ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT027-3-3-EPDA

Enterprise Programming for Distributed Application

HAND IN DATE: 2 DECEMBER 2022

WEIGHTAGE: 50%

TP Number	Student Name
TP060810	Sia De Long

Table of Contents

Assumptions 3
Part 1: Evaluation Report 5
History and Architectural Evolution of Distributed Computing 5
Enterprise Application and Architecture 10
Supply Chain Management Solution Architecture 13
Selected APU system 17
Potential Improvements 18
Part 2 19
Design of Web Components 19
User Manual 21
Part 3 33
System Architecture and Interconnection 33
Business Tier 33
UML Diagrams 34
Use Case Diagram 34
Class Diagram 35
Part 4 36
Database Design 36
Additional Features 38
Check Announced Result 38
View All System Users 39
References 40

Assumptions

The system is implementing a web-based online election system which will be presented to APU SRC election committee. The system will be developed in NetBeans using enterprise application that applying the JSP, servlet and EJB framework which achieving the architecture of multi-tier including presentation tier for interface, business tier for logic and database tier for data storing. There are some assumptions given to the system:

- All users are the student of APU with a student ID for login.
- One election will have one or many seats defined for a SRC position with a maximum contester.
- Election will be started upon created until the date given as an attribute.
- Result of election will not be announced immediately until the committee member confirm and announce the result.
- All users can enrol to many elections regardless the result of the approval request and election result.
- One user can only vote for one contester for each election while voting for themselves is allowed.
- Date data are assumed to be key in correctly with the format given by the users.
- Data validation is existing for all input.

Besides that, there will be total two type of role in the system which are student and committee member as admin while contester can be enrol from student and committee member. The functionalities of each user type are as shown below:

a) Committee Members' Functionalities:

- To add, delete, search, and update all type of system users' information.
- To approve or reject contester on chosen election
- To set the date of the election.

- To define the number of seats for each election and define the number of contesters for each seat.
- To monitor the election progress for every ongoing election and finally announce the winning contesters.
- To check every enrolment request progress.
- In addition, some reporting functionality is required. Committee members would like to get as much useful information from the system as possible and is therefore open to the inclusion of any other useful reports.

b) Students' Functionalities:

- To register and edit individual profile.
- To send enrolment request for an election and check individual request progress.
- To vote for a specific seat.
- To check the announced result.

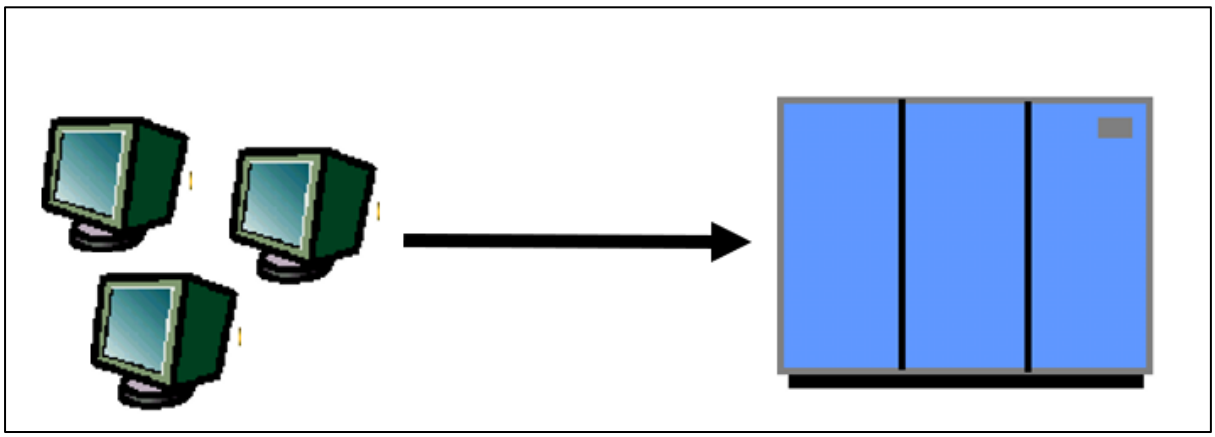
c) Contesters' Functionalities:

- To register and edit individual profile.
- To contest for a specific seat.
- To view the current vote count for every contester in the participated election.
- With the functionalities of committee member or student depend on the role of the system registered

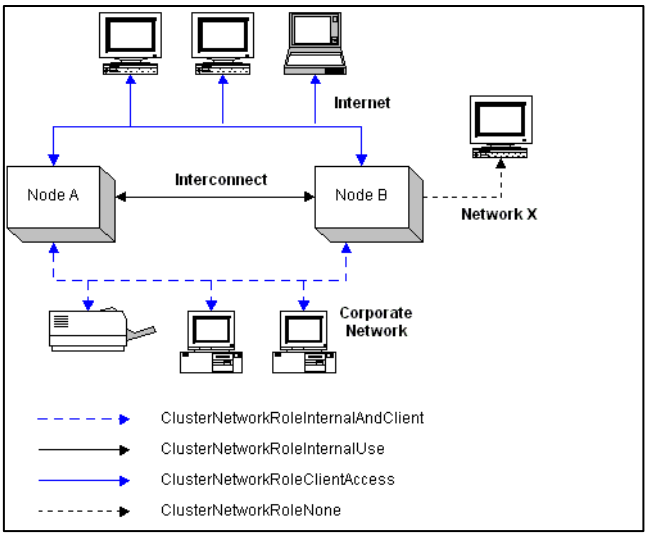
Part 1: Evaluation Report

History and Architectural Evolution of Distributed Computing

Distributed computing is an architecture that enable multiple computers work together to overcome and solve a common problem. It is an evolution that all about evolving from centralization to decentralization it started from the centralised system such as mainframe back in early 1955

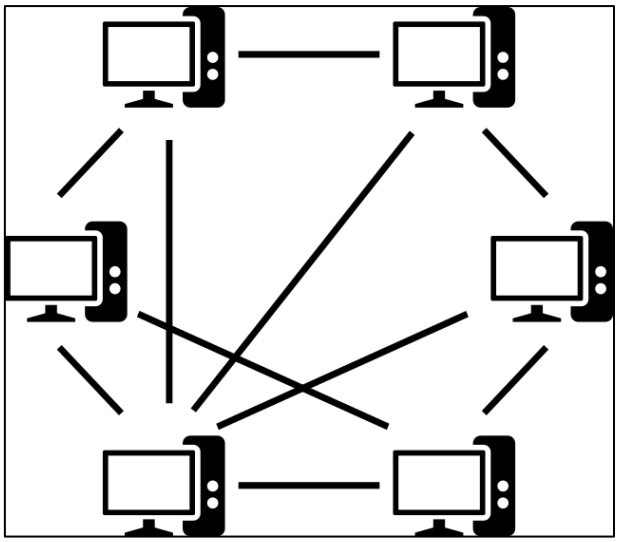


which is the best solution for processing large-scale data, mainframe conceptualised the client-server architecture and enabling a single time-sharing system.



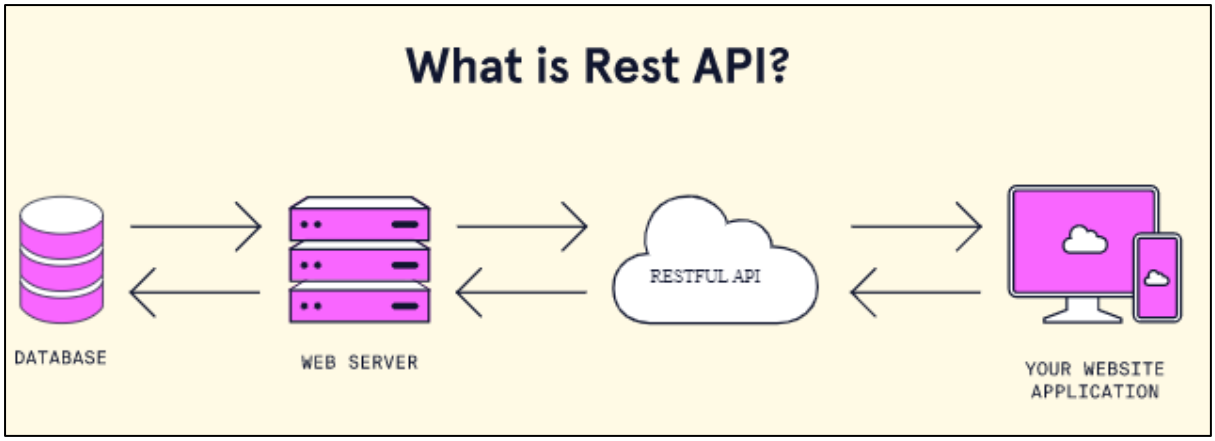
Custer network was introduced in the early 1970s which is an alternative for mainframe system and consists of collection of similar workstations or PCs. During 1967-1974, ARPANET and early network that enabled global message are created and followed with the TCP/IP that create communication mainly through datagram transport came into existence.

