Initial Design Description

For

Online National Election Voting

Group: iTeam4

Emilbek Joldoshev	1592476
Hassan Salahe Matar	1591114
Mehmet Barış Özkan	1560747
Hüseyin Lutin	1560408

27/12/2010

1. Introduction

This document describes the initial design strategies and structural properties of the Online National Election Voting System which will be developed by iTeam4. It explains the data and interface designs of the project with system architecture in order to help the developers for better design.

1.1. Problem Definition

We are living in a democratic country and voting is one of the fundamental duties of the public. In our country, manual voting system has been deployed for many years. However, manual voting process has caused some difficulties for voting process and also it has some disadvantages for the public. We can list some of these problems as follows. [1]

- Especially there have been cases of threatening in Eastern part of Turkey at polling stations and people are faced with problems during voting.
- Sometimes people may not be in village/count registration and because of that reason they don't fulfill their voting duties.
- Lots of time and problems are occurring on vote counting process since this activity is done manually.
- Due to manual voting process there is lots of paper waste during election times.
- Voter usually doesn't know too much detail about the candidates in their election region.

With the growth and expansion in technology new ways were sought to handle the electoral process such as electronic voting. Electronic voting is the process of use of computers or other electronic devices to cast votes in an election.

So in order to overcome those problems there is a need of a contemporary electronic voting system in addition to manual voting. By design of such a system people can use their votes in any selection field condition to be registered to the system before. Also by using the system voters can learn details about the candidates and they will be interacting with each other before the Election Day. This system will also facilitate the vote counting processes and produce more accurate results and within a short time thanks to the computer technology. Because of these reasons such an electronic voting system contributes to the development of the country's democracy too much.

1.2. Purpose

The purpose of the document is to make the data design and system architecture of the Online National Election Voting System easy to comprehend. It also serves the purpose of making the functionality clear to system designers.

1.3. Scope

This initial design document applies to the initial version (release 1.0) of the “Online National Election Voting System” software package. It describes the database tables, entity relations between objects and architectural structure of the system as noted in SRS document. The main aim of the system is to provide a set of protocols that allow voters to cast ballots while a group of authorities collect votes and output final results.

1.4. Overview

The remainder of this document identifies the system overview, design considerations, data design with class and table structures, system architecture with components and user interface designs. Apart from these main parts, it also states the planning strategies of the project with Gantt diagrams and describes the tools that will be used during implementation.

1.5. Definitions and Abbreviations

The following is a list of terms, acronyms and abbreviations used by the Online National Election Voting System software package and related documentation.

ABREVETIONS	DEFINITIONS
ONEV	Online National Election Voting
EC	Election Candidate
ECA	Election Commission Authority
ESS	Election Station Supervisor
VIN	Voter Identity Number
DB	Database
TCK	TC Kimlik No
VIC	Voter Identity Card
YSK	Yüksek Seçim Kurulu

For the simplicity of documentation throughout the paper we have used masculinity for all genders.

1.6. References

[1] http://www.yazilimakademisi.org/2011/detailproject.php?id=25

[2] *SRS report for ONEVS*, iTeam4, 2010, www.ceng.metu.edu.tr/~e1591114/SRS

[3] http://www.w3schools.com/html/html_forms.asp

[4] http://experiencezen.com/wp-content/uploads/2007/04/adaptive-path-ajax-a-new-approach-to-web-applications1.pdf

[5] http://www.redbooks.ibm.com/redbooks/pdfs/sg246316.pdf

[6]Aneesha Bakharia, (2001), Java Servlet Pages, Prima Tech.

[7]Simon Bernett, Steve McRobb, Ray Farmer, (1996), Object Oriented System Analysis and Design Using UML

2. System Overview

There are different types of electronic voting systems such as Punch Card Voting System, Telephone Voting and Online Voting which are being used globally at the current period. Due to the impact of the internet the system will be based on online voting type.

Online voting is a form of voting in which the individuals are able to cast their votes through a web interface. Through the use of online voting, the voter navigates to the designated election site using a web browser on an ordinary PC. The voter is then permitted to select their chosen candidate and then cast the votes which would then be sent to the election server for processing. There three main types of online voting as stated above:

Kiosk Internet Voting: Voting from computers in kiosks set up by voting authority in locations such as post offices and shopping malls.

Poll Site Internet Voting: Voting from designated polling sites to cast their votes by using web interface.

Remote Internet Voting: Voting from any from any location through the use of a computer connected to the internet. Remote voting is typically carried out at the voter's home or work place.

Due to political conditions of our country the ONEV system will be designed as two main parts namely Normal Interactive Mode and Election Mode and the voting process will be executed only at polling stations.

