

# **Electronic Voting System (EVS)**

## **JavaEE Web application course**

### **SOSE 2023**

## **Project Manual**

Date: 2023.08.31

Prepared by: Team Charlie

Submitted to: Prof. Dr. Volker Riediger

# Table of Content

Introduction .....	2
System Setup .....	3
Deploying Application on server.....	4
Problems Faced.....	10
Domain Model .....	11
Time Spent .....	12
List of Completed and Missed Requirements.....	13
Subsystems in EVS Application .....	20
Using the Application.....	21
Glossary.....	29

## Introduction

The Electronic Voting System (EVS) supports electronic voting polls that can be primarily used by university bodies such as the examination office, student councils and for organizing event polling. Mostly it is difficult to organize the meetings in person, so the objective of the application is to minimize common meetings and organize the poll through the web.

In the context of university, the Examination Office can conduct the event based on polling from the students and professors with the use of EVS regarding events, seminars, webinars and many more. The use of EVS expedites the results announcement, potentially reducing the uncertainty associated with waiting for manual counts. Moreover, it can also prevent issues like overvoting or under voting.

While they offer efficiency, accessibility, and accuracy benefits, it is essential to address the associated challenges to ensure the integrity and striking the right balance between convenience, security, and transparency will be key to the successful implementation of electronic voting system.

The Electronic Voting System (EVS) will facilitate the casting and counting of polling.

This user manual will provide instructions on how to access the EVS and its installation on the web application server.

## System Setup

EVS is a web application, which means it runs on a server machine and responds to all the requests made by the client via the client's web browser. A windows/Unix/Linux/MacOS based system with the following software and hardware requirements is required to run EVS.

## **Client-side Software requirements**

The latest JavaScript enabled browser to access the application.

## **Server-side Software requirements**

### **Application Server (Payara)**

Application server is required to host the application and provide the necessary environment for its execution. We will use Glassfish/Payara application server, available at <https://www.payara.fish/downloads>.

### **Database Server**

EVS stores all data related to the system in a database. A MariaDB database server is required to run EVS. The latest version of the MariaDB database server is available at <https://downloads.mariadb.org/>.

### **JAVA**

Java Developer Kit (JDK) or Java Runtime Environment (JRE) platform must be installed on the server's operating system. The latest version of JDK is available at <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

## **Server-side Hardware requirements**

System with min 2 GB RAM.

## **Deploying Application on server**

For an application to run, it must be deployed on the application server like Payara to start serving the user requests. But before that we must complete a few steps to complete the process of installing the application on the server.

### **Installing MariaDB**

After downloading the MariaDB database server open the file and follow the instructions to install MariaDB server.

Create a database schema with the name “javaee\_teamcharlie” and desired username and password to access it.

## **Creating JDBC resource and JDBC Connection pools**

After installing the payara server in the desired location, start the server domain by running the following command to start the domain of your choice.

Once the server is up, navigate to the http URL://localhost:4848. Please note that the administrator port may be different. Refer to the port number displayed in the message after starting the server.

### **For JDBC Connection Pool**

Navigate to “Resources/JDBC/ JDBC Connection Pools to create a new JDBC connection pool.

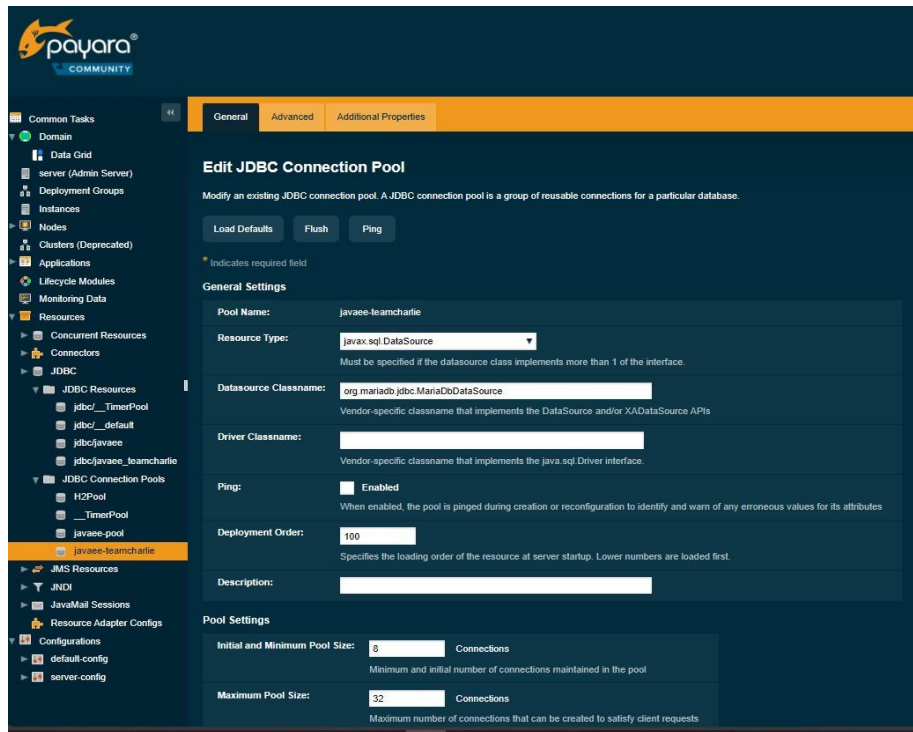
Click on the New which will navigate to New JDBC Connection Pool.