TCP/IP then begun to transform the Internet into several connected networks and linking multiple local networks to a wider Internet. It had caused the number of hosts in the network increase rapidly until the point which the centralised naming systems could not provide sufficient scalability. Hence Domain Name Systems (DNSs) was introduced in 1985 to transform domain name to IP addresses. The creation of HyperText Transfer Protocol (HTTP) and HyperText Markup Language (HTML) was happened in 1980-1990s which introduced the first web browsers, websites and web-server. With the standardisation of TCP/IP, the infrastructure was provided for interconnected networks of networks known as the World Wide Web (WWW). The complexity and challenges in the aspect of application-to-application interaction was increasing due to the number of PC-based application programs running on independent machines growing so rapidly. The advent of network computing is the one that enables remote procedure calls (RPCs) over TCP/IP which is also the most accepted method of application software communication. Every software application can be running on different hardware and operating system causing it to be a challenge when communicate with each other for data sharing. Thus, these challenges have become the main factor to lead to the concept of distributed computing applications. First distributed application architecture is Peer-to-peer (P2P) computing,

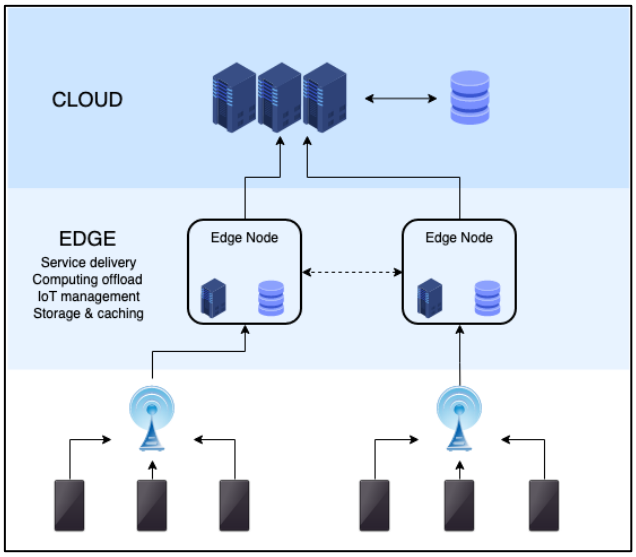


it partitions tasks between peers without the need of central coordinator while sharing equal privileges among peers meaning every client acting as both client and server which enables decentralised Internet. Multiple tasks can be completed by computers jointly connected over a network due to the introduction of grid computing, it makes use of data grid to directly interact with each other to perform similar tasks by using middleware. Web service brought up platform-independent communication which enables direct application-to-application interaction using Internet based on XML-based information exchange systems. Web service also enable Java to communicate with Perl and windows applications can communicate with Unix applications. However, the security for P2P system is weak since peers are all both clients and servers causing vulnerable to denial-of-service attack. Besides that, cloud computing came up and allow user to manage resources and application online over the Internet without having a physical hardware installed. Moreover, mobile computing allow user to transmit data over a wireless network meaning it is no longer need to connect mobile phones with switches, it also

introduced IoT which utilise sensors to exchange data. To improve the scalability, flexibility, portability, caching, and security of the system the evolution of Application Programming Interface (API) based communication over the REST model was needed.

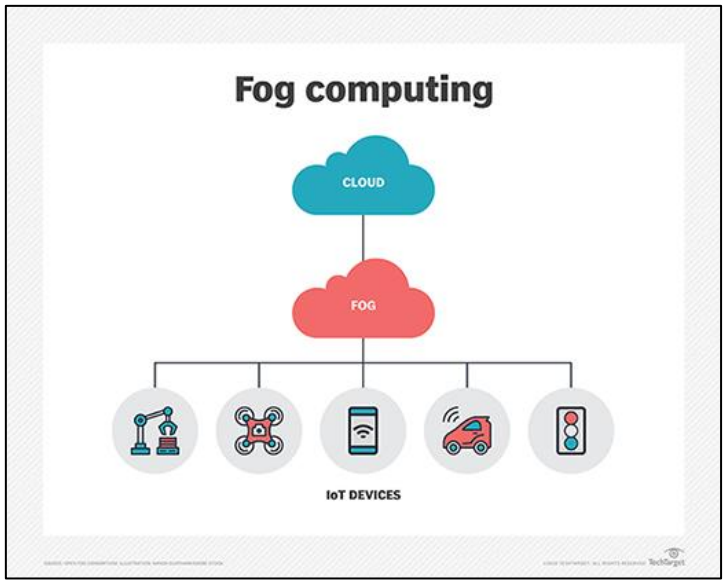


However, it is not improving these capabilities on every API separately but need for a common component to implement these capabilities on top of the API which lead to the evolution of API management platform evolution and has become one of the core features for every distributed system until today. In addition, virtual machines that enable single computer to act as multiple computers that run in parallel has come into existence because of having multiple system within one computer. With the huge data come from mobile computing and IoT services, edge computing concept was introduced to process million of data in real-time,



it is a concept which client data is processed at the periphery of the network and may cause latency issues due to data moving across a WAN. While fog computing can reduce the need

for bandwidth because it aggregating every bit of information at certain access point instead of send to cloud channel.

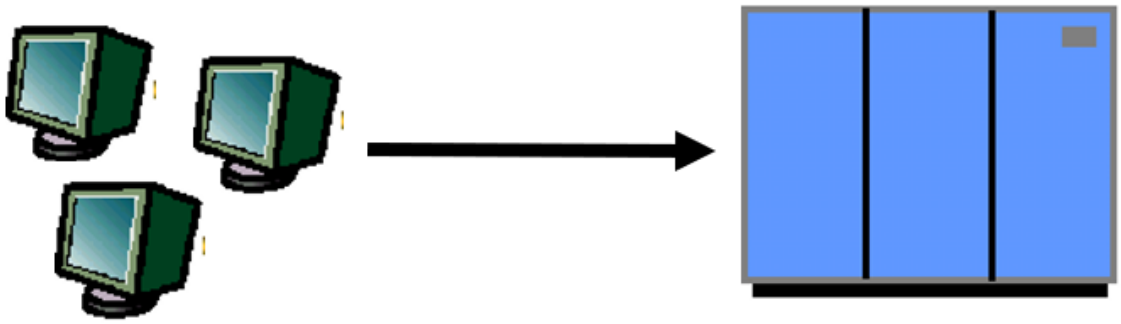


Hence, the composition of both fog and edge computing evolve cloud computing model from centralized stakeholders to decentralized multi-stakeholder systems. Last but not least, a container is a container image that can store application and dependencies and to be run on every environment that has a operating system which can run a container. Thus, it introduced the concept of container-based application deployment. In a nut shell, the current state of distributed system is programmed by application programmers while the underlying infrastructure management is done by a cloud provider, it is still kept on getting evolved. (GeeksForGeeks, 2022)

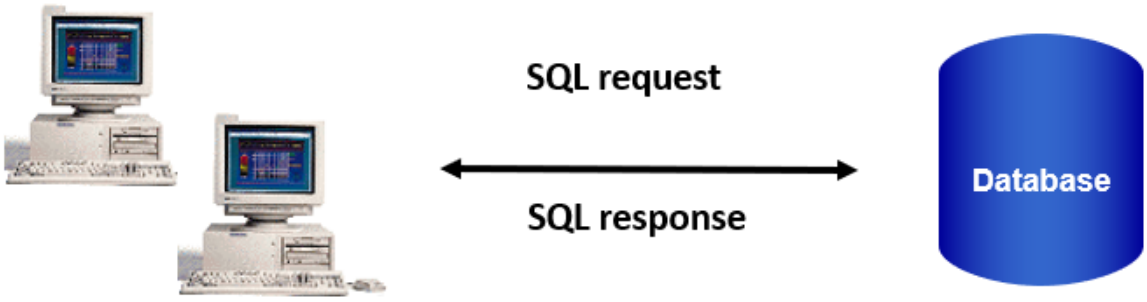
Enterprise Application and Architecture

An enterprise application is a big software system platform created for corporate environments like business or government. Enterprise application is based on the distributed computing concept and developed using enterprise architecture, so it is complex, scalable and component-based and consist of a group of program or software designed for unparalleled functionalities. (Techopedia , 2017) There are lot of types of enterprise application such as Enterprise Resource Planning (ERP) that are able to manage data from many different business processes, Supply Chain Management (SCM) which help business on supply chain transactions and operations, track inventory and manage supplier relationships. Customer Relationship Management (CRM) which manage customer data and information digitally and Knowledge Management Systems (KMS) that answer queries using the knowledge database given are also types of enterprise application. (Arora, 2021)

There are many types of enterprise architecture with their own advantages and disadvantages, the first architecture is one-tier architecture which keep all of the required elements for the application such as interface, middleware and back-end data in a single server like mainframe. This kind of architecture is the simplest and most direct and data consistence can be easily achieved but there is situation where one-tier architecture is not sufficient for web application and cloud hosting solution. (Techopedia, n.d.)



For two-tier architecture, there are two tiers for the system where client will act as the first tier while both database server and web application server will reside on the same machine will act as the second tier. The second tier in this architecture provide availability, scalability, and performance characteristics for the organization's web environment. (IBM, 2022)



Moreover, there is a well-established architecture which is three-tier architecture that organise application into three tiers. It has

```
private void DataGrid1_SelectedIndexChanged(object sender, System.EventArgs e)
{
// Object of the Property layer
clsStudent objproperty=new clsStudent();
// Object of the business layer
clsStudentInfo objbs=new clsStudentInfo();
// Object of the dataset in which we receive the data sent by the business layer
DataSet ds=new DataSet();
// here we are placing the value in the property using the object of the
//property layer
objproperty.id=int.Parse(DataGridl.SelectedItem.Cells[1].Text.ToString());

// In this following code we are calling a function from the business layer and
// passing the object of the property layer which will carry the ID till the database.
ds=objbs.GetAllStudentBsIDWise(objproperty);

// What ever the data has been returned by the above function into the dataset
//is being populate through the presentation laye.
txtId.Text=ds.Tables[0].Rows[0][0].ToString();
txtFname.Text=ds.Tables[0].Rows[0][1].ToString();
txtAddress.Text=ds.Tables[0].Rows[0][2].ToString();
txtemail.Text=ds.Tables[0].Rows[0][3].ToString();
```

presentation tier which is the user interface of the application,

```
// this is the function of the business layer which accepts the data from the
//application layer and passes it to the data layer.
public class clsStudentInfo
{
        public DataSet GetAllStudentBsIDWise(clsStudent obj)
        {
         DataSet ds=new DataSet();
         ds=objdt.getdata_dtIDWise(obj);// Calling of Data layer function
         return ds;
        }
}
```

