

Group 17

Software Engineering

CS3022

Final Project Report

Prepared by

K.Kokulan 120314K

K.Linganesan 120337H

N.Nirojan 120428M

A.Sajeevan 120552L

M.V.Vithulan 120678D

Content

Introduction	2
Project management	3-7
Requirement document	8-11
Architecture Document	12-17
System model	18-19
Software test specification	20-25

Description of the Project

Main Objective of the project - To create a highly securable web based system for the "Election Results Management System (ERMS)" to achieve the process of count of ballots transfer information from poll stations to Elections Secretariat.

Brief Project Description

This ERM system enables to submit and manage the results from the polling stations. Without transfer the ballot boxes to counting station from polling station this system helps to handles the count of votes in the polling stations by a securable main server. The officer in charge of the station sends the result via a private web interface and sends a message via mobile phone. If one this communication mechanism can't be possible then the polling station officers expected to take a signed results sheet to a designated Centre for the district. Then the staff of that Centre has to enter the results through a separate web-interface developed for them. This system also allows *viewing* the accumulated results for a selected group of officials at the Secretariat in real time and publishing the results in the web site of the Department of Elections and notifying a selected set of media organizations through e-mail.

Project Plan

Task Name	Duration	Start	Finish
Inception Iteration			
Establishing business case	1 day	15-Jan-2015	15-Jan-2015
Establish the project scope and boundary conditions	1 day	15-Jan-2015	15-Jan-2015
Outline the use cases and key requirements that will drive the design trade-offs	5 days	16-Jan-2015	20-Jan-2015
Identify risks	3 days	21-Jan-2015	23-Jan-2015
Prepare a preliminary project schedule and cost estimate	1 day	23-Jan-2015	23-Jan-2015
<i>Deliverable: An established business case at the end of Inception phase.</i>			
Elaboration Iteration			
Capture healthy system requirement	5 days	24-Jan-2015	28-Jan-2015
Analyse risk factors	2 days	29-Jan-2015	30-Jan-2015
Establish and validate system architecture	4 days	31-Jan-2015	03-Feb-2015
Plan for construction phase	1 day	04-Feb-2015	04-Feb-2015
<i>Deliverable: A well-defined requirements model of the system including UML use-cases, an architectural description, and a development plan at the end of Elaboration phase.</i>			
Construction Iteration			
Develop and tune user interface	12 days	05-Feb-2015	16-Feb-2015
Implement database	5 days	17-Feb-2015	21-Feb-2015
Implement web-services and other functions	12 days	21-Feb-2015	04-Mar-2015
Testing	3 days	05-Mar-2015	07-Mar-2015
<i>Deliverable: A working software system and documentations at the end of Construction phase.</i>			
Transition Iteration			
System deployment	1 day	08-Mar-2015	08-Mar-2015
Getting feedbacks	7 day	09-Mar-2015	16-Mar-2015
System conversion	4 days	17-Mar-2015	20-Mar-2015
User training	7 days	21-Mar-2015	27-Mar-2015
<i>Deliverable: A documented software system that is working correctly in its operational environment and training to the staff at the end of Transition phase.</i>			

Resource Allocation Table

Task Name	Responsibility	Duration (Planned)	Duration (Actual)
Establishing business case	Linganesan	1 day	1 day
Establish the project scope and boundary conditions	Linganesan	1 day	1 day
Outline the use cases and key requirements that will drive the design trade-offs	Sajeevan, Vithulan	5 days	4 days
Identify risks	Kokulan, Nirojan	3 days	3 days
Prepare a preliminary project schedule and cost estimate	Kokulan, Nirojan	1 day	1 day
Capture healthy system requirement	Sajeevan, Vithulan	5 days	5 days
Analyse risk factors	Nirojan	2 days	2 days
Establish and validate system architecture	Kokulan, Linganesan	4 days	4 days
Plan for construction phase	Kokulan	1 day	1 day
Develop and tune user interface	Kokulan, Nirojan	12 days	10 days
Implement database	Linganesan	5 days	5 days
Implement web-services and other functions	Sajeevan, Vithulan	12 days	11 days
Testing	All members	3 days	4 days
System deployment	All members	1 day	1 day
Getting feedbacks	All members	7 day	7 days
System conversion	All members	4 days	4 days
User training	All members	7 days	7 days

Minutes of group meetings

Meeting 1

Date: 15th January 2015, 11.20 a.m. – 12.20 p.m.

Venue: CSE Tutorial Room

Objective: Initiate the Project work

Minutes of the meeting: All of the members were present at the meeting. Group leader initiate the meeting by explaining the problem. Then the draft plan for inception phase is constructed and each member has been given specific tasks for the inception iteration. Scheduled next meet up to be at 11.20 a.m. on 22nd January 2015.

Outcomes: Divided and created a temporary schedule of the work to be done during Inception phase.

Meeting 2

Date: 22nd January 2015, 11.20 a.m. – 12.20 p.m.

Venue: CSE Advance Lab

Objective: Understand the problem and analyse the feasibility of the project

Minutes of the meeting: Everyone came up finishing the inception phase. We have discussed the problem in depth to finalise the business case, project scope and boundary conditions, and outline of use cases and key requirements while identifying the risks. After the discussion on feasibility of the project and cost estimates that has been prepared, we decided to continue with the project. We refined the preliminary project plan to continue with elaboration phase. Next meet up has been scheduled at 11.20 a.m. on 5th February 2015.

Outcomes: Established the business case and preliminary project plan.

Meeting 3

Date: 5th February 2015, 11.20 a.m. – 12.20 p.m.

Venue: CSE Advance Lab

Objective: Finalise requirements model of the system

Minutes of the meeting: All met after finishing the elaboration phase. We discussed the established system requirements to clarify the ambiguities in the project. Then we discussed the risk factors analysed to ensure that the project is feasible to continue. Then we refined the system architecture that has been developed during elaboration phase. We decided to develop the system logic in Java Servlet and MySQL for database. Finally, before winding up the meeting, we refined the project plan for construction phase. Next meet up has been scheduled at 11.20 a.m. on 5th March 2015.

Outcomes: Finalised requirements model of the system including UML use-cases, an architectural description, analysis of risk factors, and a development plan especially for the construction phase

Meeting 4

Date: 5th March 2015, 11.20 a.m. – 12.20 p.m.