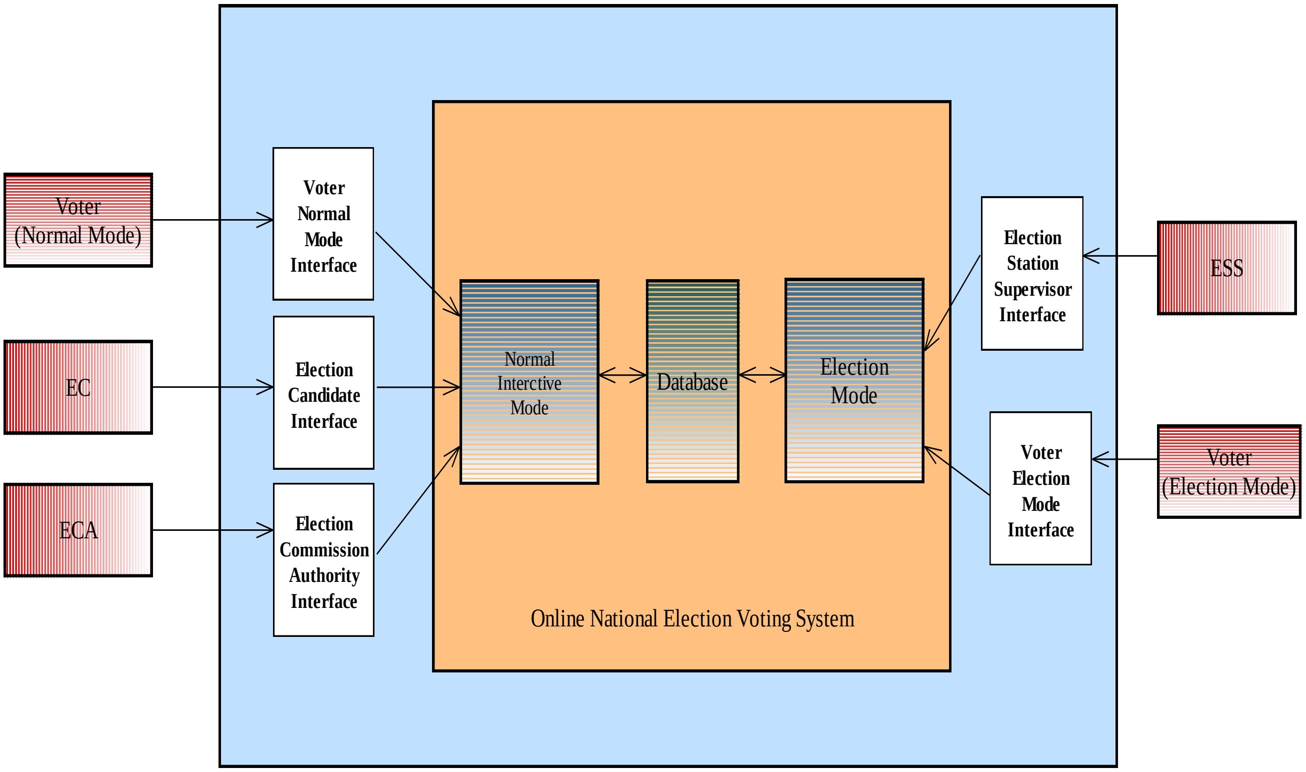
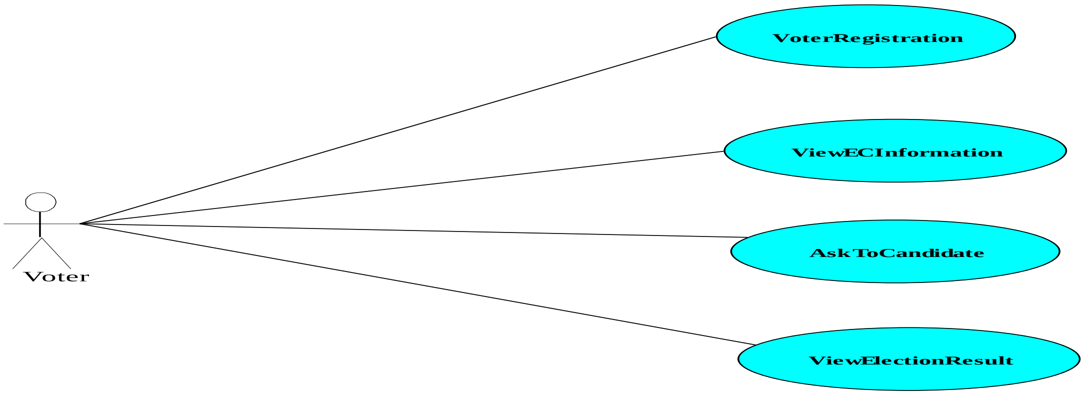

Voter
(Normal Mode)
EC
ECA
Voter Normal Mode Interface
Election Candidate Interface
Election Commission Authority Interface
Normal Interctive Mode
Database
Election Mode
Election Station Supervisor Interface
Voter Election Mode Interface
ESS
Voter
(Election Mode)
Online National Election Voting System