application tier which is the tier that has the logic to process data and

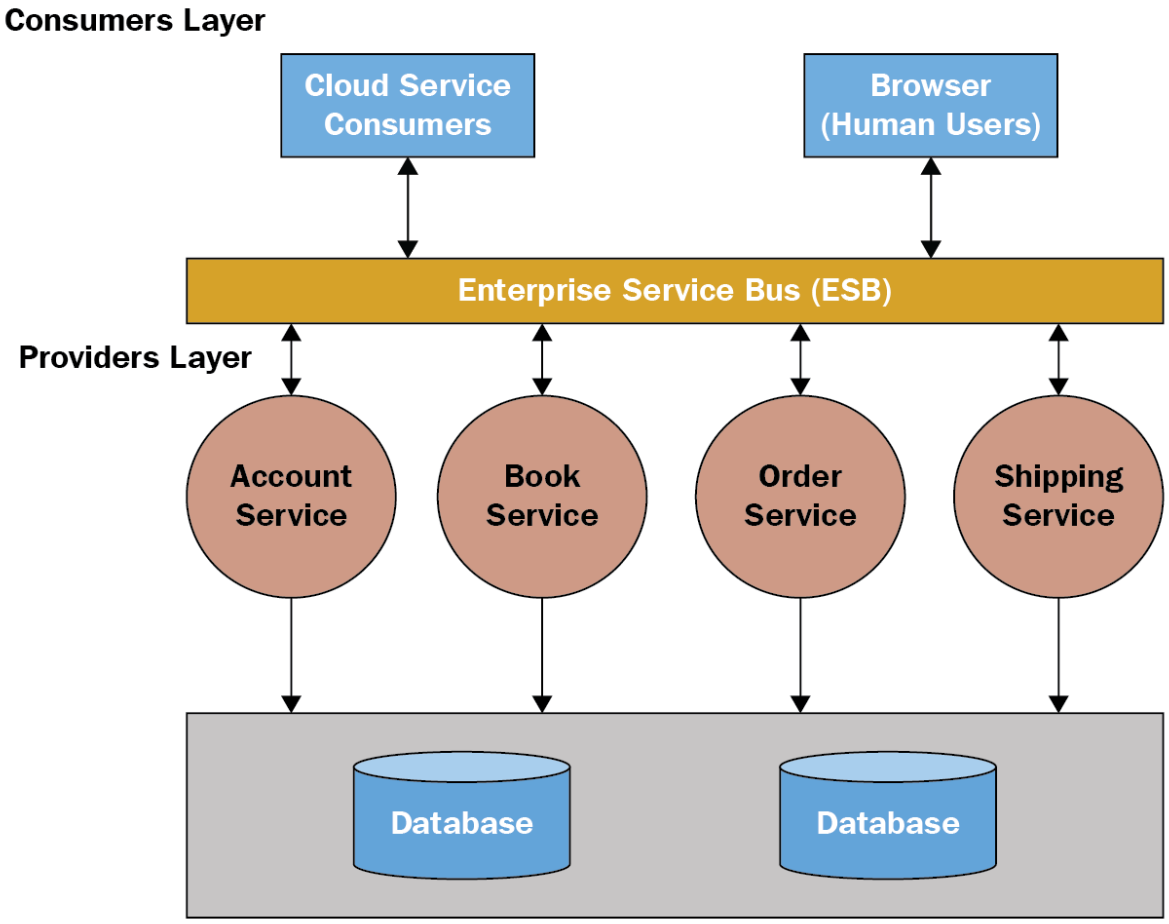
```
// this is the datalayer function which is receiving the data from the business
//layer and performing the required operation into the database

public class clsStudentData // Data layer class
{
        // object of property layer class
        public DataSet getdata_dtIDUise(clsStudent obj)
        {
         DataSet ds;
         string sql;
         sql="select * from student where StudentId=" +obj.id+ "order by StudentId;
         ds=new DataSet();
        //this is the datalayer function which accepts the sql query and performs the
        //corresponding operation
                ds=objdt.ExecuteSql(sql);
                return ds;
        }
}
```

the data tier which is the database to store and manage the data. Three-tier architecture is better than two-tier in term of application development speed, scalability, reliability and security. If the tier of the architecture is more than three which is very rare considering the cost, it is called n-tier architecture. (IBM, 2022)

Supply Chain Management Solution Architecture

Supply chain management covers all procedures that convert raw materials into final products and the management of the movement of goods and services. It is focus on an organisation's supply side to maximize customer value as well as competitive advantage among all the competitors. (FERNANDO, 2022)

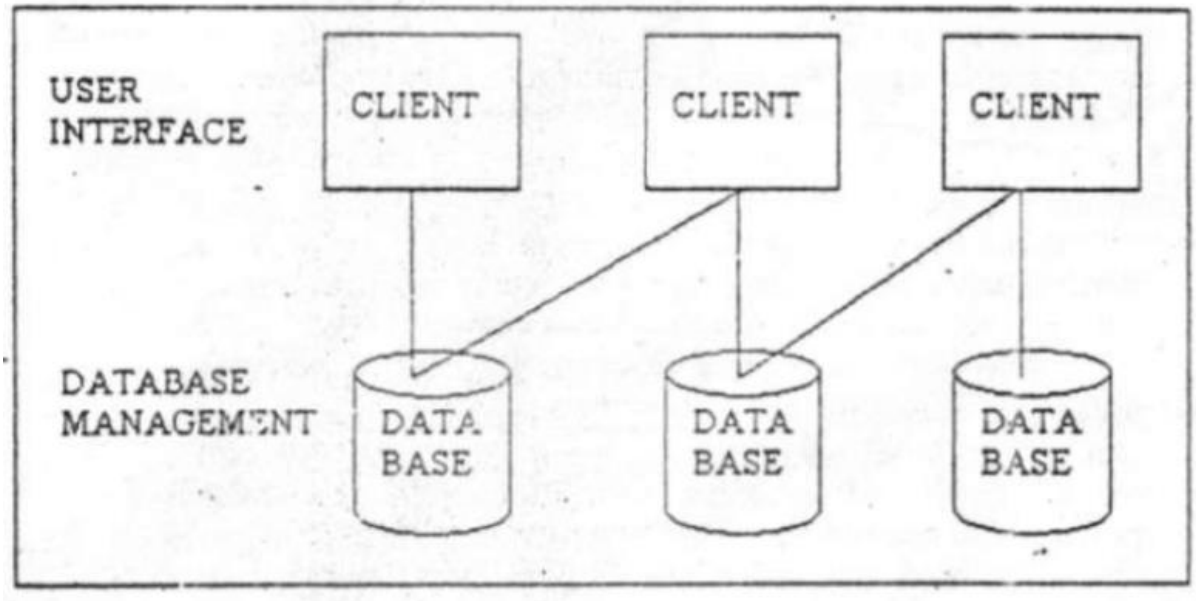
One of the enterprise architectures that can be used for supply chain management is Service-Oriented Architecture (SOA), it is a method of software design in which application components communicate with other components via a network using a communication protocol to provide services to them, this is an architecture which used by most of the banking system. The image below shows every service working independently without interrupt each other.



Its tenets are unaffected by suppliers and other technologies. Services can communicate with each other in one of two ways which are through passing data and through two or more services coordinating an activity. The concept of SOA can be varied depend on the organisation but the core values among them are Business value, Strategic goals, Intrinsic inter-operability, Shared

services, Flexibility and Evolutionary refinement. SOA can work with or without clod computing and there are some benefits to SOA, especially in a web-service based business. First and foremost, SOA create reusable code which cut down the time spend on development process as there is no need to reinvent the code when creating a new service and process, not only that, SOA allow the system to be using multiple coding languages at the same time because everything will be run through a central interface. Besides that, SOA allow various systems and platforms to function independent with each other which promote interaction of the system, not only that, it also allows to work around firewalls which enable companies to share services that are vital to operations. Moreover, SOA cuts back the client-service interaction which allowing all independent operation and achieve greater scalability from that. Last but not least, SOA limit the amount of analysis needed when developing custom solutions. (Medium, 2019)

Another of the enterprise architectures that can be used for supply chain management is two-tier client-server architecture. Two-tier architecture will have the user interface runs on the client and database will be storing in the server, while the user application logic can either run on the client or the server and both client and server process will be running on a different computer separately. Client process means by providing an interface to gather the data from the customer computer while server process means by providing an interface with the data. This is an architecture which can be used by an e-commerce system.



```
# Security group for app tier
resource "aws_security_group" "app_sg" {
    name = "app_sg"
    vpc_id      = aws_vpc.vpc.id


  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["10.0.0.0/16"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

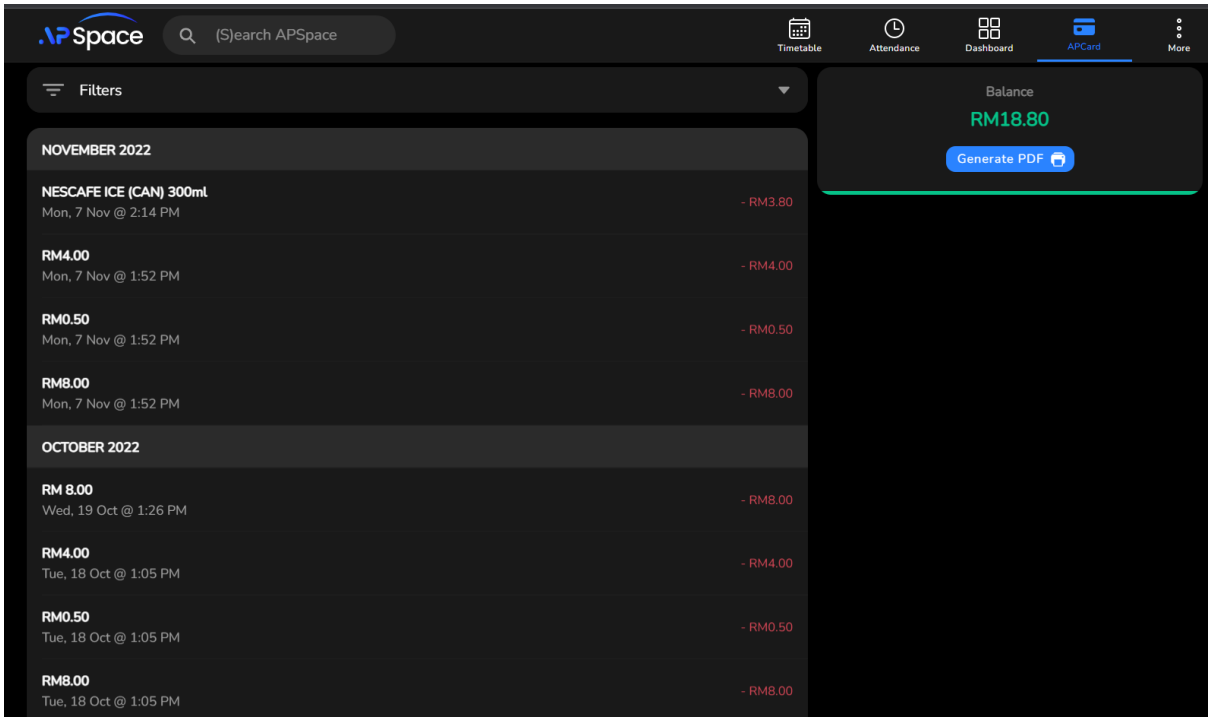
```
# Security group for database tier
resource "aws_security_group" "db_sg" {
    name = "db_sg"
    description = "allow traffic only from web_sg"
    vpc_id      = aws_vpc.vpc.id

  ingress {
    from_port   = 3306
    to_port     = 3306
    protocol    = "tcp"
    security_groups = [aws_security_group.app_sg.id]
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    security_groups = [aws_security_group.app_sg.id]
    cidr_blocks = ["10.0.0.0/16"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

