a clear picture of the project timeline.

4.1. Major Components in Proposed Solution

Key components of the online voting system include the Voter Registration System, Voting Interface, Back-End Infrastructure, Security Measures, and Results Reporting System. Each component is vital for ensuring secure, efficient, and user-friendly operations. They are described below.

- Voter Registration: Develop a secure, compliant registration platform.
- Voting Interface: Create a user-friendly, secure voting interface.
- Back-End: Build robust infrastructure for high traffic and data integrity.
- Security: Implement encryption and authentication to prevent threats.
- Results Reporting: Develop real-time, transparent election results system.

4.2. Component Dependency Relationships

Component dependencies are crucial for project planning. For example, the Voting Interface relies on the Back-End Infrastructure and Security Measures, while the Results Reporting System depends on data from the Back end. Identifying these relationships ensures efficient task sequencing and resource allocation. They are described below:

- Voter Registration: Depends on secure database and authentication integration.
- Voting Interface: Requires integration with back-end and security systems.
- Back-End: Needs scalability, security, and integration with voting and reporting systems.
- Security: Must integrate with voter registration and voting interfaces.
- Results Reporting: Relies on real-time data and back-end integration.

4.3. Gantt/Pert Chart

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration, and end date of the activity. This allows you to see briefly (Gantt.com, 2024).
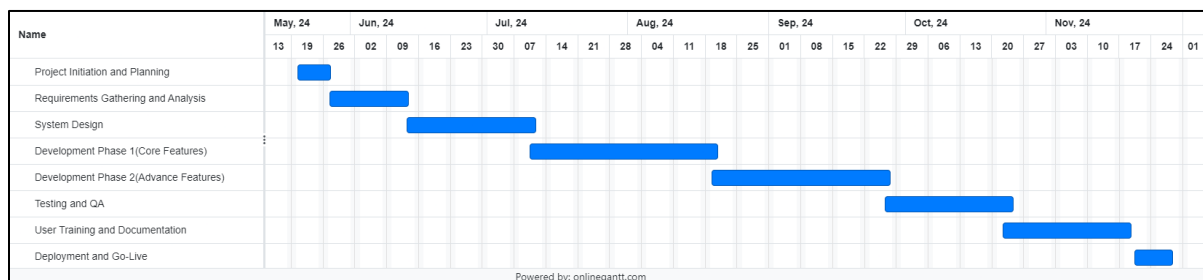
Gant chart for online voting system is show below:

Figure 5. Gantt Chart for Online Voting System

5. Risk Analysis and Mitigation Strategy

Every project faces potential risks that could impact its success. Identifying these risks early and planning mitigation strategies is essential. Below is an analysis of potential risks and the strategies we will employ to mitigate them:

1. Risk 1: **Security Breach**

   Mitigation: Implement multi-layered security, regular audits, and monitoring.

2. Risk 2: **System Downtime**

   Mitigation: Use redundant systems and regular backups.

3. Risk 3**: User Accessibility Issues**

   Mitigation: Conduct extensive user testing and incorporate feedback.

4. Risk 4: **Data loss**

   Mitigation: Implement regular methodology to adapt to changing requirements.

6. Additional Sections

6.1. Testing

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of good testing include preventing bugs and improving performance. Software testing today is most effective when it is continuous, indicating that testing is started during the design, continues as the software is built out, and even occurs when deployed into production (IBM, 2024).Online Voting system will be tested with all kind of testing like unit testing, system testing, security testing, performance testing, Usability Testing etc.

6.2. Deployment and Maintenance

The deployment will be phased to achieve stability every step and to address any related issues instantly. At first, a small user group will be chosen for beta testing of this system before releasing it on AWS entirely. Continuous supervision alongside frequent improvements will be there to make sure it performs best regarding both security measures and performance concerns, among others. IT remains dependable through maintenance by a separate technical team; this team is tasked with fixing bugs as well as providing any necessary security patches among other performance improvements.

7. Conclusion

Online Voting System aims to provide a secure, easy-to-use voting platform. Our main goal is to comply with the needs of the customer, to ensure confidence in the electoral process, through elaborate planning, advanced security measures and a stable infrastructure. Owing to the strategy of well-organized development and evaluating our solution guarantees data consistency, correct outcomes, and electorate trust.

8. References

Badkar, A. (2023, August 30). *Software Development Methodologies: Everything You Need to Know*. Retrieved from https://www.simplilearn.com/software-development-methodologies-article

Gantt.com. (2024). *What is Gantt Chart*. Retrieved from https://www.gantt.com/

Geeks for Geeks. (2024, February 9). *Use Case Diagrams | Unified Modeling Language (UML)*. Retrieved from https://www.geeksforgeeks.org/use-case-diagram/

Geeks for Geeks. (2024). *What is Stakeholder? Defination, Types and Examples*. Retrieved from https://www.geeksforgeeks.org/stakeholder-software-engineering/

IBM. (2024). *What is software testing?* Retrieved from https://www.ibm.com/topics/software-testing

Laoyan, S. (2024, February 2). *What is Agile methodology? (A beginner's guide)*. Retrieved from https://asana.com/resources/agile-methodology

Visual Paradigm. (2024). *What is Activity Diagram?* Retrieved from https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/

Visual Paradigm. (2024). *What is Class Diagram?* Retrieved from https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/

Visual Paradigm. (2024). *What is Sequence Diagram?* Retrieved from https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/