Provide the necessary parameters as shown in the picture.

- Pool Name: javaee-teamcharlie
- Resource Type: javax.sql.DataSource
- Database Drive Vendor: leave blank

Click next and set the following parameters for Datasource Classname and the properties for URL, user, and password and click finish to save the connection pool settings.

We can verify the connection by selecting the connection pool and clicking “Ping” which should display the “Ping Succeeded”



## For JDBC Resources

Navigate to the Resources/JDBC/JDBC Resources

Click on a new JDBC Resource and fill in the parameters as shown in the picture.

- JNDI Name: jdbc/javaee\_teamcharlie
- Pool Name: javaee\_teamcharlie
- Description: JavaEE Team Charlie

Click on the "OK" and verify all the parameters and add properties as required.

payara<sup>®</sup>  
COMMUNITY

User: admin Domain: domain1 Server: localhost Enable Asadmin Recorder

Deployment Groups  
Instances  
Nodes  
Clusters (Deprecated)  
Applications  
Lifecycle Modules  
Monitoring Data  
Resources  
Concurrent Resources  
Connectors  
JDBC  
JDBC Resources  
jdbc\_\_TimerPool  
jdbc\_\_default  
jdbcjavaee  
jdbcjavaee\_teamcharlie  
JDBC Connection Pools  
JMS Resources  
JNDI  
JavaMail Sessions  
Resource Adapter Configs  
Configurations  
default-config

### Edit JDBC Resource

Edit an existing JDBC data source.

Load Defaults

JNDI Name: jdbcjavaee\_teamcharlie

Pool Name: javaee-teamcharlie  
Use the [JDBC Connection Pools](#) page to create new pools

Deployment Order: 100  
Specifies the loading order of the resource at server startup. Lower numbers are loaded first.

Description: JavaEE Team Charlie

Status: ☒ Enabled

Additional Properties (0)

Add Property Delete Properties

Select	Name	Value	Description
No items found.			

## Deploying the application

Before we start the deployment, we should ensure that the domain and the database server are up and running.

## Installing the file realm on the server

To ensure that only authenticated users can access a security policy “realm” is defined. Users are assigned to user groups and are given specific roles.

## Edit Realm

Edit an existing security (authentication) realm.

[Manage Users](#)

\* Indicates required field

**Configuration Name:** server-config

**Realm Name:** evs-file-realm

**Class Name:** com.sun.enterprise.security.auth.realm.file.FileRealm

**Properties specific to this Class**

**JAAS Context:**   
Identifier for the login module to use for this realm

**Key File:**   
Full path and name of the file where the server will store all user, group, and password information for this realm

**Assign Groups:**   
Comma-separated list of group names

1. Start payara server
2. Go to payara admin page: localhost:4848
3. Go to Configurations > server-config > Security > Realms
4. Create new realm
  - Name: evs-file-realm
  - Class Name: com.sun.enterprise.security.auth.realm.file.FileRealm
  - JAAS Context: fileRealm
  - Key File: \${com.sun.aas.instanceRoot}/config/evs-keyfile
  - Assign Groups: UNI-AUTH
5. Click ok
6. create a text file named "evs-keyfile" with following content and place in the following directory:

C:\Users\<username>\payara5\glassfish\domains\domain1\config



7. Check on the file realm you created; you should see the listed users; you can change the passwords accordingly.

Current Security Roles

USER: Authenticated Users

STAFF: University Employees

ADMIN: Admin

## **Installation Guide of TestSMTP Server**

To enable email service in the application we need to set up SMTP server to send email reminders. Please follow the following steps to set up SMTP server.

1. Go to Payara Admin Console by right clicking on the Payara Server and click on “View Admin Console”. Before going there, make sure that Payara Server is running. (You can start the Payara Server from Project Explorer -> Services tab -> Servers -> Payara).
2. From Admin Console click on Resource Tab, after that click on JavaMail Sessions.
3. Click the New Button and fill in the following information as shown in the image.

**Basic**

JNDI Name: mail/evs-mail

Mail Host: localhost

Default User: admin

Password: admin

Auth: Disabled

Default Sender Address: admin@localhost

Deployment Order: 100

Description: Unrecognized connection to use SMTP server

Status: Disabled

**Advanced**

Store Protocol: smtp

Store Protocol Class: javax.mail.smtp.SMTPStore

Transport Protocol: smtp

Transport Protocol Class: javax.mail.smtp.SMTPTransport

Debug: Disabled

**Additional Properties (1)**

Name	Value	Description
mail/evs-mail	localhost	

4. Click save where we can see the listing of JavaMail Session settings.

5. Deploy the TestSMTPServer application in the same domain and the EVS application is ready to send emails.

## Problems Faced

The Pre-Development process consisted of a series of online meetings to discuss the coarse-grained details of the project. Issues during this face included timing constraints and limited availability when matching team schedules. During these meetings we discussed and separated tasks among group members. A Jira Board was then used to mark and keep track of progress throughout the development phase.