A standardised, abstract interface is provided by the client-server architecture to enable communication between various components. These components constitute an integrated business application when they are coupled. Each module is a reusable and shared item that may be used in different business applications. Since functions of business application are separated, the application logic can be modified easily, not only that, business application object is working with its own encapsulated data structures that fit with a certain database, so

data will not be sent as entire database records instead they only send the specified data as parameters when communicate. As the advantages from it, it reduces the latency of the data communication since the network traffic reduced significantly and the development process can be more easy since developer can design the interface of presentation layer without knowing the business application logic. (MyReadingRoom, n.d.)

Selected APU system

The chosen APU system is the APCard system. It is a system that allow user to do money transaction through the card. Money can be reloaded to the card balance through a physical machine and it will update the balance in the database and display the balance and transaction history on APSpace application. Since it is a system that related to money transaction services, SOA architecture will be suitable to be implemented for it as banking system use the same architecture.



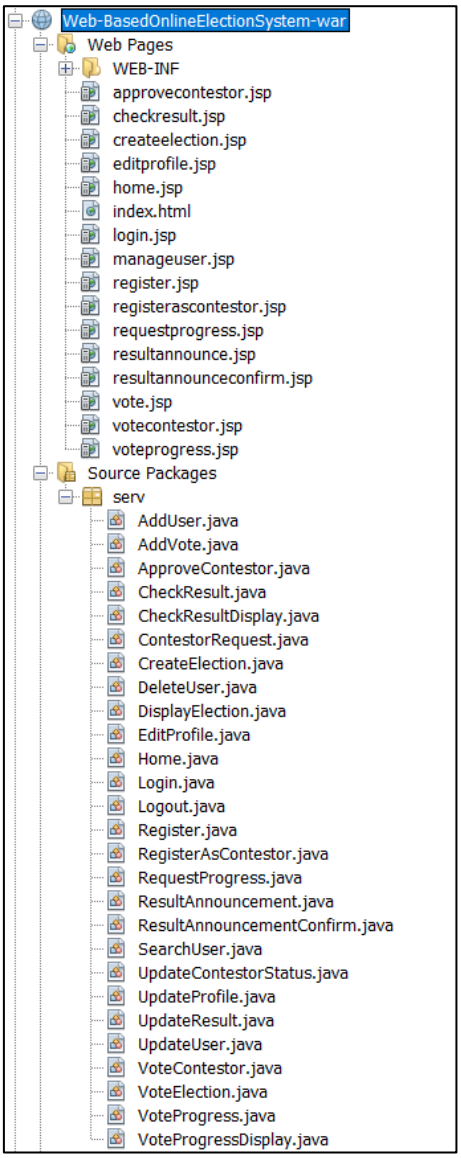
Potential Improvements

With the implementation of SOA to the APCard system, every service can work independently which is very useful for money transaction activities that may involve may type of services. First and foremost, the database can be improved to cloud computing-based, so that the card is not only limited for the usage on campus but also allow to do money transaction with associated library outside of the campus as an example since any service can be added easily to the system because it allows multiple coding language at the same time while the central interface will solve the issue. Besides that, balance of the account should be allowed to reload using bank merchant services such as public bank, Maybank and Hong Leong bank, platform of the bank side should not have any problem since it works separately with the overall system. To conclude, the implementation of SOA to APCard system can significantly improve the scalability of the system because of no dependencies can get in the way in this architecture.

Part 2

Design of Web Components

The web components used for the system are JSP and servlet. The JSP and servlet of the system are shown as bellow:



JSP will be the interface design for every webpage by using HTML language, required input will be collected from the form created on JSP with a submit button that will call for a servlet to run. Those input can be used on servlet by fetching the data using the parameter name declared on the JSP.

```
<td>ID              : </td>
<td><input type="text" name="a" size="20"></td>
```

This is the parameter name defining on the JSP while data fetching on servlet is shown as below:

```
String id = request.getParameter("a");
```

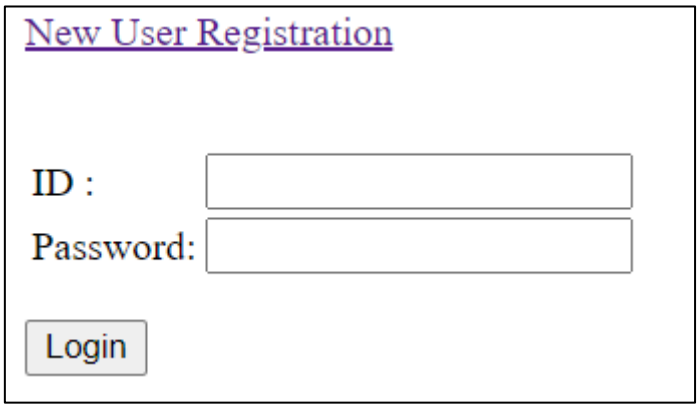
The servlet will then use the received data to run through the defined logic to determine the next JSP or servlet navigation as well as the message display for next page as shown below:

```
request.getRequestDispatcher("login.jsp").include(request, response);
out.println("<br><br><br>Registration done! Login to continue!");
```

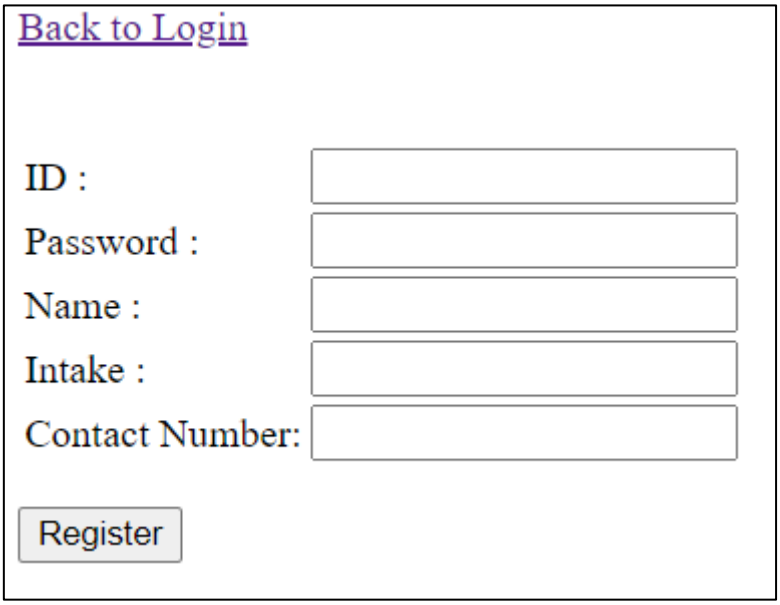
In order to link the database layer, EJB will be used from the servlet by declaring as shown below:

```
@EJB
private StudentFacade studentFacade;
```

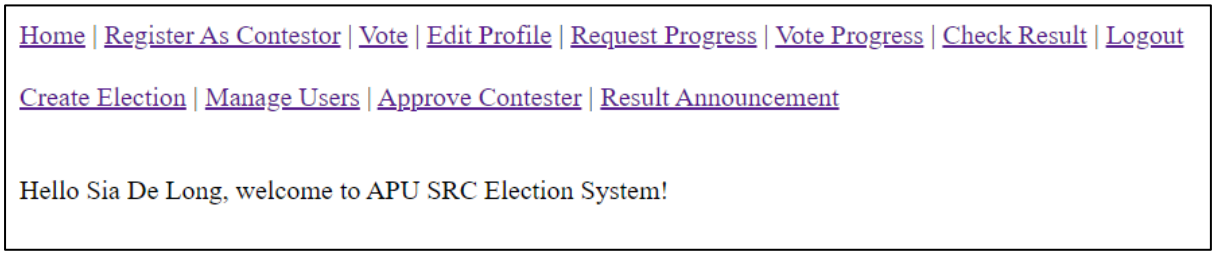
User Manual



This is the login page of the system, user can input their registered id and password in the text box to login to their account, user who do not have an account can click the registration to move to register page.