Figure 1: Block diagram showing interaction between users and the system

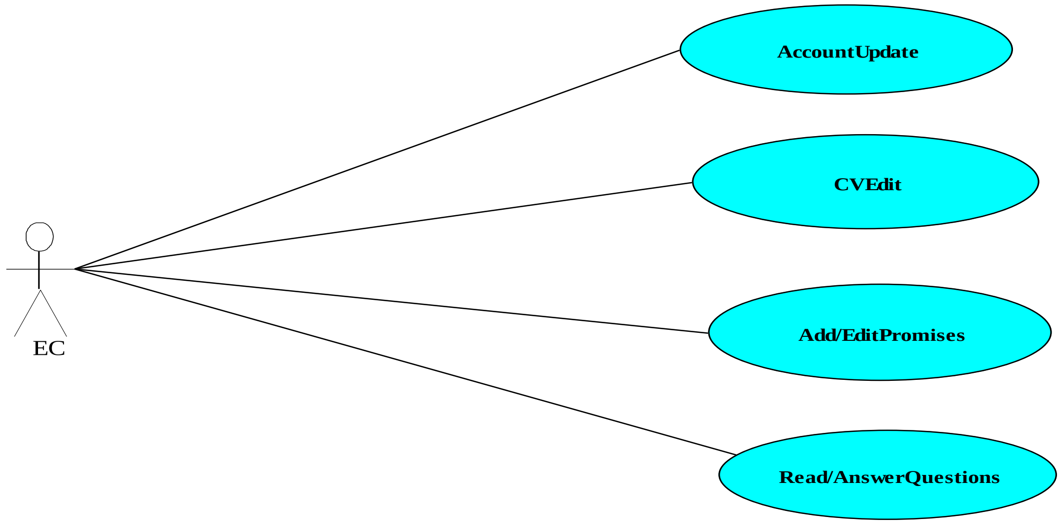
Normal Interactive mode will be used by Voters, ECs, and ECAs for the pre-election and ordinary activities. For every stakeholder there will be a web interface that he can use the system functionalities that are described in the SRS report.

In Normal Interactive Mode,

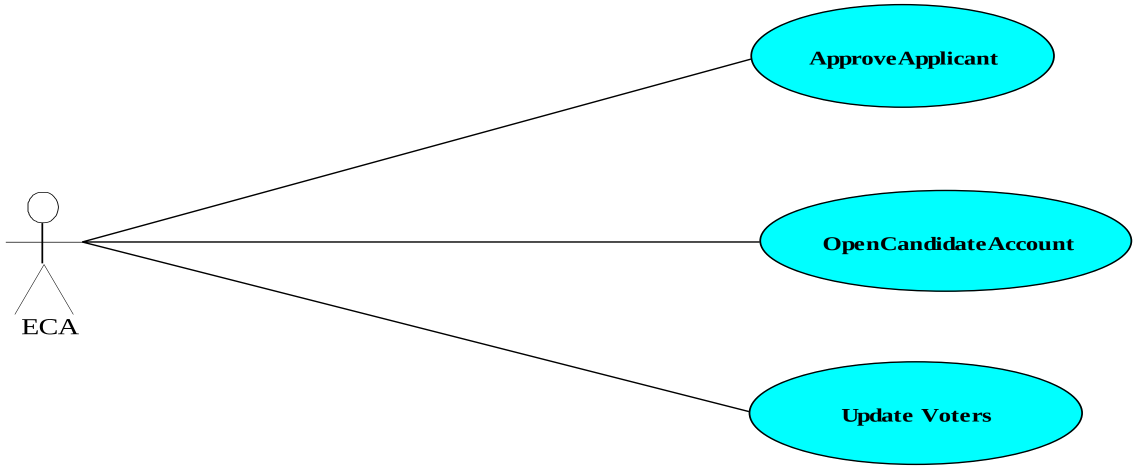
Voters will be able to register to system, see the details of the ECs, ask questions to ECs about their election campaigns and view the past years' election results.



ECs will be able to update their accounts, edit their CVs, add promises about their election campaigns and answer the questions from the voters.