Venue: University Library

Objective: Finalise the system developed and relevant documentations

Minutes of the meeting: This meet up mainly scheduled to test the system developed during construction phase and correct any shortcomings. All the members were satisfied with the user interfaces developed and the database design. All members came up with sample test cases to test the functionality and reliability of the system. System was tested with variety of data including erroneous data to ensure the reliability of the system. Except some minor logical issues, all the functions worked without any problem. Those logical errors were corrected then and there and then we decided to use testing tools to ensure all the test cases are passed. Earlier, while developing the system, we used JUnit for unit testing and now decided to use JWebUnit and Apache Tomcat Server to test the system including the web services. All the members have instructed to conduct the testing mentioned above and given the refined plan for transition phase as well. Finally, we refined the documentations prepared altering the changes made during the first phase of testing. Next meet up has been scheduled at 11.20 a.m. on 20th March 2015.

Outcomes: A working software system, relevant documentations, plan for testing, and refined project plan for transition phase.

Meeting 5

Date: 20th March 2015, 11.20 a.m. – 12.20 p.m.

Venue: CSE Tutorial Room

Objective: Analyse the issues arisen during system deployment and received customer (staff) feedbacks, construct a plan for system conversion, and schedule user training.

Minutes of the meeting: After the testing of system by developer as well as the client, this meeting was scheduled to finalise and release the system to use in real environment. In this meeting, the issue arisen during system deployment was discussed. we planned to conduct a detailed training to staffs to reduce such issues. Then we planned for system conversion and prepared a schedule for user training. In conclusion, leader announced this meeting as the final meeting and if there any other issues arise later, a meeting will be scheduled and informed to all members.

Outcomes: Refined plan for system conversion and final user training.

Requirements Document

1. Introduction

Main Objective of the project - To create a highly securable web based system for the "Election Results Management System (ERMS)" to achieve the process of count of ballots transfer information from poll stations to Elections Secretariat.

2. Actors

Staff



Staff counts the votes and check it

Officer at polling centre



Submit the result and send message

Officer of Secretariat



- 1. Confirm the message**
- 2. Release the result**
- 3. Send email notification to authorized result publishing centre**

3. Specifications of the use cases

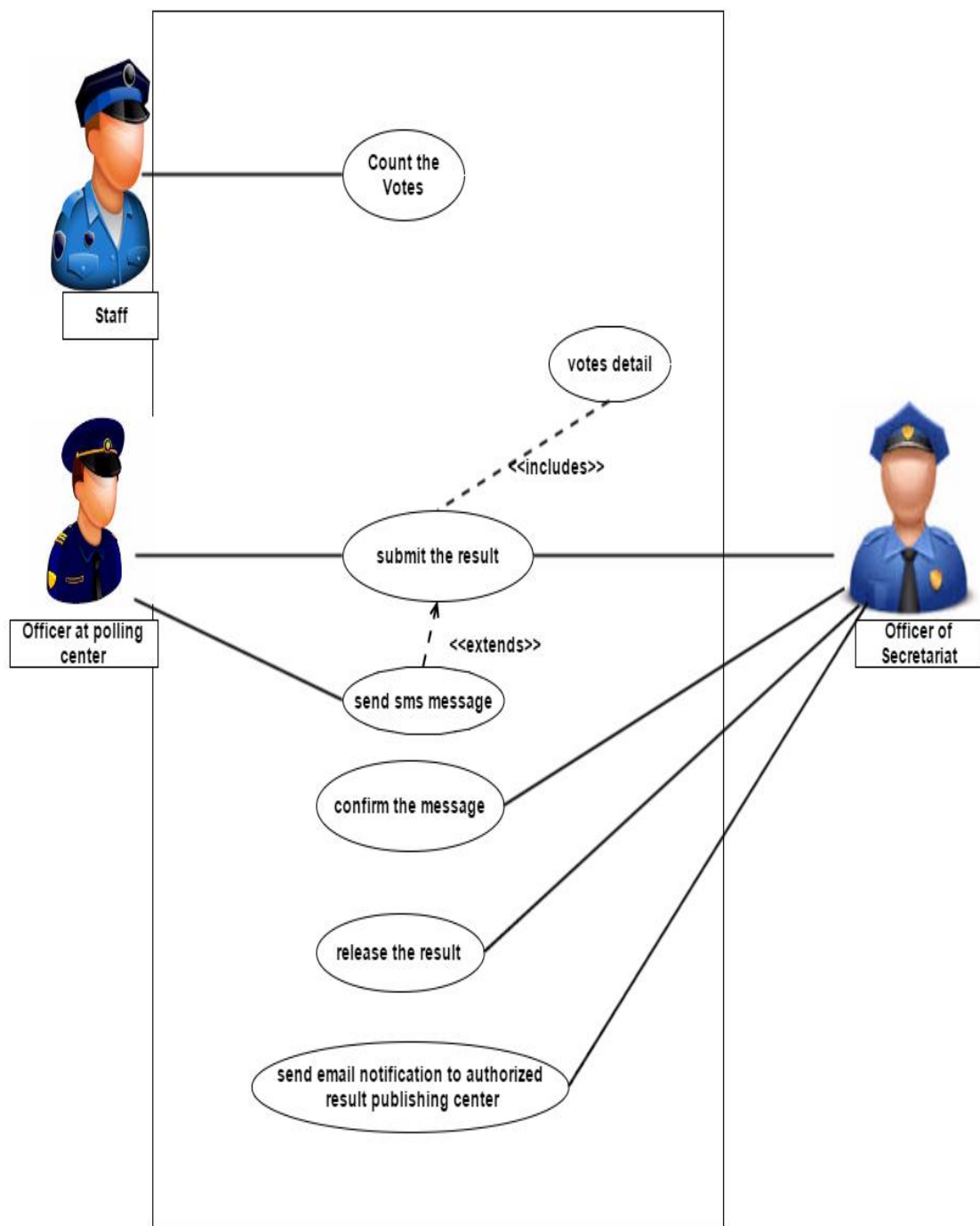
Use case name: Count the votes
Actors: The officer in charge of the station, Staffs
Count the ballots in the polling stations.

Use case name: Submit the result
Actors: The officer in charge of the station
Enter the number of votes on the securable web interface to ERMS.

Use case name: Send SMS messages
Actors: The officer in charge of the station
Send a set of SMS messages to the system indicating the results.

Use case name: Confirm the messages
Actors: The officer in charge of Secretariat
When the results are received by the Secretariat, they have to be confirmed by an officer there before being accumulated.

Use case name: Release the result
Actors: The officer in charge of Secretariat
Once counting has been completed for a particular electoral district, they can officially release the results through ERMS.



4. Non Functional Requirements

Essentially, Non-functional requirements define the overall qualities or attributes of the resulting system. The key non-functional requirements that we consider here are Security, Usability, Reliability, Technical and Performance.