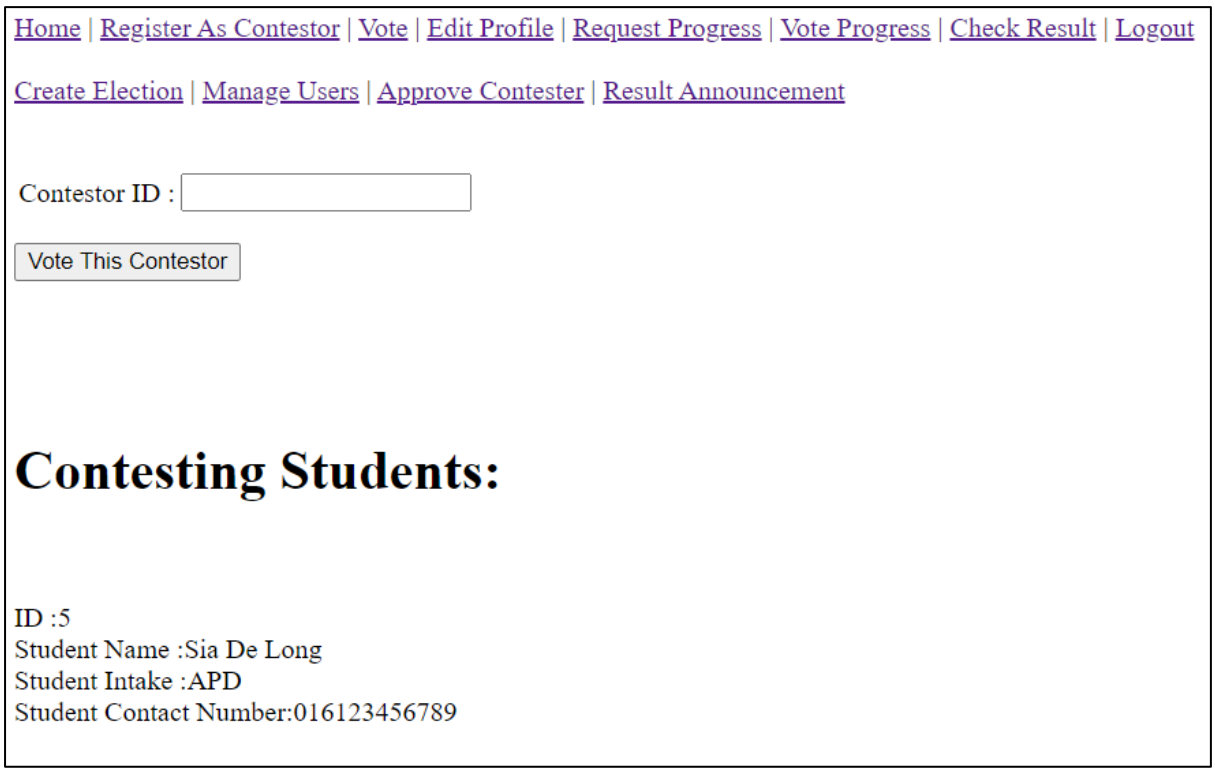
This is the register page of the system, user can input their information to register an account, account duplication and empty field will be detected after the button clicked. Page will be navigated to login page after a successful registration.



This is the home page of the system, the navigation banner will be appearing throughout the whole login session until the user logout, only the system admin can access to the admin functionalities which are the navigation for the second row of the banner. Every use case will finally be navigated back to home page with a message displaying at the bottom whether the previous action is successful or not.

This is the register as contester page of the system, any election that are available to enrol for the current user will be displaying at the bottom and validation will be checked for the election ID input, election will be only be available if date and maximum contester number is allowed.

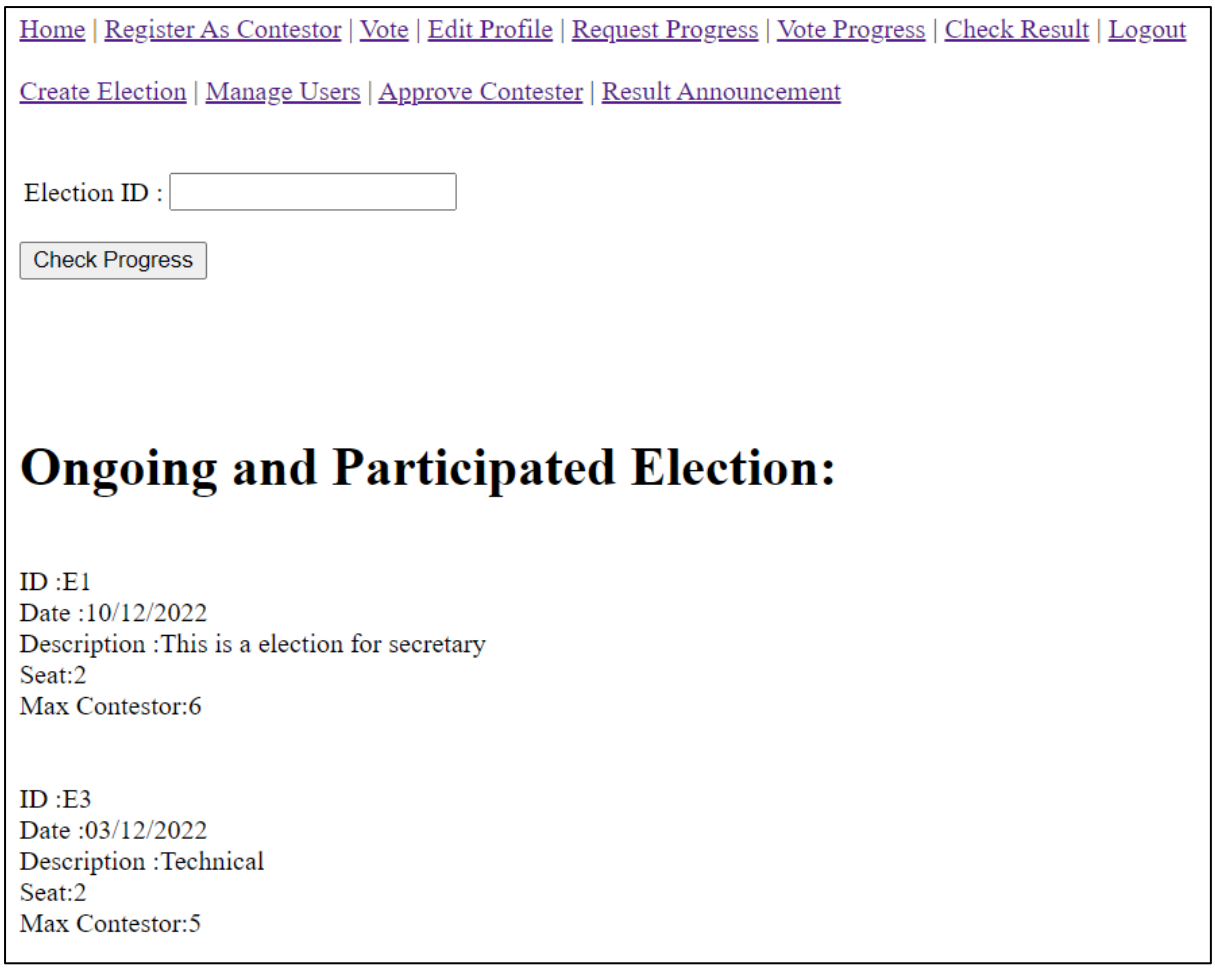
This is the election selection page for vote function, any ongoing election will be displaying at the bottom and validation will be checked for the election ID input. Page will be navigated to contester selection page after select an election ID.



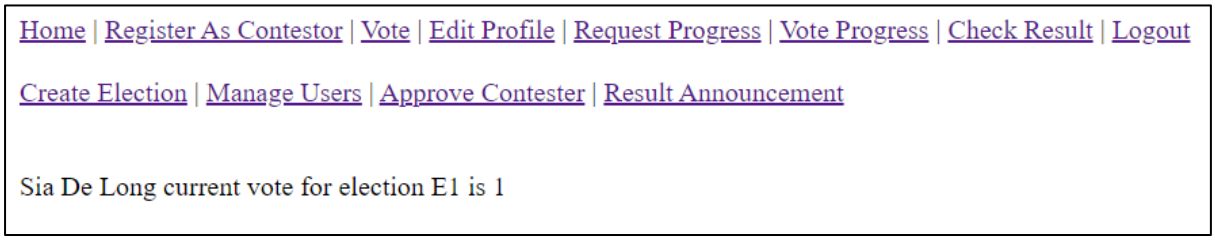
This is the contester selection page for vote function, any contester of the selected election will be displaying at the bottom and validation will be checked for the contester ID input.

This is the profile edit page of the system, user can input a new information to the text box to update their profile while the current profile information will be displaying at the bottom.

This is the request progress checking page of the system, admin of the system can check all of the request progress while other type of user only can check for their own request progress.

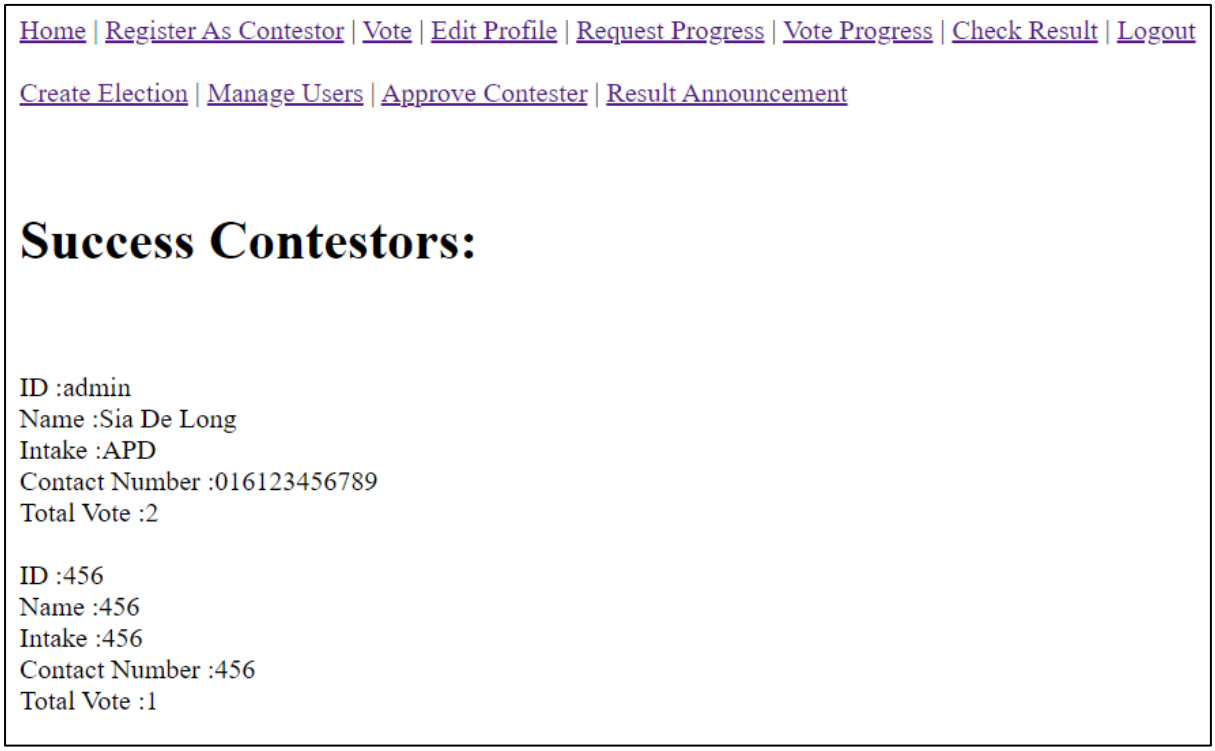


This is the vote progress checking page of the system, any election that are participated will be displaying at the bottom and validation will be checked for the election ID input. All ongoing election will be displayed if the current user is admin.