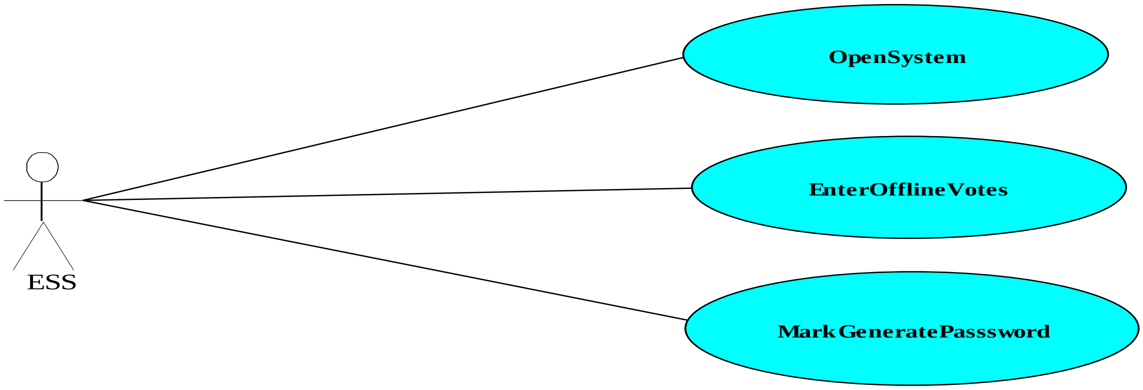


ECA s will be able to approve the applications from the voters, update current voters and open candidate account.



In Election Mode,

The main users of the system are ESSs and Voters. Voters will cast their votes at polling stations with their user id's and passwords. By using the Election Mode, the ESSs will be able to open the system, enter the offline votes to the system and generate hash password that will be used by voters during the voting process [2].



Such a system will provide more contemporary election activities not only for voters but also for election candidates. It will provide the voters to cast their vote from any polling station in case he is not in his election region. Also it provides candidates to conduct their election campaigns through web environment and describe themselves to the voters more clearly.

3. Design Considerations

3.1. Design Assumptions, Dependencies and Constraints

In Turkey, people cast their votes nearly in 170,000 ballots from 81 different cities. Due to this fact the system must work on those ballots at the same time. Since the system divided into two parts, time constraints are different for these parts. In Normal Interactive Mode, the system is expected to serve up to 50000 voters instantly and each voter may be active for a long time. Similarly in Election Mode, the system is expected to serve a maximum of up to 50000 voters however each voter may be active for at 5 minutes for voting operation.

Since the ONEV is a safety critical system, security and safety constraints are the main issues of the system. The system should provide means for protecting and securing recounts of ballots cast in election. By using SSL technologies the data transaction between client and server will be encrypted and all the passwords will be stored in database in an encrypted form. A random word will be generated by the system to prevent attacks and the system will ask the user to enter it correctly for multiple trying.

For performance constraints the system will response in a reasonable sort time. The voter should be able to login and should be able to get response in 2-3 seconds. In Election Mode, the system will handle about 2000 transactions each second and it will be working at 100% peak efficiency during voting process.

Apart from these constraints the system should satisfy the some assumptions and dependencies such as a working internet connection, a web server Java installed on the machine with Java's cryptographic packages. Also the election server will run on a http server that JSP is enabled.

3.2. Design Goals and Guidelines

Since our system is safety critical system, in design of system architecture and database we have to take security principles into account. Since the system will work on web services, it must prevent all attacks from the outside and only authorized people must access the database. It must prevent the manipulation of the votes from unauthorized people.

Also another major principle that the system must provide is reliability. People must rely on the system and they must use the system in confidence. The system must not keep information about which voter cast to which party during execution. The main function of the system must be correct and fast calculation of the votes and results.

For interface designs we have to follow KISS principle. Because for voting operation, every voter has different technological and educational background so the interfaces must be clear to every user. For voting task the voter will only use a radio button to selection operation and a submit button to casting operation. The other interfaces will also designed clearly and simple to all stakeholders.

4. Data Design

4.1. ER Design

The poll server runs on http server that is enabled to handle server pages. It uses a relational database to keep track of the polls, which it connects through standard database connectivity interfaces. In order to run the setup software, the environment needs to have a Java Virtual Machine running on it.