In the case of security, the system should be provided with the secure user login using strong user identification and authentication capability. Because of the system is web based application, the chances of data hacking and virus attacking can be occurred significantly. As the system runs on web server, unauthorized access of database and data corruption must be avoided to run fully worked software. So we should provide best security protections such as use cookies securely, create safe error messages, back up often and keep the backup physically secure and run application with least privileges. Further, accessing level of the system should be restricted according to the priority level of the members who access the system. For example one polling centre officer cannot see other polling centre details.

As next if we consider the usability, the user interface should be provided as familiarizing with the interface with in a short period by users and while the system hand over to the election board, proper documentation and user manual should be delivered in order to use the system effectively. Consequently in the case of reliability, the system should perform its required functions under stated conditions for a specific period of time. For example if the transcript requirement is more than one per year then the system should query the payment details from the user who access the system. As a second example, if all the requirements are submitted then the system must automatically update the information of a relevant user as well as do the printing task without any service outages and incorrect processing.

In the case of technical requirement, the new system has to be implemented along with the existing polling management system. If there are any untested configurations in the new system that can affect the existing system as well. Therefore it should be carefully avoided. The staffs may not have on hand experience in web based applications so sufficient training to the staffs or data protection mechanism is needed to ensure the data in the system are not get lost due to erroneous use by the staffs. As precautions the system should be tested properly and there should be a backup database management system to protect from data loss.

Next in the case of performance, the speed of the operation of a system reacts to a user input and how much system can accomplish within a specified amount of time is mainly considered. As the system is web based, there are lots of users' accesses the system concurrently and all the data related to the system stored in a database server therefore fast storing and retrieval mechanisms have to be considered in the database segment of the system.

Architecture Document

Introduction

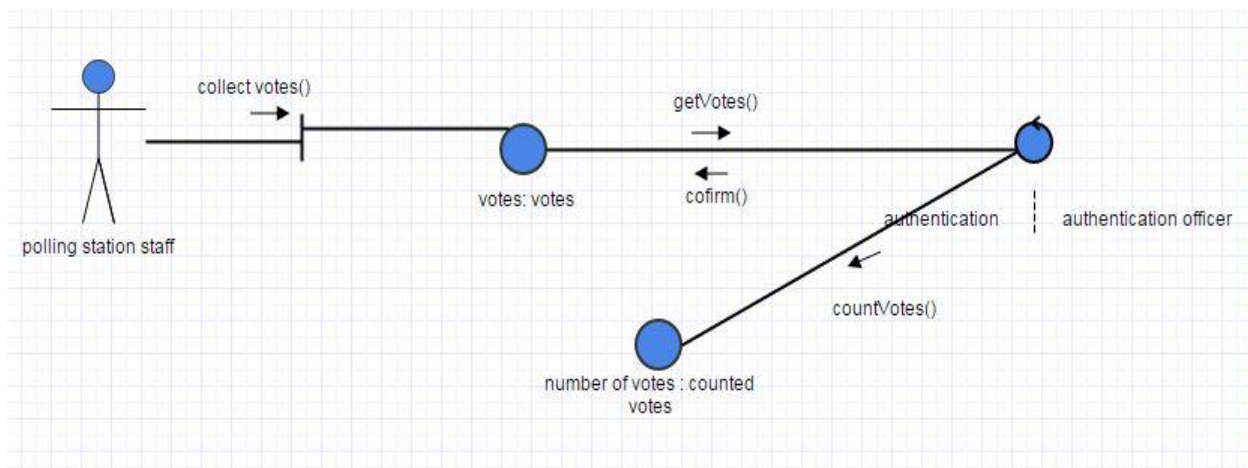
We have decided to use the 4+1 view model as our architecture model since it describes the software architecture from 4 different perspectives and as scenario.

Logical view

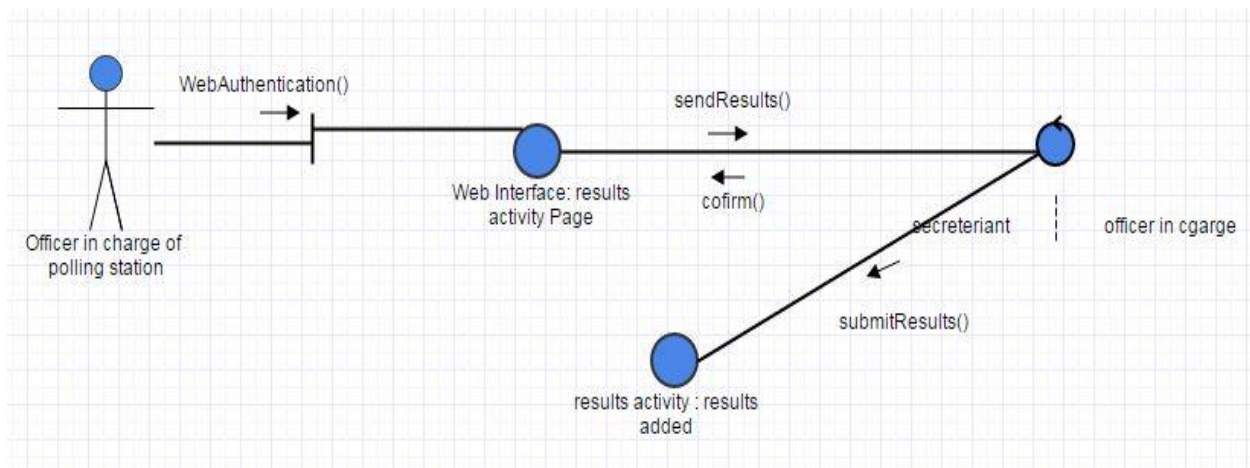
This view concerned with the functionality that the system provides to end-users. We have used class diagram, communication and sequence diagram to show this view of the architecture model. Since class diagram and the sequence diagram are part of the system model, we have provided the UML communication diagrams of some main activities.