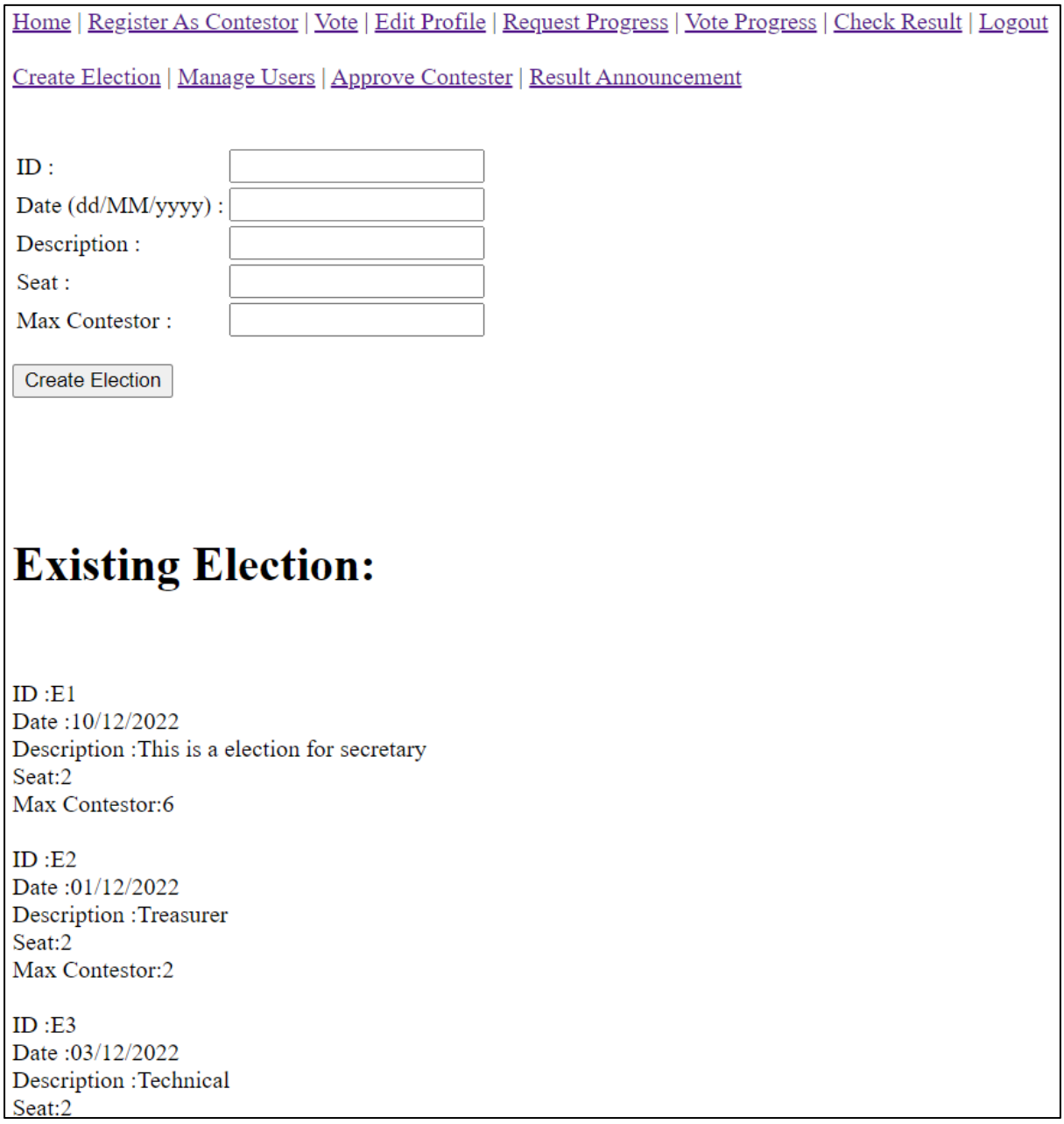


The current vote will be displaying with the navigation while showing all contester of the selected election.

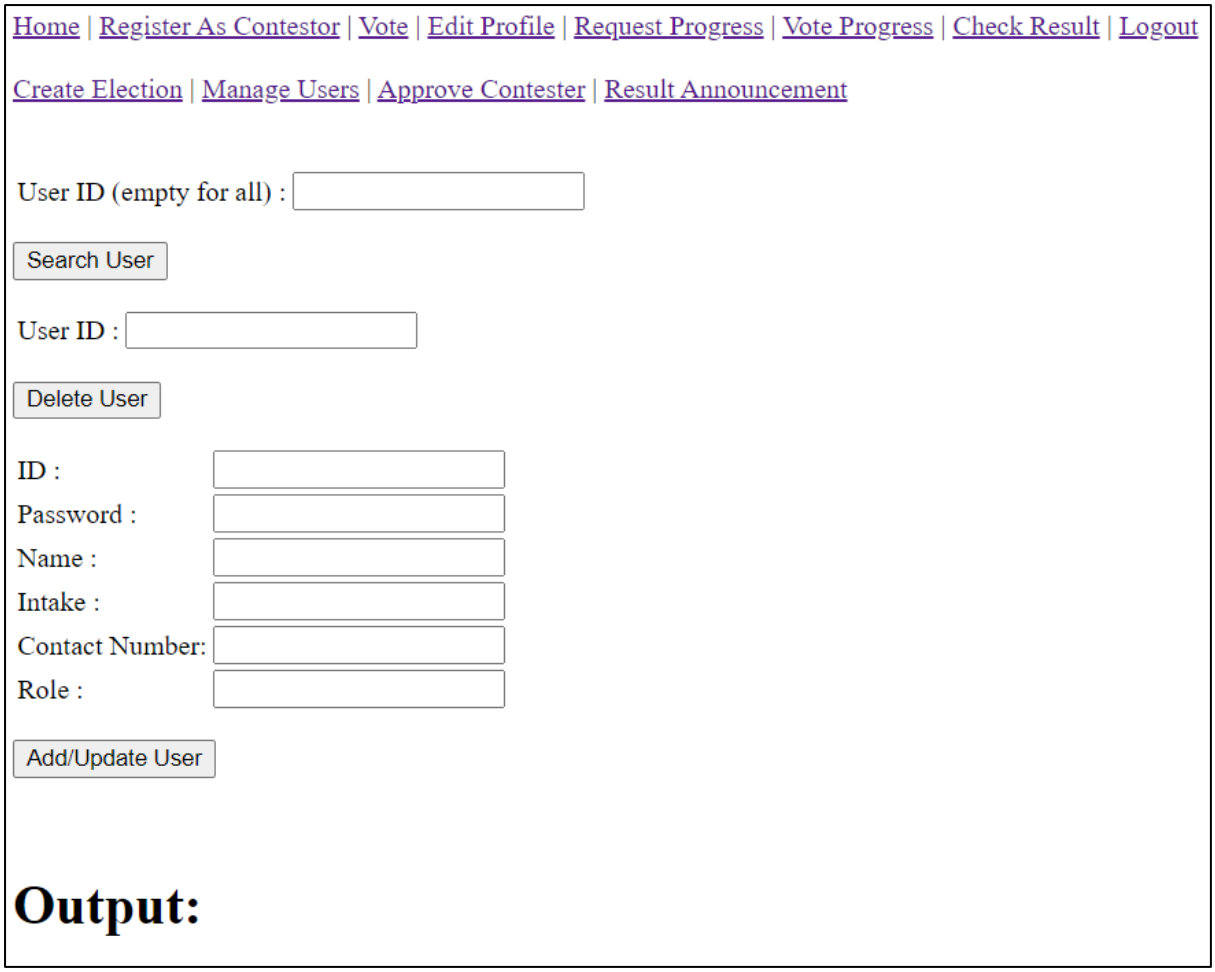
This is the result of election checking page of the system, any election that are announced will be displaying at the bottom and validation will be checked for the election ID input.