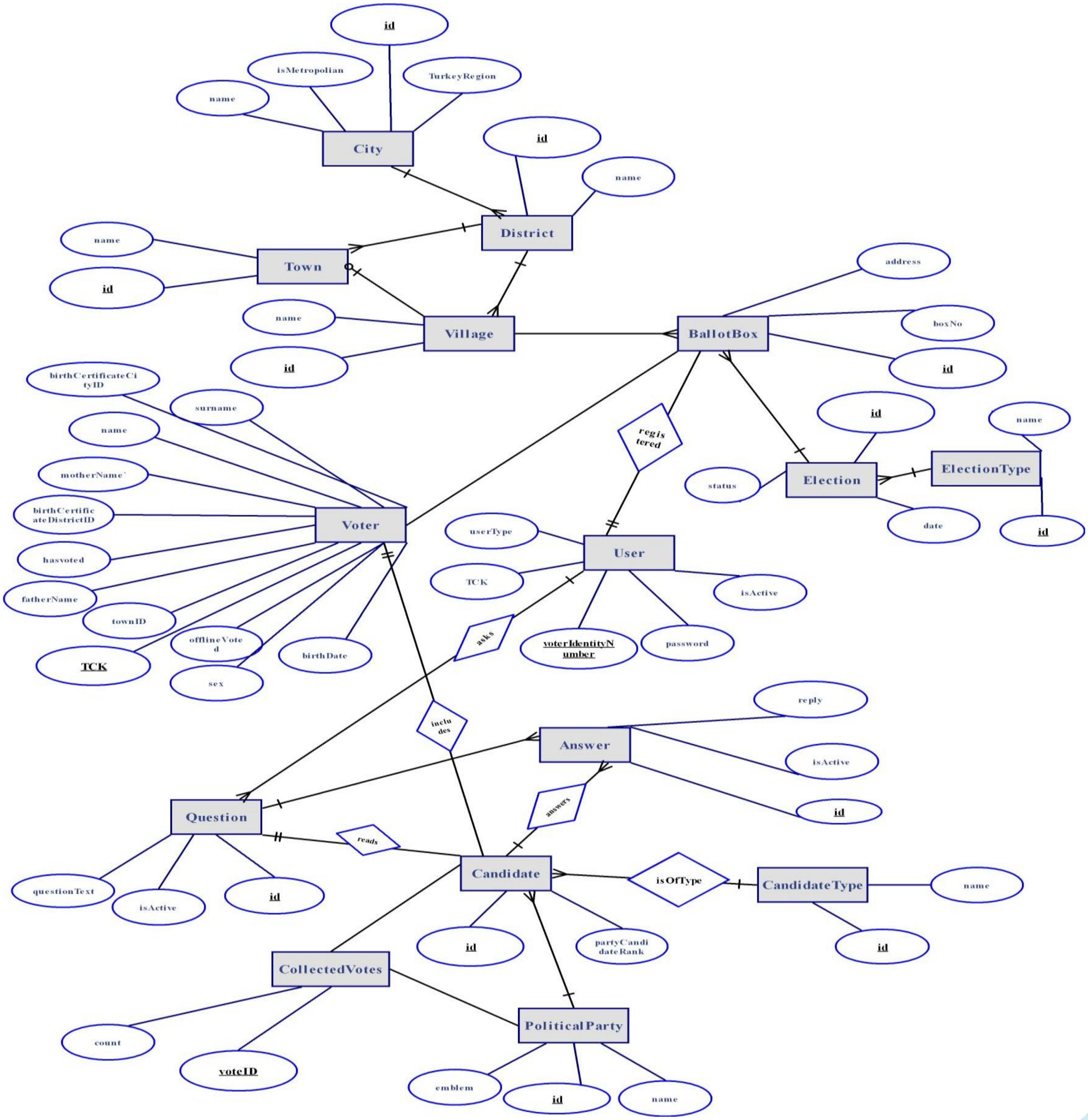


Figure 2: ER Diagram

4.2. Data Schemas

To keep information of some data's location information we designed following tables. Turkey is divided into Cities (İl), Cities are composed of Districts (İlçe), and Districts are composed of both Towns (Belde) and Villages (Mahalle, Köy). Towns are the set of Villages. One exception is: villages do not need to be bound to towns. Some villages are directly bound to districts.

4.2.1. City Table

Field	Type	Null	Foreign Key	References
id(P.K.)	Integer	No	No	-
name	Nvarchar(50)	No	No	-
isMetropolian	Boolean	No	No	-
TurkeyRegion	Integer	No	No	-

City table holds basic attributes of item city. Its primary key is id.

4.2.2. District Table

Field	Type	Null	Foreign Key	References
id(P.K.)	Integer	No	No	-
name	Nvarchar(50)	No	No	-
cityID	Integer	No	Yes	City

This table holds attribute 'name' to keep the name of the district. Its primary key is id. And it also includes cityID as a foreigh key, so we can understand to which city it is bound.

4.2.3. Town Table

Field	Type	Null	Foreign Key	References
id(P.K.)	Integer	No	No	-
name	Nvarchar(50)	No	No	-
cityID	Integer	No	Yes	City
districtID	Integer	No	Yes	District

Town table holds information about towns.Its primary key is id. We could only give districtID as a foreign key and avoid giving cityID as a foreign key. The main reason is, most often we want to know information of towns or villages of some specific city. To, avoid additional query execution we designed as shown above.

4.2.4. Village Table

Field	Type	Null	Foreign Key	References
id(P.K.)	Integer	No	No	-
name	Nvarchar(50)	No	No	-
cityID	Integer	No	Yes	City
districtID	Integer	No	Yes	District
townID	Integer	Yes	Yes	Town

Village table also holds informations such as its name, city, district and town.
Its primary key is id.