Communication Diagrams

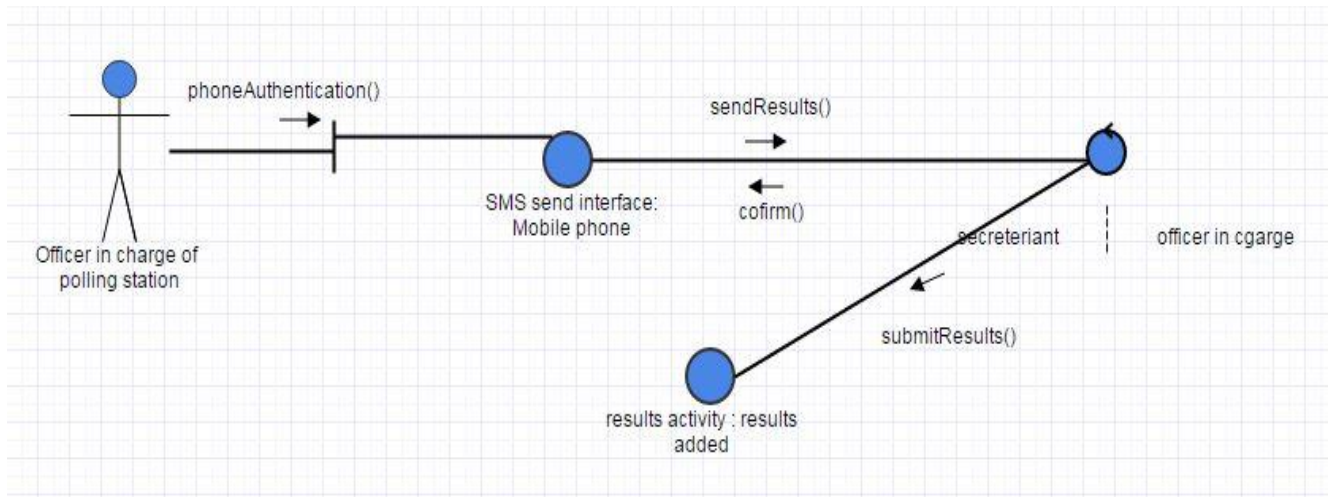
For counting the votes



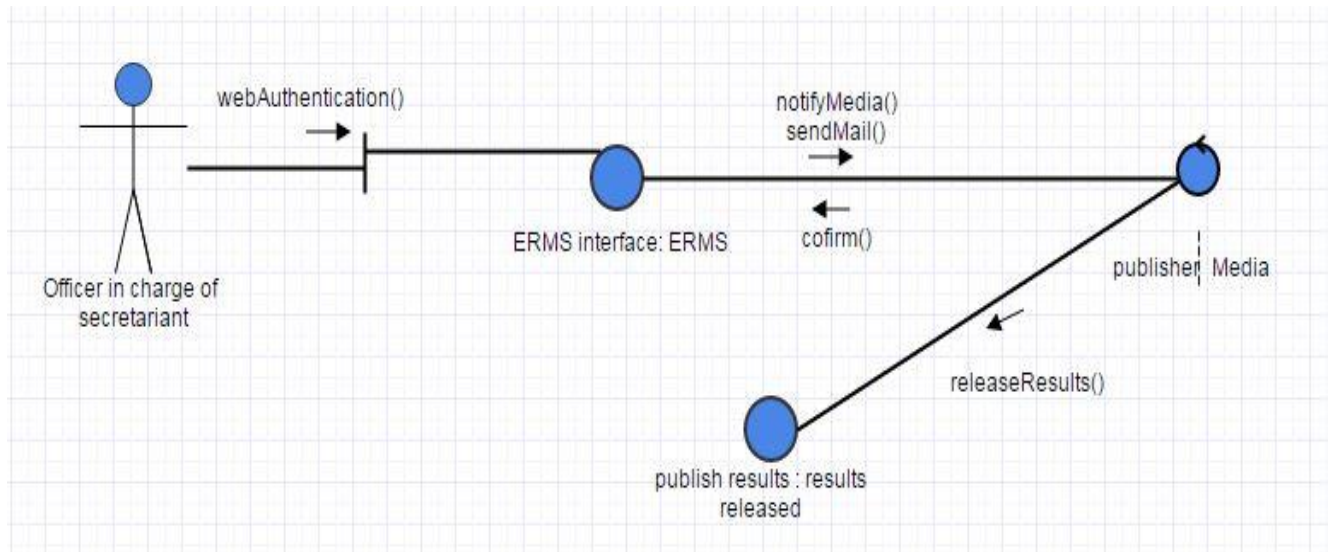
For send results



For send SMS



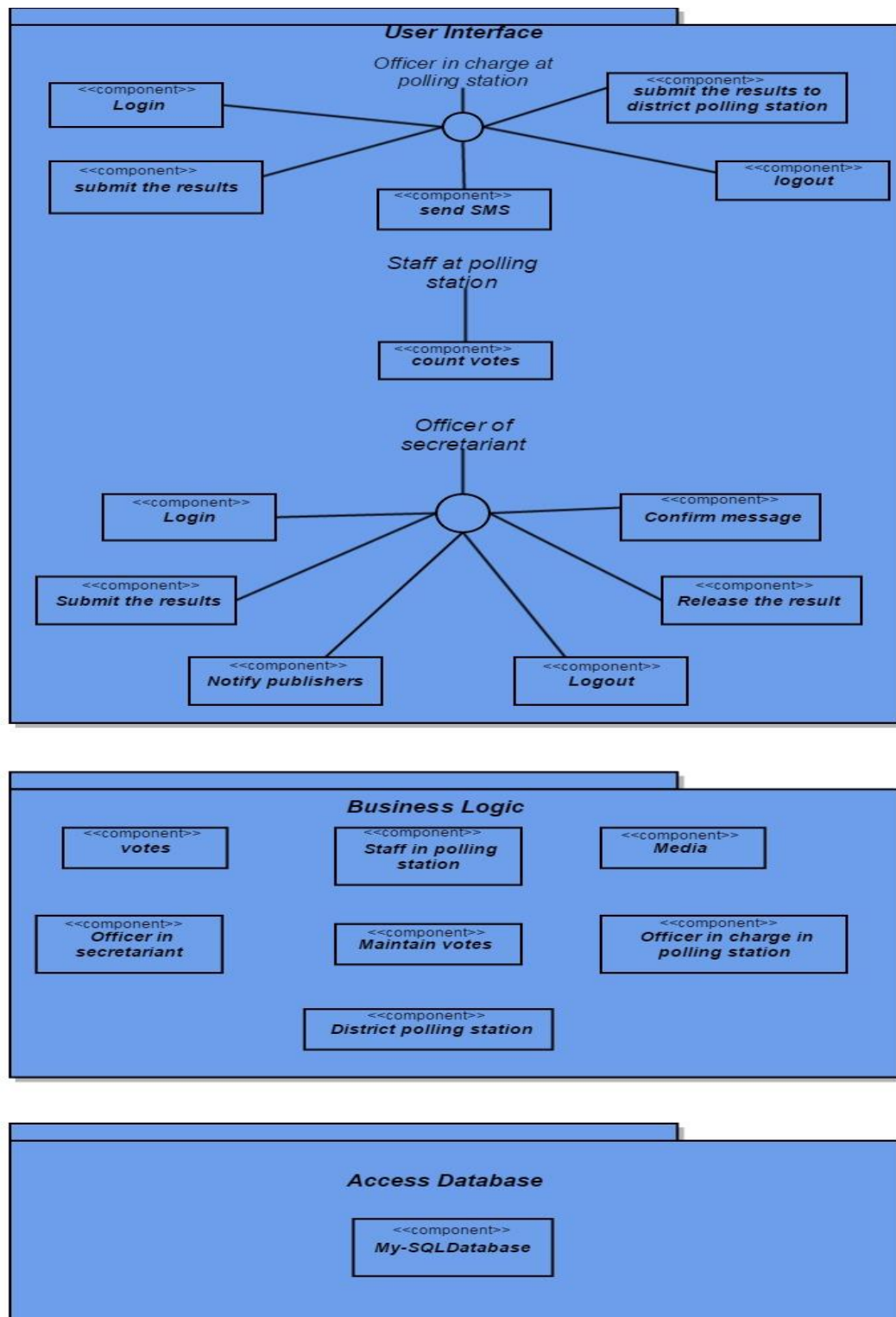
Publishing the Results



Development view

This view illustrates a system from a programmer's perspective and is primarily concerned with software management. This is also known as the implementation view. We have used UML component diagram to show this view of the architectural model.