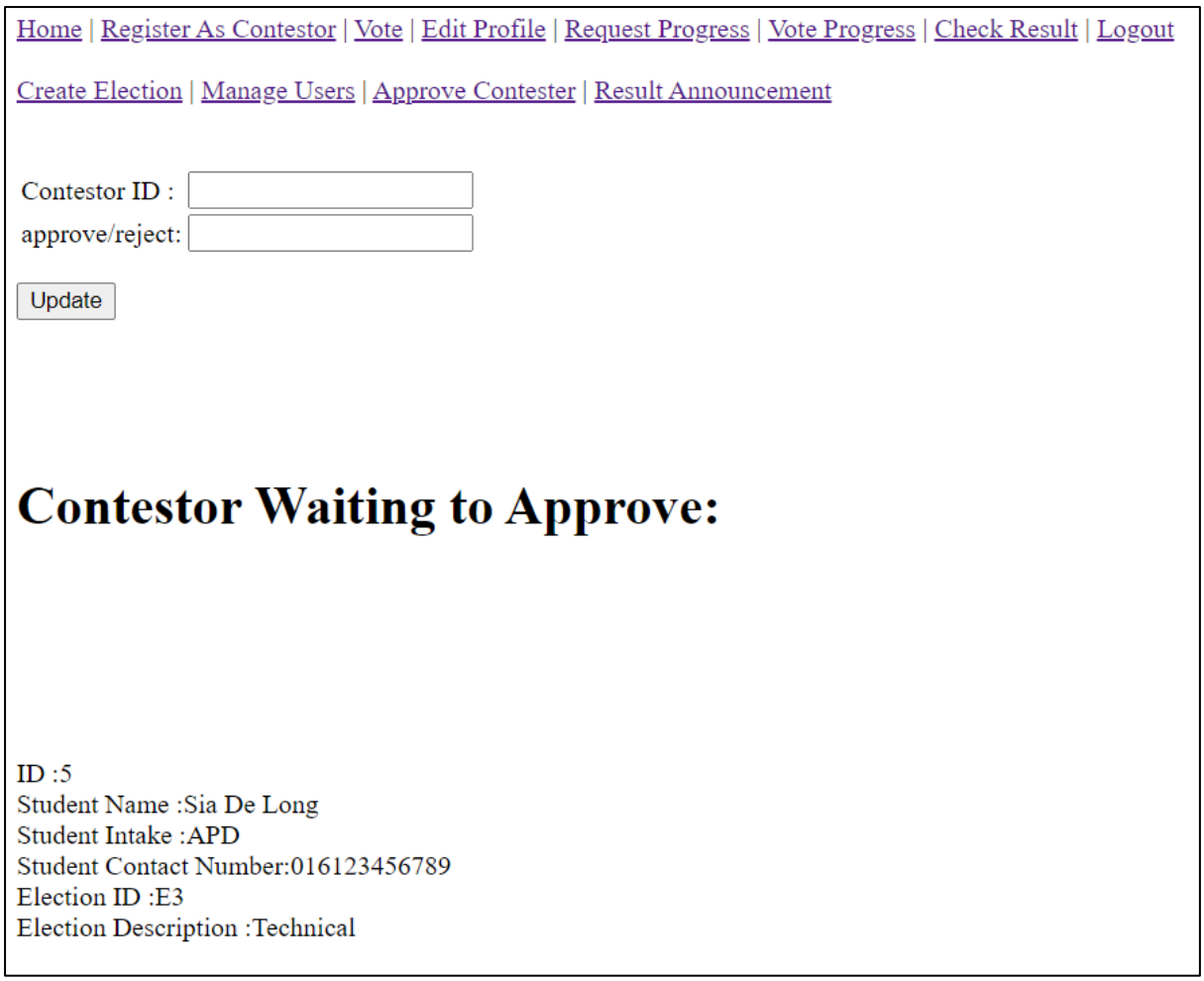
The result will be displaying with the navigation while showing all success contester of the selected election.



This is the create election page of the system, any election that are announced will be displaying at the bottom and validation will be checked for the election ID input.



This is the manage user page of the system, search can use to display all user existing in the system if empty value is given, all field will be gone through validation and output will be displayed at the bottom of the page.



This is the approve contester page of the system, any contester waiting for approval will be displaying at the bottom and validation will be checked for the contester ID input. Approval of approve or reject can be given for every request.

This is the announce result page of the system, any election waiting for announce will be displaying at the bottom and validation will be checked for the election ID input.

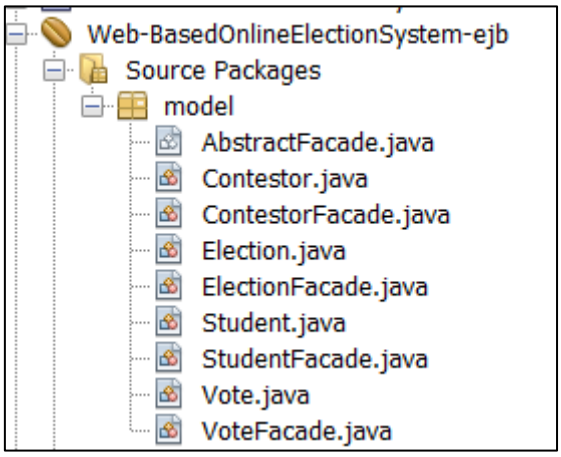
This is the confirm announce page of the system, any contester approved for the election will be displaying at the bottom with respective vote count and the final success contester who have the highest vote count will be displayed at the bottom too.

Part 3

System Architecture and Interconnection

There are three layers for the system architecture which are presentation tier responsible by JSP and servlet, business tier responsible by EJB which will be called function from servlet to process the connections to the backend database and database tier which is the database application used to store all relevant data of the application.

Business Tier



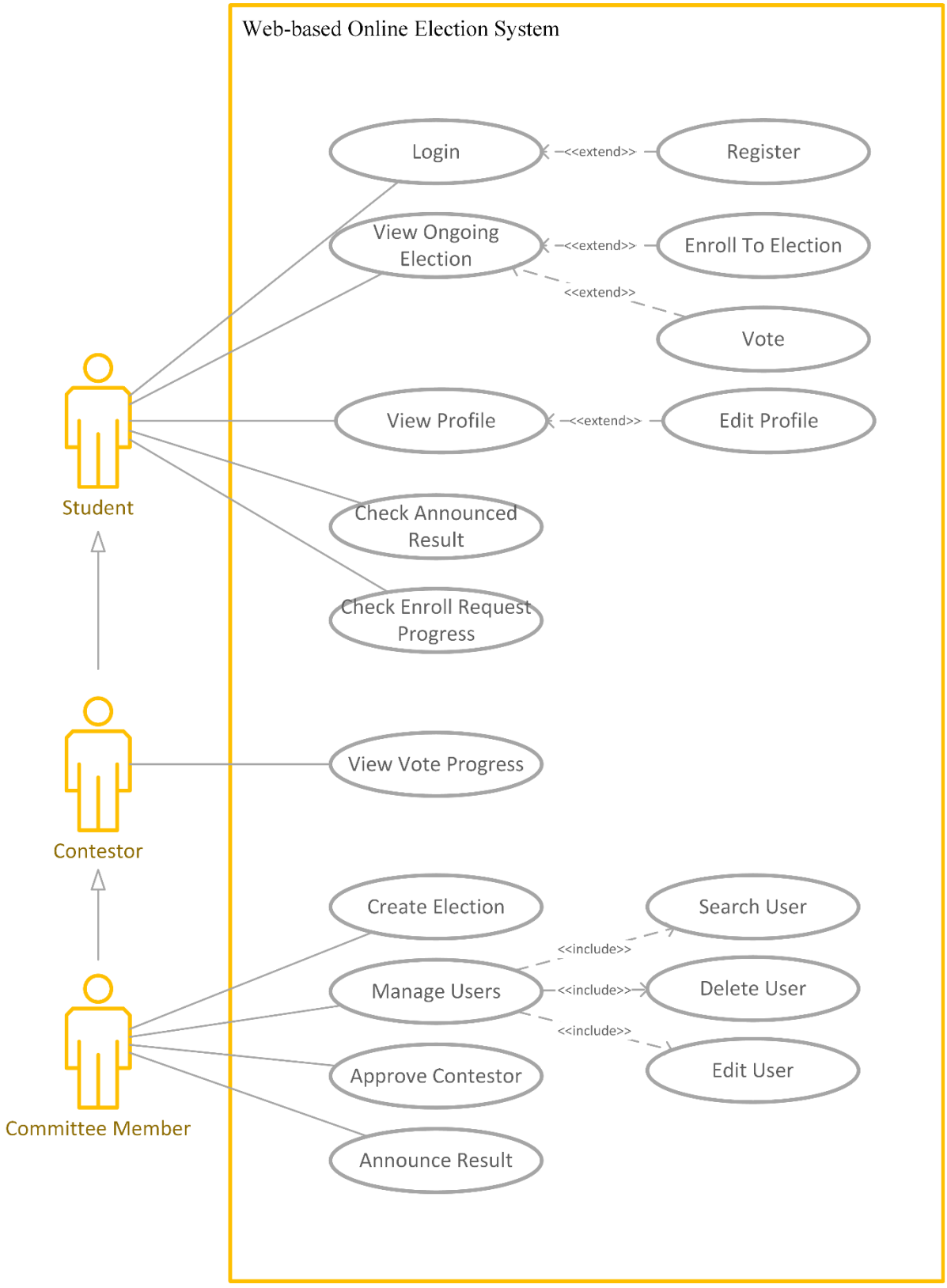
EJB of the system contain entity class and session bean for the created entity. Entity class is the blueprint that carrying necessary attributes to create a data row for the database storing. Facade is the session bean for created entity, they will connect the entity with respective table in the database. All facade will have the common function declared in the abstract facade such as create, edit, find and findAll. Any extra function needed can be declared on respective facade.

For example, the function of facade can be called on servlet after declaration of facade using EJB notation as shown below:

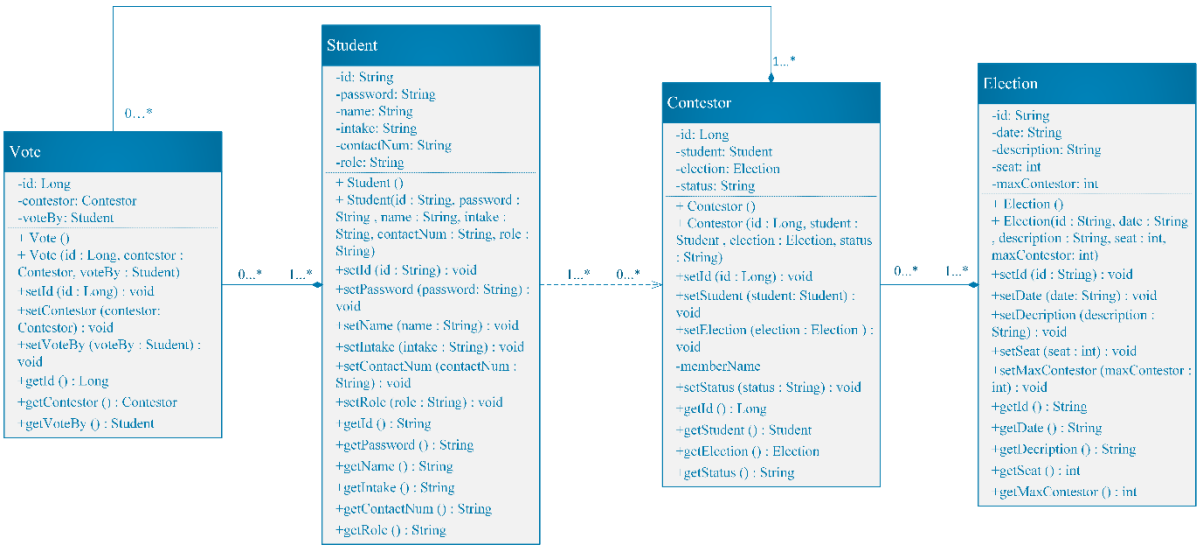
```
Student found = studentFacade.find(id);
```