4.2.5. Voter Table

Field	Type	Null	Foreign Key	References
TCK(P.K.)	Nchar(11)	No	No	-
Name	Nvarchar(50)	No	No	-
Surname	Nvarchar(50)	No	No	-
motherName	Nvarchar(50)	Yes	No	-
fatherName	Nvarchar(50)	Yes	No	-
Sex	Integer	No	No	-
Birthday	Date	No	No	-
cityID	Integer	No	Yes	City
districtID	Integer	No	Yes	District
townID	Integer	Yes	Yes	Town
villageID	Integer	No	Yes	Village
birthCertificateCityID	Integer	No	Yes	City
birthCertificateDistrictID	Integer	No	Yes	District
boxID	Integer	No	Yes	BallotBox
hasVoted	Boolean	No	No	-
hasOfflineVoted	Boolean	Yes	No	-

Voter table holds information about official voters such as their registered address, where they born, name, surname, sex, birthday, sex, etc. Its primary key is TCK (TC Kimlik No).
It also includes boxID as a foreign key to BallotBox to keep information in which station he uses his vote. 'hasVoted' attribute is used to know whether voter has voted or not.
'hasOfflineVoted' keeps information if voter has voted 'Offline' – with paper.
If 'hasOfflineVoted' is false, it means that the voter used ONEV system and voted 'Online'.

Below, the tables related to Election are described.

4.2.6. Election Table

Field	Type	Null	Foreign Key	References
id(P.K.)	Integer	No	No	-
electionType	Integer	No	Yes	ElectionType
date	Date	No	No	-
isActive	Boolean	No	No	-

Since our system should hold past elections' results, we must have election table to hold results for every election. Users can see filtered results of any past election.
We keep electionType, date and isActive to determine type of the election, date it occurred and if it is active or not.

4.2.7. ElectionType Table

Field	Type	Null	Foreign Key	References
id(P.K.)	Integer	No	No	-
Name	Integer	No	No	-

Our system can handle every kind of election. Now, there are four types of election in Turkey. These are: Genel Seçim, Yerel Seçim, Cumhurbaşkanlığı Seçimi and Referandum. The voting behavior is different for every type of election.

4.2.8. BallotBox Table

Field	Type	Null	Foreign Key	References
id(P.K.)	Integer	No	No	-
boxNo	Integer	No	No	-
electionID	Integer	No	Yes	Election
cityID	Integer	No	Yes	City
districtID	Integer	No	Yes	District
townID	Integer	Yes	Yes	Town
villageID	Integer	No	Yes	Village
Address	Text	No	No	-

This table is to hold information about Election Centers (Sandık).
'boxNo' is numbering of boxes. But this numbering is particular to every city. Because of this, we haven't marked it as a primary key. The box's place information is can be found by its city, district, town and village.

4.2.9. User Table

Field	Type	Null	Foreign Key	References
voterIdentityNumber(P.K.)	Nchar(15)	No	No	-
Password	Nvarchar(50)	No	No	-
isActive	Boolean	No	No	-
TCK	Nchar(11)	No	Yes	Voter
UserType	Integer	No	Yes	UserType

User table holds information about the registered user of ONEV. It holds basic attributes of the user entity such as voterIdentityNumber, password, and userType. TCK is a foreign key to Voter table, so detailed information of the user is kept in Voter table.
Primary key of the user table is voterIdentityNumber.

4.2.10. UserType Table

Field	Type	Null	Foreign Key	References
id(P.K.)	Integer	No	No	-
Type	Nvarchar(50)	No	No	-

In our system, there is more than one type of users. These are Voter, Candidate and ECA. This table is to hold types of users.

4.2.11. PoliticalParty Table

Field	Type	Null	Foreign Key	References
id(P.K.)	Integer	No	No	-
Name	Nvarchar(50)	No	No	-
Rank	Integer	No	No	-
emblem	Image	No	No	-

This table holds the list of Political Parties.

The ‘rank’ attribute is used to keep the rank of the specific Party among Parties to be showed in ‘Voting Card’ or in our system while voting.

4.2.12. Candidate Table

Field	Type	Null	Foreign Key	References
candidateID(P.K.)	Integer	No	No	-
TCK	Nchar(11)	No	Yes	Voter
candidateType	Integer	No	No	-
partyID	Integer	Yes	Yes	Party
partyRank	Integer	Yes	No	-
electionID	Integer	No	Yes	Election
cityID	Integer	Yes	Yes	City
districtID	Integer	Yes	Yes	District
townID	Integer	Yes	Yes	Town
villageID	Integer	Yes	Yes	Village

This table holds basic information about Candidate.

It has foreign key TCK to Voter, so detailed information can be got from Voter table.

For candidates that are member of a party, its partyID is stored and is a foreign key to Party table.

partyRank is used to show the Candidate’s rank among same party’s candidates in his region.

candidateType is foreign key that is used to show the type of the candidate.

4.2.13. CandidateType Table

Field	Type	Null	Foreign Key	References
id(P.K.)	Integer	No	No	-
name	Integer	No	No	-

CandidateType table is used to hold types of candidates.