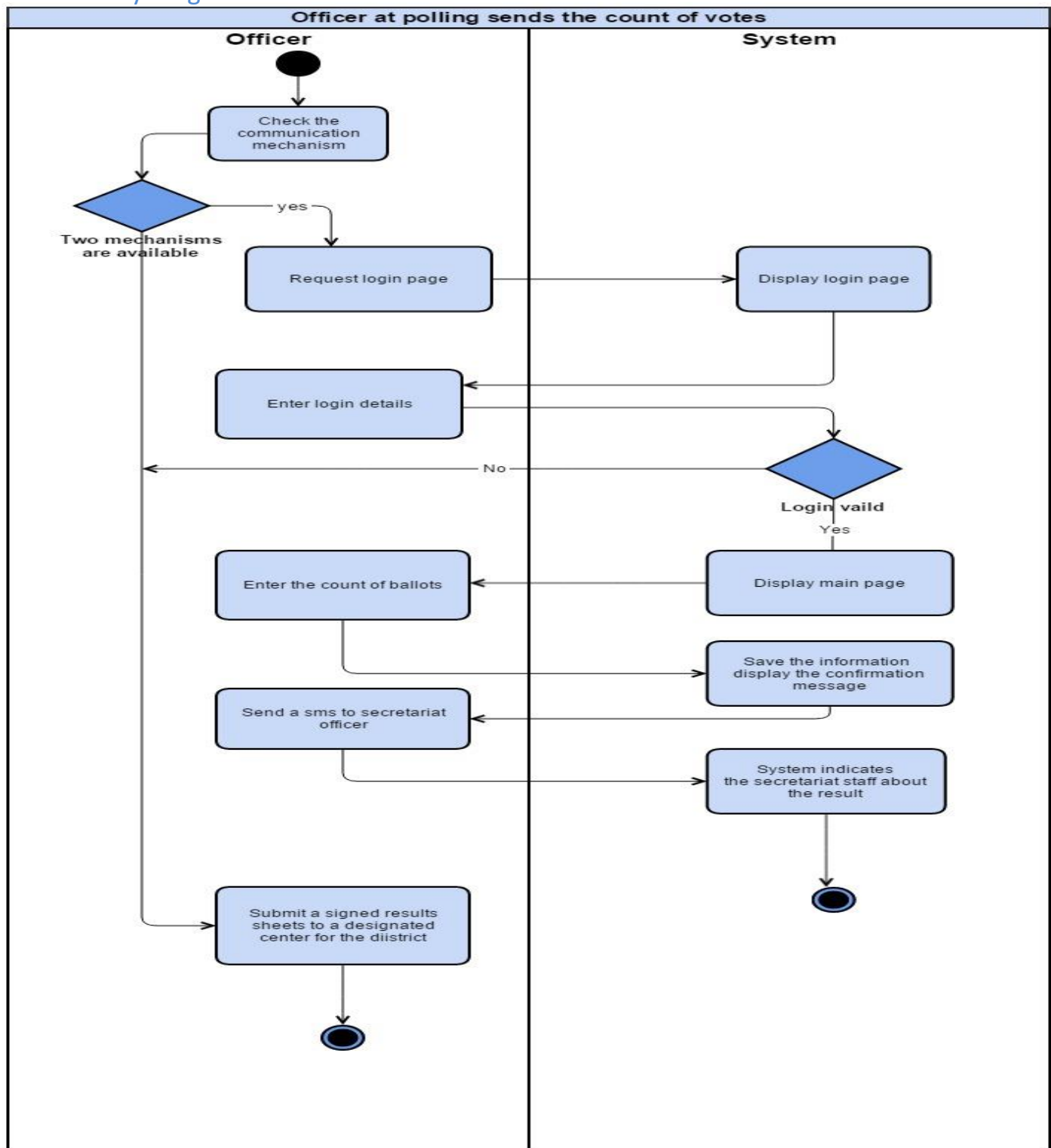
Component Diagram



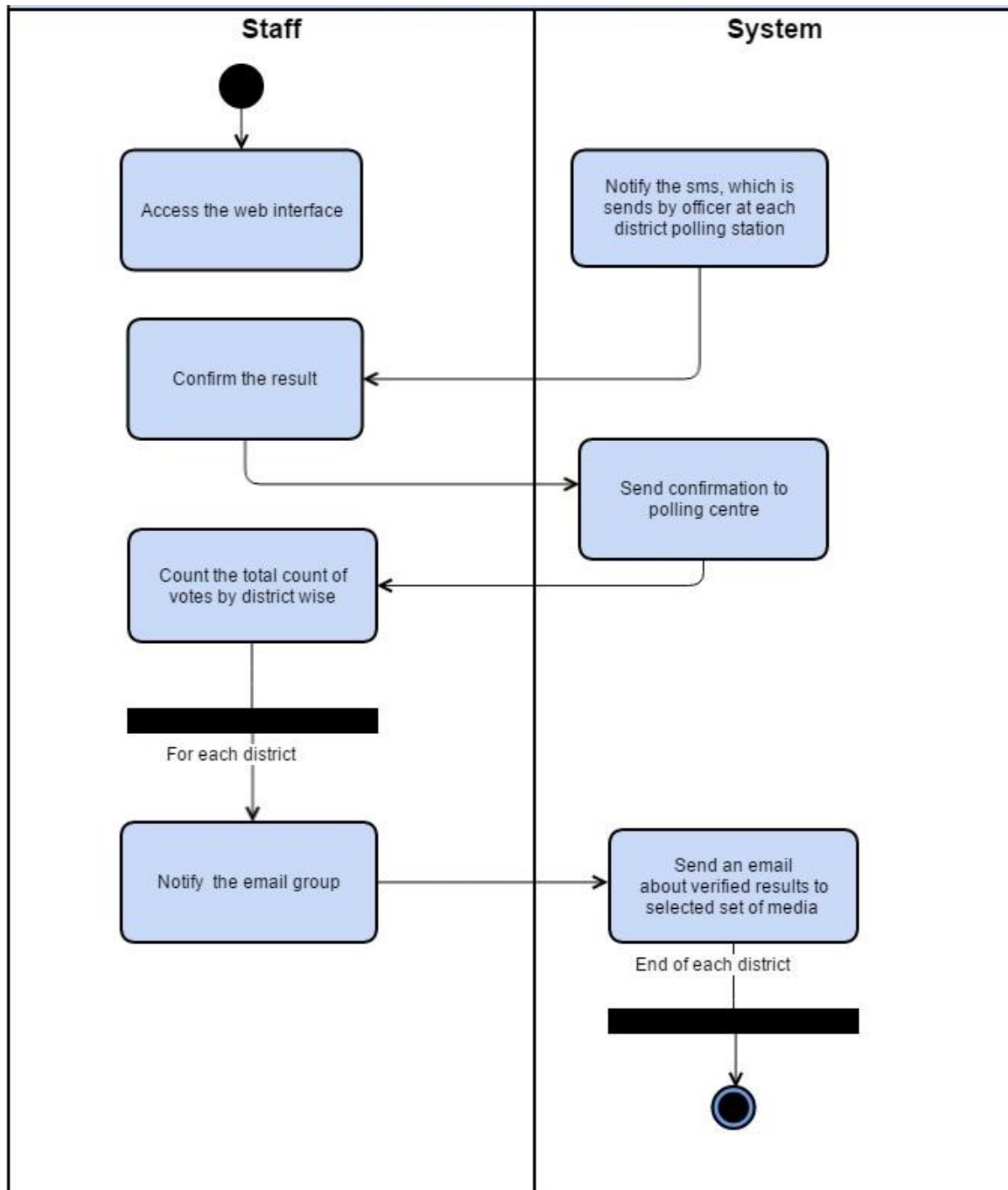
Process view

This view deals with the dynamic aspect of the system. It explains the system processes and how they communicate. It focuses