UML Diagrams

Use Case Diagram

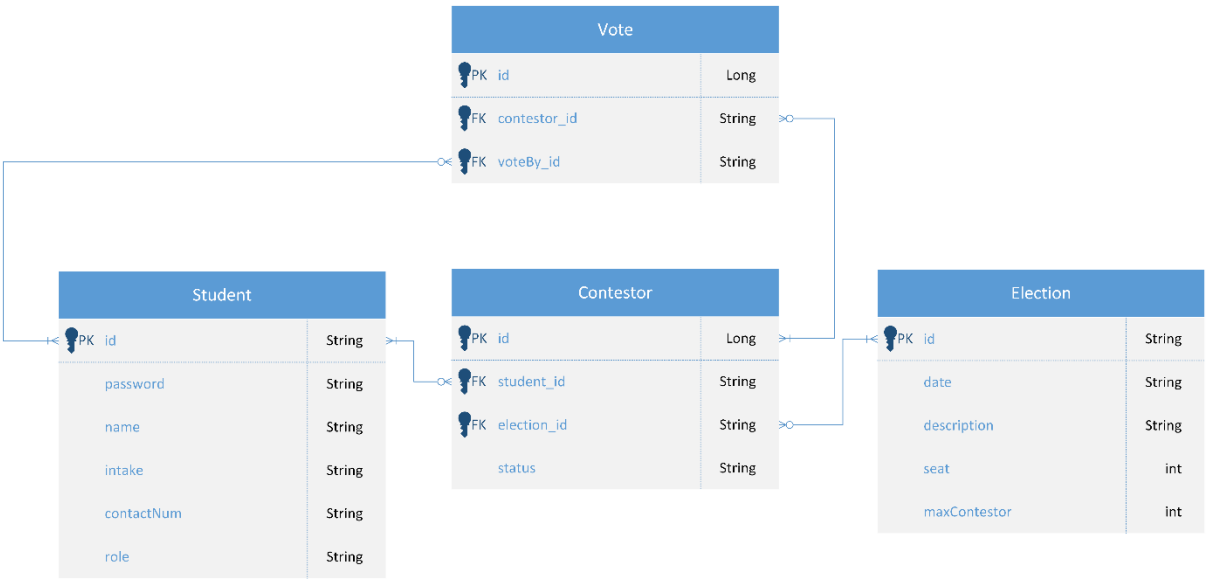


Class Diagram

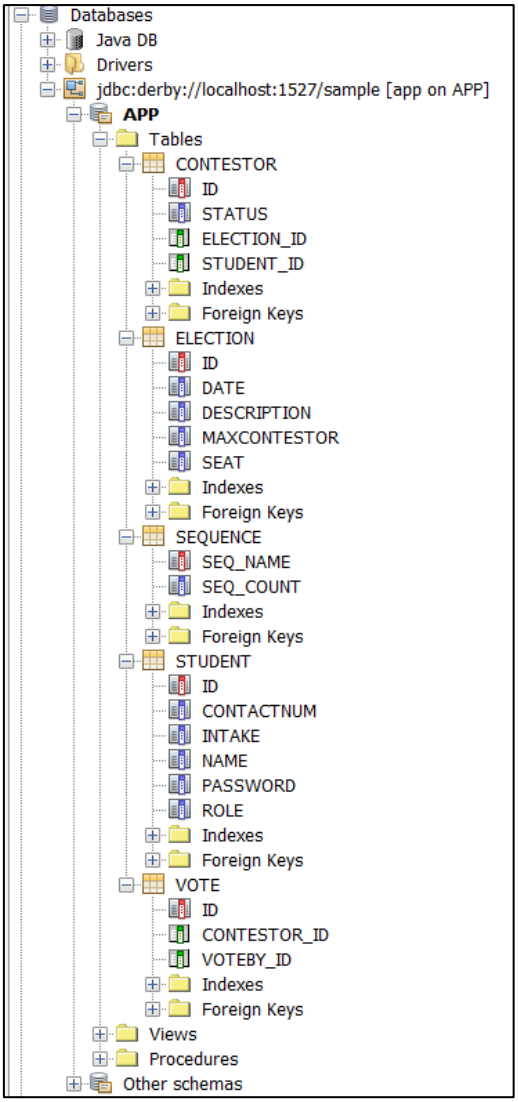


Part 4

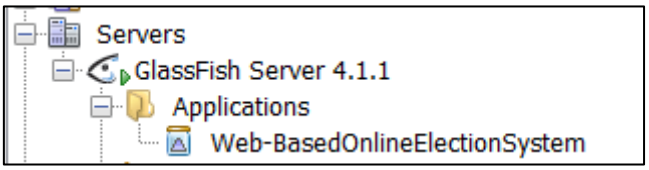
Database Design



There are total four table which are student, contestor, election and vote. Student table will be used to store every user account of the system with the role attribute differentiating between admin and student. Election table will be used to store every election created by the admin. Contestor table will be used to store data that having the student id request for which election while the status of the request can be waiting for approval, approve, reject, success and fail depend on the approval from the admin and election result. Vote table will be used to store voting from the system user, every vote will have the contestor id and student id which indicate who did the vote.



The database will be used by the system is local database which have the server of GlassFish Server 4.1.1 as shown below:



The server will deploy the application to build a connection between them.

Additional Features

Check Announced Result

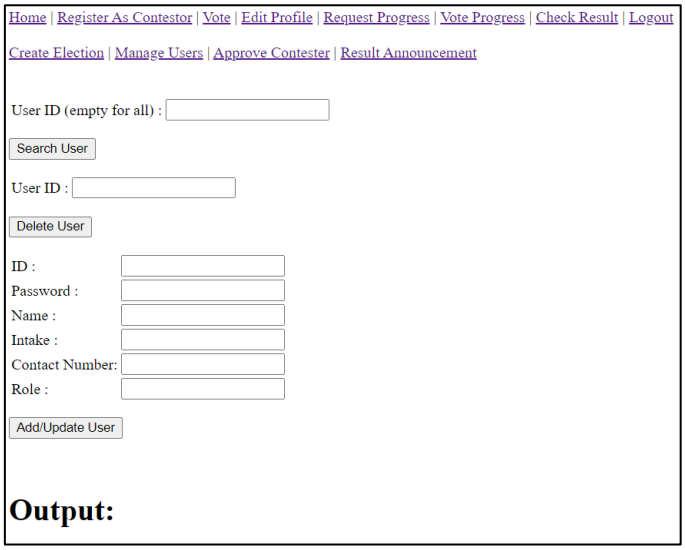
Every type of user of the system can check the result of the confirmed and announced result by accessing to the check result page.

This is the result of election checking page of the system, any election that are announced will be displaying at the bottom and validation will be checked for the election ID input.

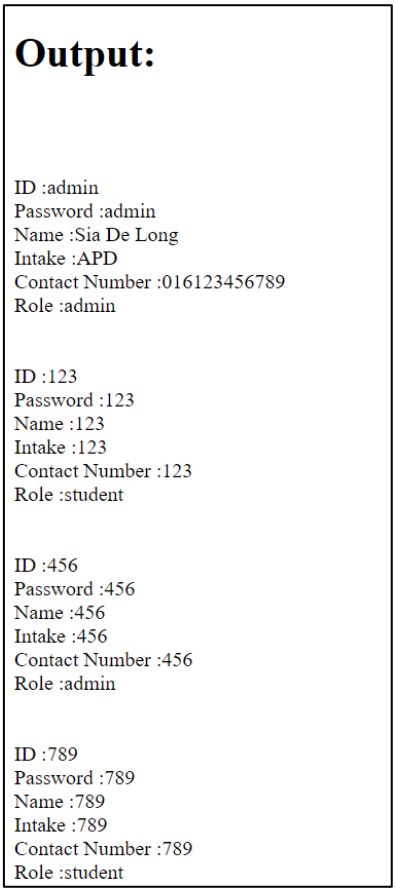
The result will be displaying with the navigation while showing all success contester of the selected election.

View All System Users

Aside from allowing admin to search user with their ID, they can also display all user existing in the system by giving an empty value to the id.



The example output is shown as below:



References

Arora, V. (2021, May 25th). *Planet Crust*. Retrieved from What are the Four Key Enterprise Systems?: https://www.planetcrust.com/what-are-the-four-key-enterprise-systems?utm_campaign=blog

FERNANDO, J. (2022, July 7th). *Investopedia*. Retrieved from Supply Chain Management (SCM): How It Works and Why It Is Important: https://www.investopedia.com/terms/s/scm.asp#:~:text=Supply%20chain%20management%20(SCM)%20is,consumer%20faster%20and%20more%20efficiently.

GeeksForGeeks. (2022, Nov 22nd). *GeeksForGeeks*. Retrieved from Evolution of Distributed Computing Systems: https://www.geeksforgeeks.org/evolution-of-distributed-computing-systems/

IBM. (2022, January 26th). *IBM*. Retrieved from Architectural characteristics of web-based applications: https://www.ibm.com/docs/en/db2-for-zos/11?topic=environment-architectural-characteristics-web-based-applications

IBM Cloud Education. (2020, October 28th). *IBM*. Retrieved from Three-Tier Architecture: https://www.ibm.com/cloud/learn/three-tier-architecture

Medium. (2019, February 14th). *Medium*. Retrieved from What Is Service-Oriented Architecture?: https://medium.com/@SoftwareDevelopmentCommunity/what-is-service-oriented-architecture-fa894d11a7ec

MyReadingRoom. (n.d.). *MyReadingRoom*. Retrieved from E-Commerce Architecture: http://www.myreadingroom.co.in/notes-and-studymaterial/66-e-commerce/517-e-commerce-architecture.html

Techopedia . (2017, January 23rd). *Techopedia* . Retrieved from Enterprise Application (EA): https://www.techopedia.com/definition/24804/enterprise-application-ea

Techopedia. (n.d.). *Techopedia*. Retrieved from One-Tier Architecture: https://www.techopedia.com/definition/17374/one-tier-architecture