4.2.14. CollectedVote Table

Field	Type	Null	Foreign Key	References
voteID(P.K.)	Integer	No	No	-
boxID	Integer	No	Yes	BallotBox
partyID	Integer	Yes	Yes	PoliticalParty
candidateID	Integer	Yes	Yes	Candidate
voteCount	Integer	No	No	-

This table is used to hold information of collected votes of a party or an individual candidate.
boxID is a foreign key to BallotBox, to show from which box the result is.
partyID is to show which party's result this is.
candidateID is to show which candidate's result this is.

4.2.15. Question Table

Field	Type	Null	Foreign Key	References
id(P.K.)	Integer	No	No	-
userID	Integer	No	Yes	User
candidateID	Integer	No	Yes	Candidate
questionText	Text	No	No	-
isActive	Text	No	No	-

In our system, users can ask questions to candidate. This table is used for that aim.

4.2.16. Answer Table

Field	Type	Null	Foreign Key	References
id(P.K.)	Integer	No	No	-
questionID	Integer	No	Yes	Question
candidateID	Integer	No	Yes	Candidate
reply	Text	No	No	-
isActive	Boolean	No	No	-

This table holds answers to question. There can be more than one answer for a question. So, id is a primary key, not questionID.

5. System Architecture

5.1. Architectural Design

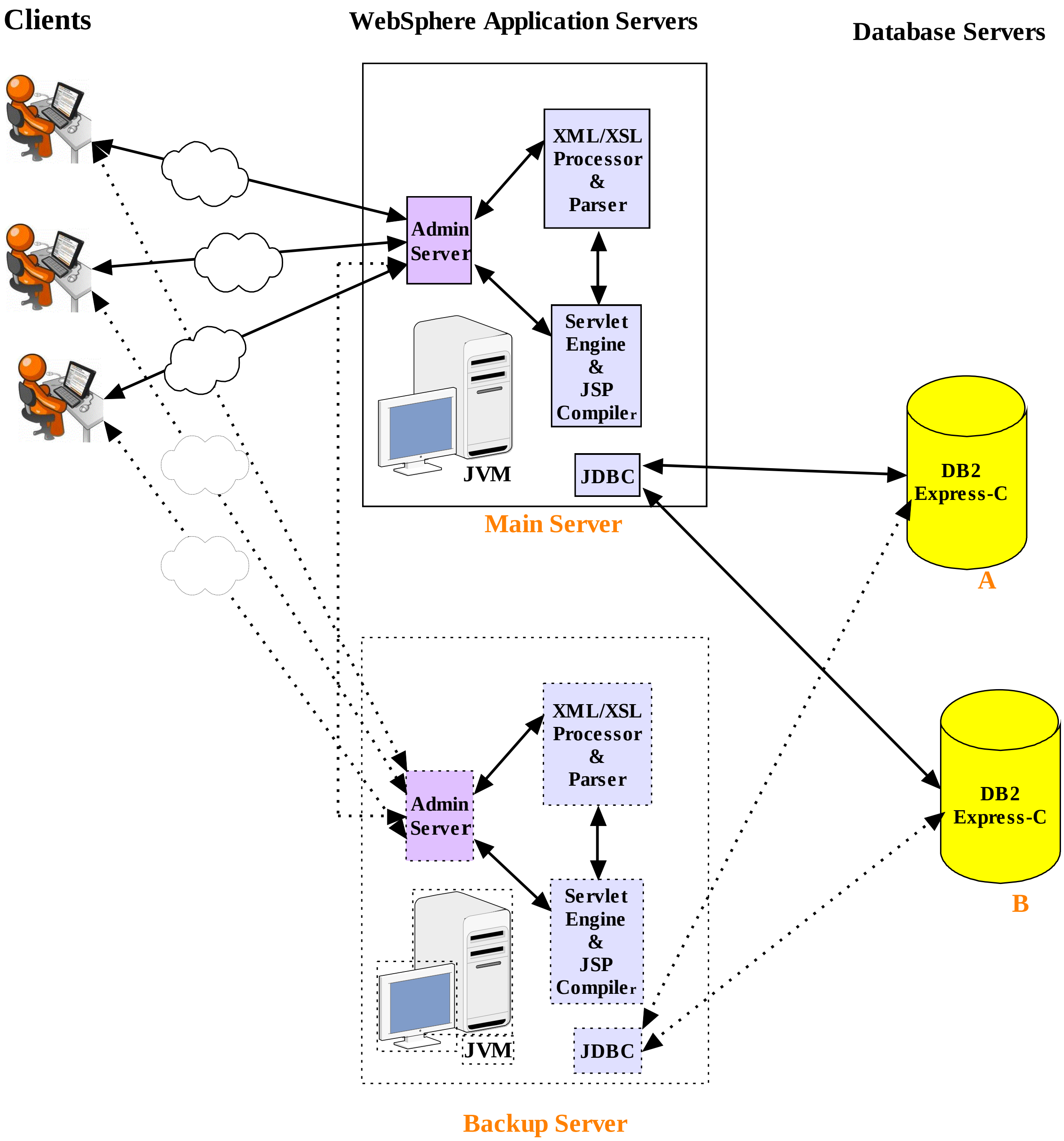

Clients
WebSphere Application Servers
Database Servers
XML/XSL Processor & Parser
Admin Server
Servlet Engine & JSP Compiler
JVM
JDBC
Main Server
DB2 Express-C
A
XML/XSL Processor & Parser
Admin Server
Servlet Engine & JSP Compiler
JVM
JDBC
Backup Server
DB2 Express-C
B

Figure 3: General View Of ONEV System

Basically, Our system is a 3-tier Client/Server architecture comprising of two databases, two Application servers and PC stations. The additional application server presented in dotted lines in the diagram above acts as a backup to the working main server. Therefore, during critical operations, in case of failure the reserve server comes into operation. The two databases work together during critical operations of polling votes. However, the backup server is responsible