on the runtime behavior of the system. We have used UML activity diagram to show this view of the architectural model.

Activity Diagrams



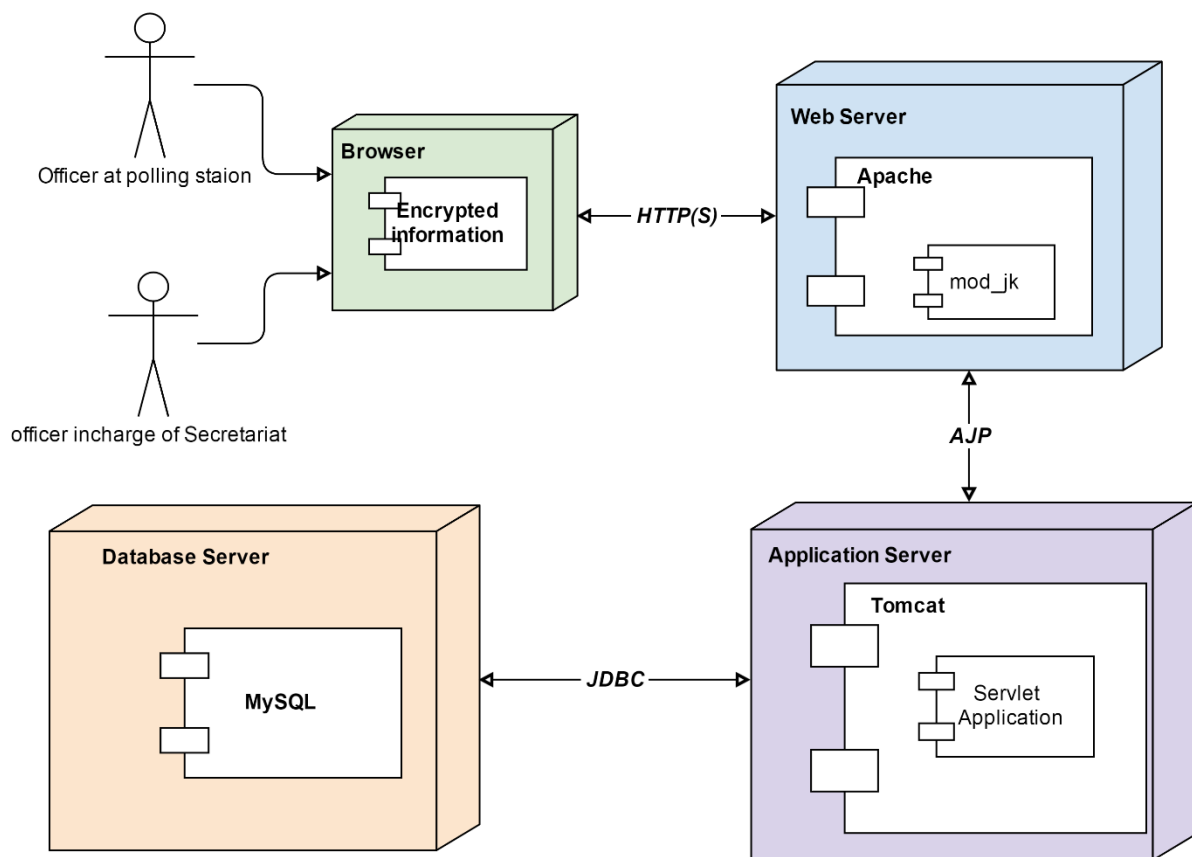
The officer in charge of Secretariat access the vote counts