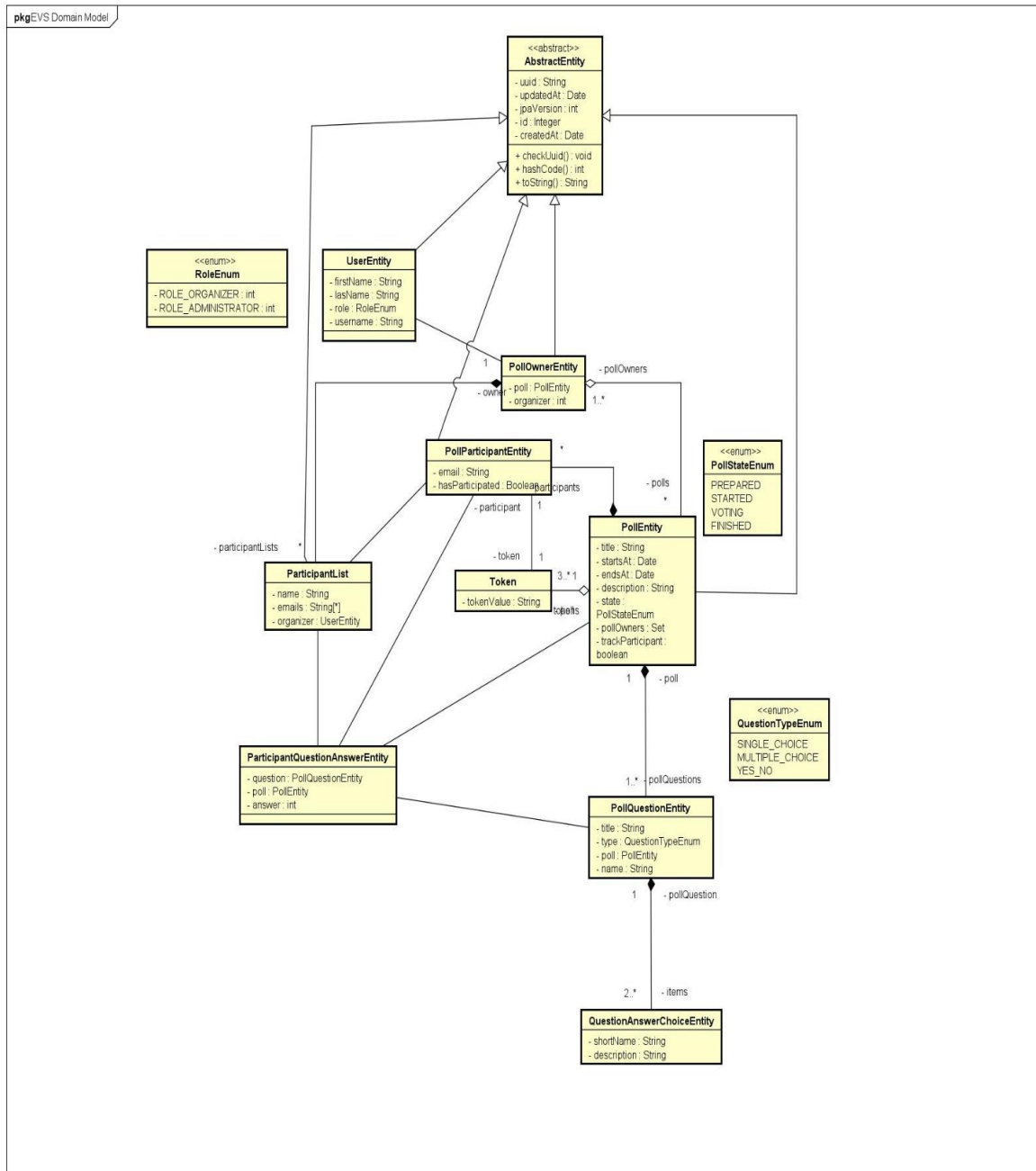
During the development stage we encountered constant “cannot create stateless EJB errors” as well as technical integration problems when combining jsf-frameworks. During the chart creation phase we initially wanted to display our results using primefaces, but found the integration with bootfaces incompatible and in the end used highfaces.

Additionally, we found in our final test, adding a css file had introduced a bug which stopped our modals from loading correctly and removing the css which created the bug introduced several formatting issues which we were unable to fully fix.

Integrating the concept of a backing bean into our structure, as well as using jsf were constant technical struggles which had to be adhered to and collaborated on throughout the project.