Physical view

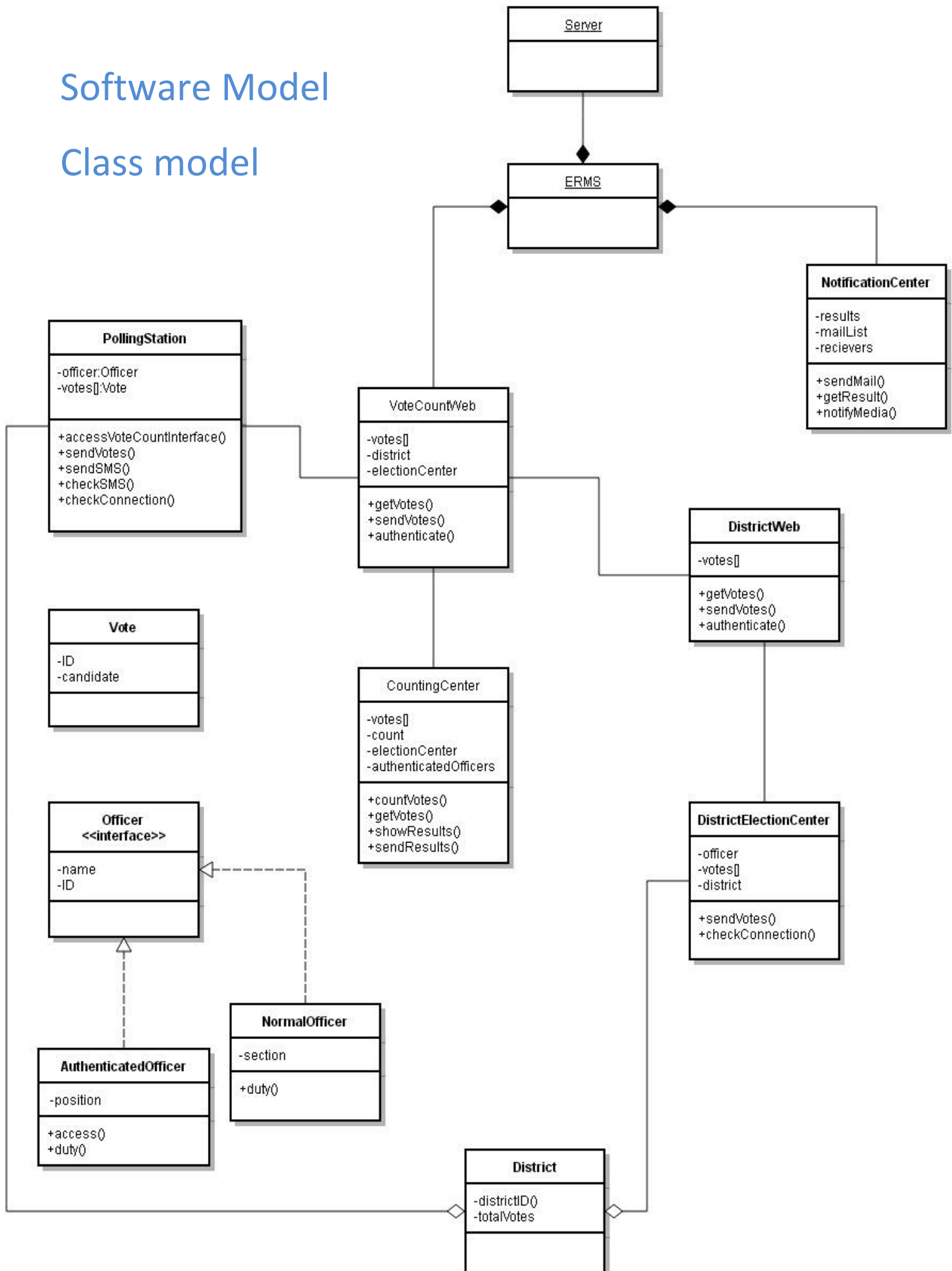
This view depicts the system from a system engineer's perspective. It is also known as the deployment view. We have used UML deployment diagram to show this view of the architectural model.

Deployment diagram

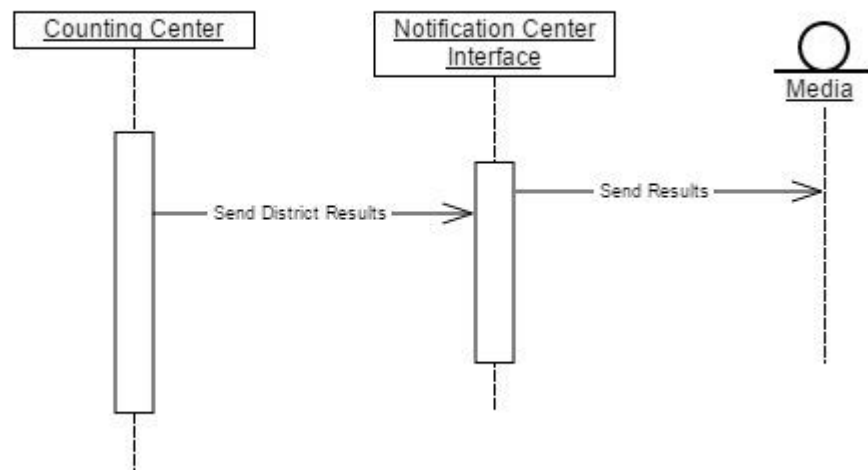
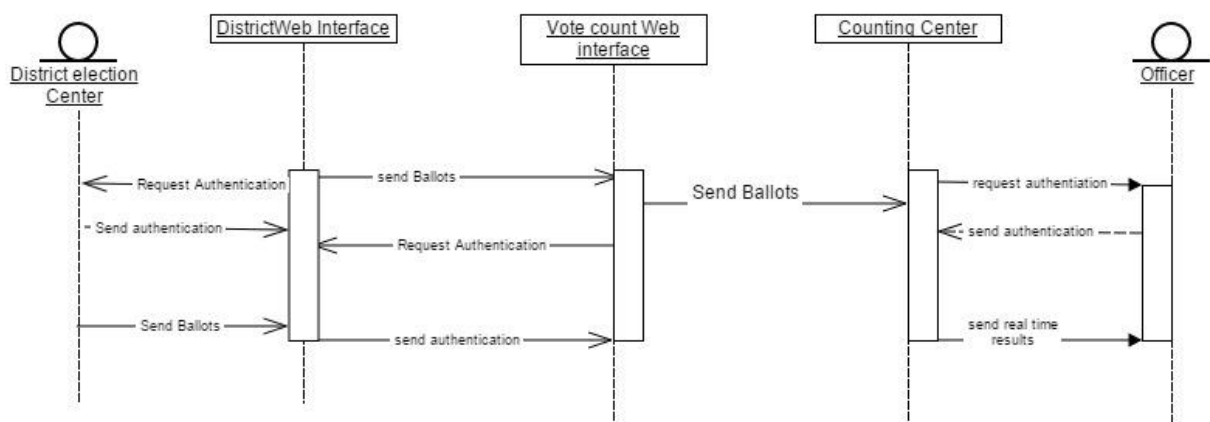
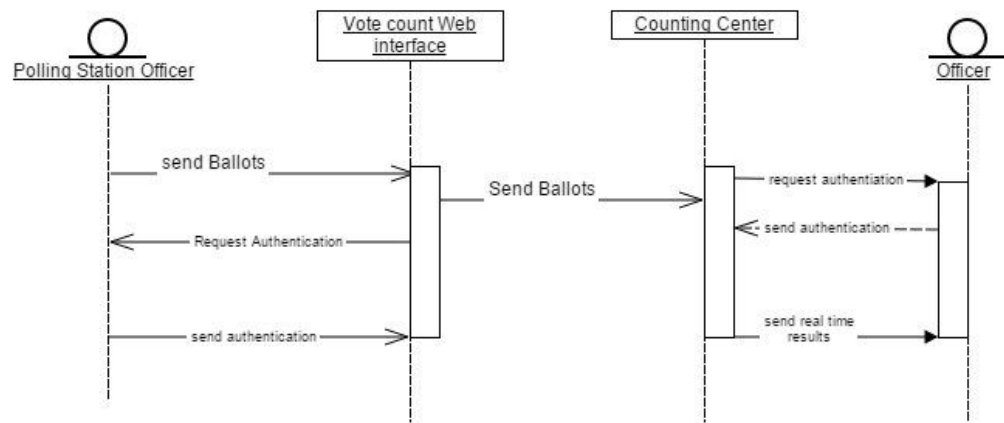