We finally faced the loss of a group member late in the project introduced an organizational issue into the structure late in the game, as many administration and translation tasks had to be redistributed to other members.

Domain Model



**Fig. Domain Model**

## Time Spent

We studied some of the courses, videos and weblinks prior to the project implementation.

1. Java EE: Getting started with Javasever Faces – by Jesper De Jong
2. Java Persistence API 2.1 - by Antonio Goncalves
3. <https://javaee.github.io/glassfish/documentation>

### Sanjeev Sun Shakya

Description	Time spent (in hrs.)
Lectures and Tutorials	10hrs
Understanding demo project	3hrs
Project setup	10hrs
Feature implementation	60hrs

### Eric Wayne Babcock

Description	Time spent (in hrs.)
Understanding Java EE	20 hrs.
questions/items	20 hours
User interface	5 hours

### Prajwol Thapa

Description	Time spent (in hrs.)
Understanding Java EE (courses and videos)	20 hrs.
Going through the application from lecture	10 hrs.
Working on application	20 hrs.

### List of Completed and Missed Requirements

The list encompasses the following completed and missed functionalities of the EVS.

Requirement No.	Description	Status
-----------------	-------------	--------

P001	<p>EVS provides organizers to conduct electronic polls where each poll shall have title, description, and a voting period.</p> <ul style="list-style-type: none"> <li>- Each poll has a unique title.</li> <li>- Each poll has a description.</li> <li>- Each poll has a voting period.</li> </ul>	Completed
P002	The number of questions in a poll is at least 1.	Completed
P003	EVS provides organizers with the ability to group many arbitrary questions in a poll.	Completed
P004	<p>EVS provides organizers with the ability to clone a poll.</p> <ul style="list-style-type: none"> <li>- If a poll to clone has votes, EVS clone the votes.</li> <li>- If a poll to clone has results, EVS will not clone the results.</li> </ul>	Not Completed
QU01	EVS provides organizers to add several types of questions to a poll.	Completed
QU02	EVS allows organizers to define the minimum and maximum number of choices for MCQ's.	Completed
QU03	The questions of a poll have a title.	Completed
QU04	The items of a question have a short name and an optional description.	
PS01	The state of poll is one of PREPARED, STARTED, VOTING, FINISHED.	Completed
PS02	EVS ensures that the description, initial voting period, questions, and participant list of a poll only changes in state PREPARED.	Completed
PS03	EVS ensures that the participant list of a poll only changes in state PREPARED.	Completed
PS04	Once the organizer starts the poll, the EVS sets the poll state to STARTED.	Completed
PS05	Once the voting period starts, the EVS sets the poll state to VOTING.	Completed
PS06	As soon as the voting period ends or all participants submits their vote, the EVS sets the poll state to FINISHED.	Completed
PS07	EVSs allow organizers to extend the voting period when a poll is in state STARTED or VOTING.	Completed
OR01	<p>EVS provides all university employees with the ability to act as organizers.</p> <ul style="list-style-type: none"> <li>- EVS ensures that organizers authenticate themselves with username and password.</li> <li>- EVS identifies university employees using the LDAP service provided by the ZIMT computing center.</li> </ul>	Completed
OR2	EVS provides organizers with the ability to create many arbitrary polls.	Completed

	<ul style="list-style-type: none"> <li>- Each poll is owned by at least one organizer.</li> <li>- EVS ensures that organizers can only work on polls that they own.</li> </ul>	
OR3	EVS provides organizers with the ability to preview their polls from the participant's view.	Not Completed
OR4	<p>EVS provides organizers with the ability to add further organizers to a poll.</p> <ul style="list-style-type: none"> <li>- EVS provides organizers with the ability to change the organizer list.</li> <li>- EVS ensures that an organizer cannot remove herself from the organizer list.</li> </ul>	Completed
OR5	<p>EVS provides organizers with the ability to delete a poll.</p> <ul style="list-style-type: none"> <li>- If a poll-to-delete is not in state PREPARED or FINISHED, EVS sends a notification email to all organizers and participants to inform them about the removal.</li> </ul>	Completed
OR6	EVS provides organizers to start the poll when it has at least 3 participants and in the state PREPARED.	Completed
AD01	<p>EVS provides admins the ability to delete any poll.</p> <ul style="list-style-type: none"> <li>- EVS sends a notification email to all organizers and participants to inform them about the removal, if a poll is not in state PREPARED or FINISHED.</li> </ul>	Completed
AD02	EVS ensures that administrators cannot access votes and results of polls that they do not own.	Completed
AD03	<p>EVS provides administrators with the ability to manage organizer accounts.</p> <ul style="list-style-type: none"> <li>- Once the organizer account is deleted, EVS deletes all the polls that belong exclusively to that organizer.</li> </ul>	Completed
PL01	<p>As long as a poll is in state PREPARED, EVS provides organizers with the ability to add arbitrary many participants to a poll.</p> <ul style="list-style-type: none"> <li>- Each participant is identified by a valid email address.</li> <li>- The EVS ensures there are no duplicate entries in poll participants.</li> </ul>	Completed
PL02	EVS provides organizers with the ability to add participants from outside the university.	Completed
PL03	As long as a poll is in state PREPARED, EVS provides organizers with the ability to delete participants from the participants of a poll.	Completed

	<ul style="list-style-type: none"> <li>- If the poll is not in state PREPARED, EVS ensures that the participants of the poll cannot be changed.</li> </ul>	
PL04	<p>As soon as the organizer starts the poll, EVS sends a notification to each participant.</p> <ul style="list-style-type: none"> <li>- The notification email contains poll title, description, voting period and other details.</li> <li>- The notification email contains a link that already holds a prefilled token of the participant, so that they can start to vote.</li> </ul>	Completed
PL05	EVS provides organizers with the ability to comfortably create participants lists.	Completed
PL06	<p>EVS provides organizers with the ability to store participant lists for easy reuse in subsequent polls.</p> <ul style="list-style-type: none"> <li>- EVS is designed such that stored participant lists are private to each organizer.</li> <li>- Each participant list has a unique name.</li> </ul>	Completed
TO01	As soon as a poll is started, EVS assigns a random token to each participant of a poll.	Completed
TO02	EVS is designed in a way that all tokens are unique in the scope of the entire system.	Completed
TO03	EVS is designed in a way that a token that was previously issued can never be regenerated.	Completed
TO04	The length of the tokens is equal to or greater than 16 characters.	Completed
AN01	EVS ensures anonymity.	Completed
AN02	EVS ensures that a token cannot be associated with a vote.	Completed
PT01	EVS provides organizers with the ability to track the participants for a poll.	Completed
PT02	If participation tracking is enabled for a poll, EVS provides organizers with the ability to activate automatic reminders to participants who did not vote.	Not Completed
SV01	As long as a poll is in state VOTING, EVS provides participants with the ability to submit a vote.	Completed
SV02	<p>The voting page presents an input field for a participant's token.</p> <ul style="list-style-type: none"> <li>- If the voting page is accessed by the link in the notification email, the token input field is pre-filled.</li> </ul>	Completed
SV03	As soon as the token is verified, the EVS displays the questions and items.	Completed
SV04	EVS presents a button to submit a vote.	Completed



SV05	As soon as the vote is submitted, EVS invalidates the token used for that vote.	Completed
SV06	EVS ensures that participants cannot change their vote after submitting.	Completed
SV07	EVS provides participants with the ability to cancel a vote.	Completed
SV08	EVS discards all choices of cancelled voting.	Completed
SV09	EVS ensures that subsequent participants using the same browser window cannot restore the previous choices	Completed
SV10	EVS provides participants with the ability to abstain from voting for each question in a poll.	Completed
SV11	EVS provides participants with the ability to abstain from voting for a complete poll.	Completed
RS01	As soon as a poll is FINISHED and the number of submitted votes is at least 3, EVS provides organizers with the ability to view the results of a poll.	Completed
RS02	EVS ensures that nobody can view results of polls with less than 3 submitted votes	Completed
RS03	As long as a poll is not FINISHED, EVS ensures that nobody is able to view (intermediate) results.	Completed
RS04	As soon as a poll is FINISHED and the number of submitted votes is at least 3, EVS provides organizers with the ability to publish the results.	Completed
RS05	As soon as a poll's results are published, EVS sends a notification email to all participants and organizers.	Completed
RS06	The result notification email contains a randomized result hyperlink to the results page.	Completed
RS07	The length of the random part of the result hyperlink is equal to or greater than 16 characters.	Completed
RS08	EVS shows the number of votes for each item of each question of a poll, the number of abstentions, and the voter participation in the result view.	Completed
RS09	EVS displays the results in graphical form.	Completed
UI01	EVS has implemented JSF technology.	Completed
UI02	EVS uses third-party CSS libraries.	Completed
UI03	EVS uses third-party JSF components.	Completed
UI04	EVS uses advanced technologies to enhance user experience.	Completed
UI05	EVS is designed in a way that supports at least one of the following desktop/laptop web browsers.	Completed
UI06	EVS is designed to support mobile device browsers.	
IN01	EVS is designed in a way that the user interface language can be switched.	Completed