Software Model

Class model



Sequence diagrams



Software Test Specification

Introduction to Testing

The primary purpose for testing is to detect software failures so that defects may be uncovered and corrected. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do. Information derived from software testing may be used to correct the process by which software is developed.

There are various types of testing procedures that can be carried on this project. However we can basically group them as Development testing, release tests and user testing. This system had been undergone to both development testing and release testing.

Development Testing

Under the development testing the system was tested based on unit testing, component testing and system testing. Additional to the black box testing the inspections and reviews also carried on to ensure the code quality and for error detection.

Inspection

Some carried on white box testing procedures are shown below.

The source code of the program was thoroughly analyzed and fixed the errors as well as the code was refactored.

Case 1:

When an officer try to logging in to the system, system has to be checking username and password for null value and empty value. This code segments clearly ensures that null or empty username and password caused to display an error message to the user.

```
//more codes here
```

```
if (username == null || username.length() == 0) {
```

```
    isValidated = false;
```

```
    request.setAttribute(ERROR_MESSAGE,USER_NAME_REQUIRED);
```

```
} else if (password == null || password.length() == 0) {
```

```
isValidated = false;  
  
request.setAttribute(ERROR_MESSAGE,PASSWORD_REQUIRED);  
  
}  
  
//more codes here
```

Case 2:

This code fragment shows that the program handles the invalid username and password (in other words invalid login) and displays the warning to the user.

```
//more codes here  
  
else if (loginStatus.equals(INVALID_LOGIN)) {  
  
request.setAttribute(ERROR_MESSAGE,INVALID_LOGIN);  
  
requestDispatcher = request.getRequestDispatcher("/viewLocationForWelcome.action");  
  
}  
  
//more codes here
```

Unit testing

Unit testing is automated by using JUnit framework since this project is developed using Java. The automated testing decreases the time and allowed the regression testing whenever there is a modification in the source code.

Component and system testing

Component testing and system testing were done manually by our team members in an informal pair testing manner however a part of the component and system testing were automated by JWebUnit framework. Identified interface errors were fixed and regression testing was carried on after each and every modification in the components.

Release Testing

Under the release testing the security and functionality of the system was tested. This was carried out as a black box testing with specially designed test cases.

Functional Testing

Test Type	Unit Test			
Test Case Number	01			
Test Case Name	User Login			
Test Case Description	This is to test the User login of the system.			
Item(s) to be tested				
User Login				
Specifications				
Input		Expected Output/Result		
Log in to the system using the a username and a password		Validate the provided data Prompt the error messages when authentication fails		
Procedural Steps			Result	
		Pass	Fail	N/A
1	Enter the Username and Password	X		
2	Connect the central server	X		
3	Authenticate the User & Display Error message	X		
4	Create a User Session	X		
5	Display the relevant dashboard	X		

A message notified when an Invalid username or password is submitted. It provides a link to request the password in case of a loss. When the password or username field is left blank, a message is prompted notifying it.

Successful submission of username and password will redirect the officer to the relevant web based interface.

Test Type	Unit Test			
Test Case Number	02			
Test Case Name	submit the vote count			
Test Case Description	when officer enter the values of each politician ids and their vote count system has to check whether the entries are correct or not			
Item(s) to be tested				
submission of result				
Specifications				
Input		Expected Output/Result		
set of politician ids and relevant vote count		Validate the data and if the data is in correct form submit it to the central server else generate an error message for the officer.		
Procedural Steps			Result	
		Pass	Fail	N/A
1	enter politician ids and vote count	X		
2	checking for the validation of ids and vote count	X		
3	generate error messages related to the errors	X		
4	If there is no errors, accept the submit request and update the central server	X		
5	ask officer to conformation	X		

If the politician id is not in the vote list, the vote count is not a positive integer or the results of a given district is already in the server then the system have to send an error message to the officer.

Nonfunctional testing

The system was checked for better user experience, security and performance. Even though the user interface was tested and approved successfully because of the available hardware server restrictions. The performance testing did not meet the real time applications' requirement. The security is ensured by code inspection in development testing as well as in black box testing.

Tools needed for testing

Software:

JUnit

JUnit is a simple framework to write repeatable tests. It is an instance of the xUnit architecture for unit testing frameworks. In our project JUnit with Netbeans IDE was used for automated unit testing

Apache Tomcat Server

Apache Tomcat is an open source software implementation of the Java Servlet and JavaServer Pages technologies. We used the Tomcat as the software server; on top of that our software was tested

JWebUnit

JWebUnit is a Java-based testing framework for web applications. It wraps existing testing frameworks such as HtmlUnit and Selenium with a unified, simple testing interface to allow us to quickly test the correctness of our web application.