	<ul style="list-style-type: none"> <li>- The language switch is realized either by a URL prefix, by detecting the client browser's language setting, or on user request.</li> <li>- If the language switch is done on user request, EVS remembers the user's language setting.</li> </ul>	Not Completed
IN03	EVS supports at least two user interface languages.	Completed
IN04	EVS supports English as UI language.	Completed
IN05	EVS supports German as UI language.	Completed
IN06	EVS supports multilingual text also in the data items, so that the participants can view their options in their preferred language.	Not Completed
IN07	EVS sends notification and reminder mails to participants in their preferred language.	Not Completed
AR01	EVS has been implemented according to the layered architecture and the architectural rules provided in the document evs-architecture.pdf.	Completed
AR02	EVS contains two modules i.e., web and EJB.	Completed
AR03	The team asked for consent before using third-party libraries not presented in the lecture.	Completed
AR04	EVS uses one of the database servers MariaDB/MySQL (preferred), PostgreSQL, or Java DB/ Derby.	Completed
AR05	The naming conventions for projects, JavaEE modules, database connection pools, databases, and other files and resources have been followed strictly.	Completed
AR06	The project team has prefixed all global names with its team name.	Completed
AR07	The Java naming conventions have been followed.	Completed
SE01	EVS is designed in a way that all passwords are stored in hashed/encrypted form.	Completed
SE02	EVS is designed in a way that network communication with its users is encrypted.	Completed
SE03	EVS is designed in a way that network communication with the database server is encrypted.	Completed
SE04	EVS is designed in a way that access to polls is only granted to the related organizers and administrators. Administrators may see only the poll title, voting period, and the organizers.	Completed
PM01	EVS has been documented in a project manual so that the supervisor can install the system on fresh Payara and MariaDB servers.	Completed

PM02	The project manual contains a list of completed/missed requirements. For the missed requirements, there is a rationale.	Completed
PM03	The project manual contains descriptions of any problems that occurred during the implementation.	Completed
PM04	Decisions made to change requirements have been documented in the project manual.	Completed
PM05	The individual team members' time spent on the project has been recorded.	Completed

**Note:** We could not complete some of the tasks, as one of our team members withdrew from the team in the last hours. Therefore, we were not able to reach the goal of completing all the tasks. However, we have completed all the “shall” tasks and only a few of the "should” tasks are not completed.

## Subsystems in EVS Application

The following subsystems exist in EVS application: -

### **1. Email reminder subsystem**

The email reminder subsystem is responsible for sending email reminders based on the configuration approved by the system administrator.

### **2. Scheduler subsystem**

The scheduler subsystem is responsible for scheduling the email reminders to the poll organizers, participants etc.

## Using the Application

### Login

The first page of the application is the login page. The application will only allow the user listed in the realm file; hence the new registration of the user is not possible.

Log In

Please provide your Username and Password

Please log in

User Name \*

Password \*  [Log In](#)

Current user: (not logged in)  
User locale: en - Deutsch - English  
Current Date: Aug 31, 2023, 8:42:54 PM  
Electronic Voting System - JavaEE Web Applications - Summer Term 2023

### Admin Page

Admin Dashboard

Welcome Admin, admin

☒ Manage Polls  
Manage And View All Polls  
[Manage Polls](#)

☒ Manage Organizer  
Manage And View All Organizers  
[Manage Organizers](#)

Current user: admin - [Log Out](#)  
User locale: en - Deutsch - English  
Current Date: Aug 31, 2023, 9:03:15 PM  
Electronic Voting System - JavaEE Web Applications - Summer Term 2023

The admin can login to the page using the credential. The admin page provides two features; where they can manage the polls and manage the organizers.

### Manage Polls

Electronic Voting System

Dashboard

Polis By sanjeev

Manage Participant's List

Manage Question

Polis Owned By Others

View Results

Welcome, sanjeev

Manage Polls

Add Poll

#	Title	Description	Start Date	End Date	Poll State	Track Participant	Primary Owner	Action
1	sdfsdf	fsdfsdf	Fri Sep 01 22:50:00 CEST 2023	Sat Sep 02 22:50:00 CEST 2023	PREPARED	false	sanjeev	<div>Start Poll</div> <div></div>

1

Current user: sanjeev

Staff member

Log Out

User locale: en - Deutsch - English

Current Date: Aug 31, 2023, 8:51:58 PM

Electronic Voting System - JavaEE Web Applications - Summer Term 2023

The organizer can manage the polls through this dashboard. They can choose the period to start and end the polls and actions depending on the state of the poll. The state of the poll changes from prepared to started and finished; on the different phases of the poll period.

## Add Poll

Electronic Voting System

Dashboard

Polis By sanjeev

Manage Participant's List

Manage Question

Polis Owned By Others

View Results

Welcome, sanjeev

Add Poll

Title

Description

Starts At

mm/dd/yyyy --:-- --

Ends At

mm/dd/yyyy --:-- --

Track Participant

☐

Submit

Current user: sanjeev

Staff member

Log Out

User locale: en - Deutsch - English

Current Date: Aug 31, 2023, 8:52:25 PM

Electronic Voting System - JavaEE Web Applications - Summer Term 2023

The organizer can add the poll by clicking on the “Add Poll” from the manage poll menu. Adding the poll requires the title, description, and period for the associated poll.

## Edit Poll

Electronic Voting System

Welcome, sanjeev  
Back to List

Title \* edfd

Description fcdsdf

Start Date \* 09/01/2023 10:50 PM

End Date \* 09/02/2023 10:50 PM

Poll State PREPARED

Track Participant ☐

Owner sanjeev

[Edit Participant's List](#) [Add new organizer](#) [Add Question](#) [Save Changes](#)

Current user: sanjeev [Log Out](#)  
User locale: en - Deutsch - English  
Current Date: Aug 31, 2023, 8:52:42 PM  
Electronic Voting System - JavaEE Web Applications - Summer Term 2023

The edit poll page can be reached from the “manage poll” page by clicking on the edit button on the action column.

## Add Question

The application provides organizers to add three distinct types of questions: single choice, multiple choice, and yes/no questions. The question includes its title, description and multiple answers depending on the type of questions. The organizer can define the minimum and maximum number of choices for multiple choice questions. Once the questions are selected, we can see the poll questions in the manage question webpage.

[Back to List](#)

### Add Question

Title \*

test

Description

test

Type \*

MULTIPLE\_CHOICE

Answer 1

sdf

Remove

Answer 2

sdf

Remove

Answer 3

sdf

Remove

Minimum Choice \*

2

Maximum Choice \*

3

Add Answer

Submit

Fig. Multiple Choice Question

Welcome, sanjeev

[Back to List](#)

### Add Question

Title \*

Description

Type \*

YES\_NO

Submit

Current user: sanjeev [Staff member](#) - [Log Out](#)

User locale: en - [Deutsch](#) - [English](#)

Current Date: Aug 31, 2023, 8:53:45 PM

Electronic Voting System - JavaEE Web Applications - Summer Term 2023

Fig. Yes/No Question



Welcome, sanjeev  
[Back to List](#)

### Add Question

Title \*

Description

Type \*

SINGLE\_CHOICE

Answer 1

Remove

Answer 2

Remove

Minimum Choice \*

Maximum Choice \*

Add Answer

Submit

Fig. Single Choice Question

## Manage Organizer

Manage Organizers

Go to Home Page

Show 10 entries

Search:

Username	Last Name	First Name	Role	Action
sanjeev	sanjeev	sanjeev	ROLE_ORGANIZER	<div></div>

Showing 1 to 1 of 1 entries

Previous

1

Next

Current user: admin - [Log Out](#)

User locale: en - [Deutsch](#) - [English](#)

Current Date: Aug 31, 2023, 9:02:59 PM

Electronic Voting System - JavaEE Web Applications - Summer Term 2023

This webpage is directed from the admin home page after clicking on the manage organizer. It allows admin to assign the role to the organizer.

## Add Organizer

The admin can add new organizer where the webpage shows the details of organizer with its id, name, and action it can perform.

The screenshot shows the 'Add Organizer' page in the 'Electronic Voting System'. On the left is a dark sidebar with navigation links: Dashboard, Polls By sanjeev, Manage Participant's List, Manage Question, Polls Owned By Others, and View Results. The main content area has a header 'Welcome, sanjeev' and a title 'Add Organizer To Poll'. Below the title, there's a 'Poll Name' field with the value 'sdfs'. A section titled 'Organizers:' contains a table with columns 'ID', 'Organizer', and 'Action'. The table has one entry with ID '20' and Organizer 'sanjeev'. Below the table, it says 'Showing 1 to 1 of 1 entries'. There are 'Previous' and 'Next' buttons, with '1' highlighted. A blue button 'Add new organizer' is at the bottom left. At the bottom right, there's a footer area with user information: 'Current user: sanjeev Staff member - Log Out', 'User locale: en - Deutsch - English', 'Current Date: Aug 31, 2023, 8:53:22 PM', and 'Electronic Voting System - JavaEE Web Applications - Summer Term 2023'.

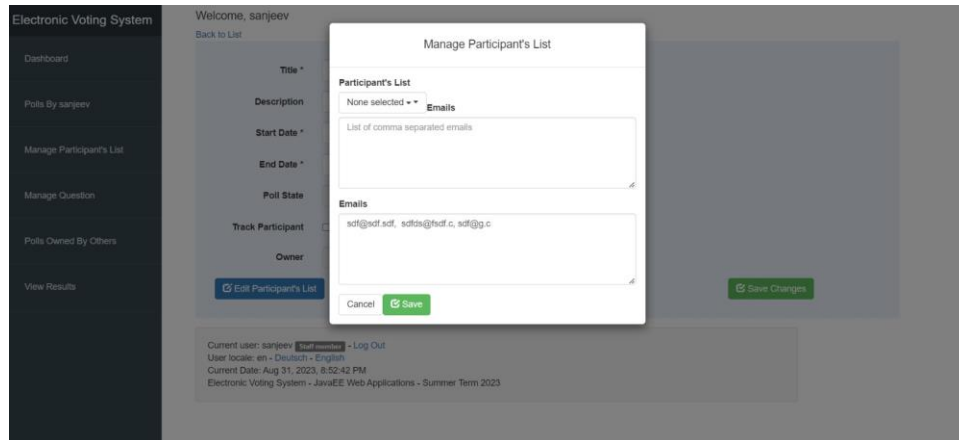
## Create Participant List

The organizer can create the participant list by clicking on the “Create Participant’s List” where the name and emails of the participant can be saved for the polling.

The screenshot shows the 'Create Participant's List' modal in the 'Electronic Voting System'. The modal has a title 'Create Participant's List' and two input fields: 'Name \*' with the value 'Participant List's Name' and 'Emails \*' with the value 'List of comma separated emails'. There are 'Cancel' and 'Save' buttons at the bottom. The background shows the same 'Add Organizer' page as the previous screenshot, but it is dimmed. The sidebar and footer are also visible.

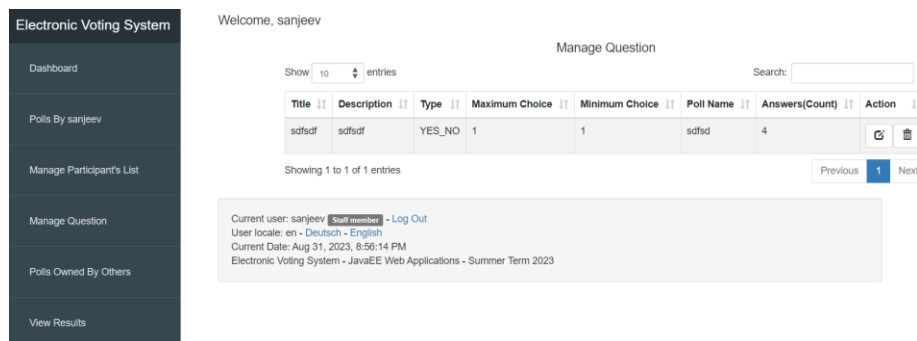
## Manage Participant List

The organizer can manage the participants list by sending a notification email about the state of the poll.



## Manage Questions

Once the question is chosen by the organizer to conduct the polling, the questions are directed to this section with all the details of the poll that includes: - title, description, type, number of choices, poll name, answer counts and action. Action has two features, delete the poll, and edit the poll.



## Edit/Delete Questions

The action button provides two features: -

- a. Edit the question: It allows organizer to edit the question, its description, title, and nature.
- b. Delete the question: It allows organizer to delete the poll question.

## Manage Results

The poll requires at least 3 votes to finish, where the organizer can view, publish the results of the poll from the action column.

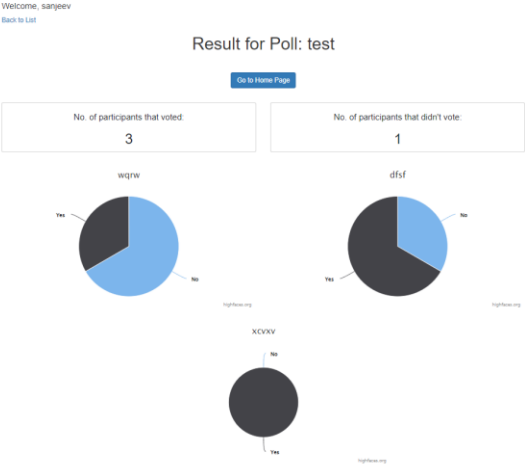
The screenshot displays the 'Electronic Voting System' interface. On the left is a dark sidebar with the following menu items: Dashboard, Polls By sanjeev, Manage Participant's List, Manage Question, Polls Owned By Others, and View Results. The main content area has a header with 'Welcome, sanjeev' and a 'View Results' button. Below this is a table with the following columns: Title, Start Date, End Date, Poll State, Number of votes, and Action. The table contains one entry with the title 'test', start date 'Thu Aug 31 22:58:00 CEST 2023', end date 'Thu Aug 31 23:02:00 CEST 2023', and poll state 'FINISHED'. The 'Number of votes' is 3. The 'Action' column has two buttons: 'Publish' (green) and 'View' (blue). Below the table, it says 'Showing 1 to 1 of 1 entries'. At the bottom, there is a footer with the following text: 'Current user: sanjeev [Staff member] - Log Out', 'User locale: en - Deutsch - English', 'Current Date: Aug 31, 2023, 9:04:03 PM', and 'Electronic Voting System - JavaEE Web Applications - Summer Term 2023'.

Title	Start Date	End Date	Poll State	Number of votes	Action
test	Thu Aug 31 22:58:00 CEST 2023	Thu Aug 31 23:02:00 CEST 2023	FINISHED	3	<a href="#">Publish</a> <a href="#">View</a>

## View Results

The results can be viewed in the pie chart after clicking on the view button on the action column.

Electronic Voting System
Dashboard
Polls By sangeev
Manage Participant's List
Manage Question
Polls Owned By Others
View Results



Glossary

Term	Definition
------	------------

Organizer	An organizer is a university staff member who conducts and organizes voting. They are responsible for managing, configuring, and operating the EVS.
Application server	It is a software framework that is responsible for hosting and managing applications and services that serve client applications over a network
Administrator	An administrator is also a university staff member who manages organizer's account and deletes poll.
Participant	A participant may or may not be a university member who can submit their vote and view results.