for storing critical information like votes and results of election. In the front phase of the system architecture lies the clients. The clients represent the PC centers formed throughout the country during election periods. It also represents any PC that can connect to our server during normal working days for regular applications like viewing election results, editing profiles and so on. The middle phase of the architecture comprises of Application servers we have discussed above. It should be noted that the servers consist of back-end applications to handle different tasks delegated by the administration server. The far end phase is comprised of storage subsystems, mainly the databases. These phases communicate in a formal protocol. That is, application server communicates directly with the clients and the storage devices. However, clients-databases communication is not directly. The application server – through a database connector- handles all database requests from the clients side to the database, as well as the responses are controlled by the server.

HIGH ABSTRACT MODULAR SYSTEM STRUCTURE

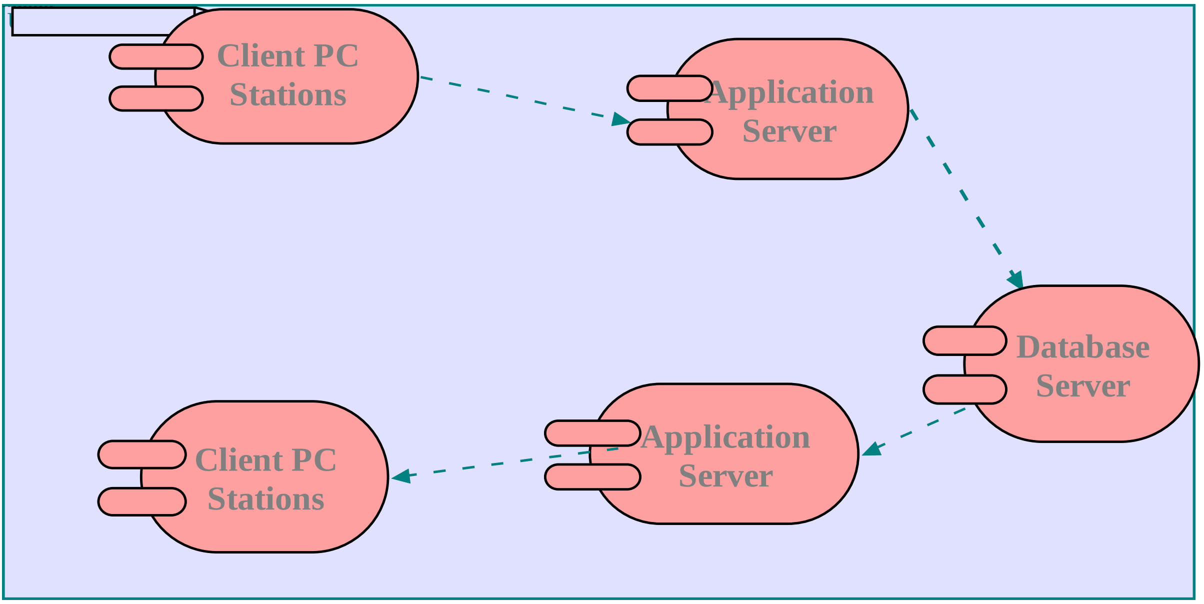


Figure 4: High Abstract Modular System Structure

The major components in the system can be represented in form of modules. Therefore, we have three unique major modules Clients, Application server, and Database server modules. The diagram below shows the application sequence of the modules. The normal flow of actions in the system follows this order. A client issues a communication or data request with the server. The server (in many functions of the systems) checks the validity and eligibility of the client to the system by contacting the data storage server. Upon the response from the database server; the application server responds to the client request with positive or negative acknowledgement. Again, it should be noted that there is no direct communication between the clients and the database server.

5.2. Description of Components[7]

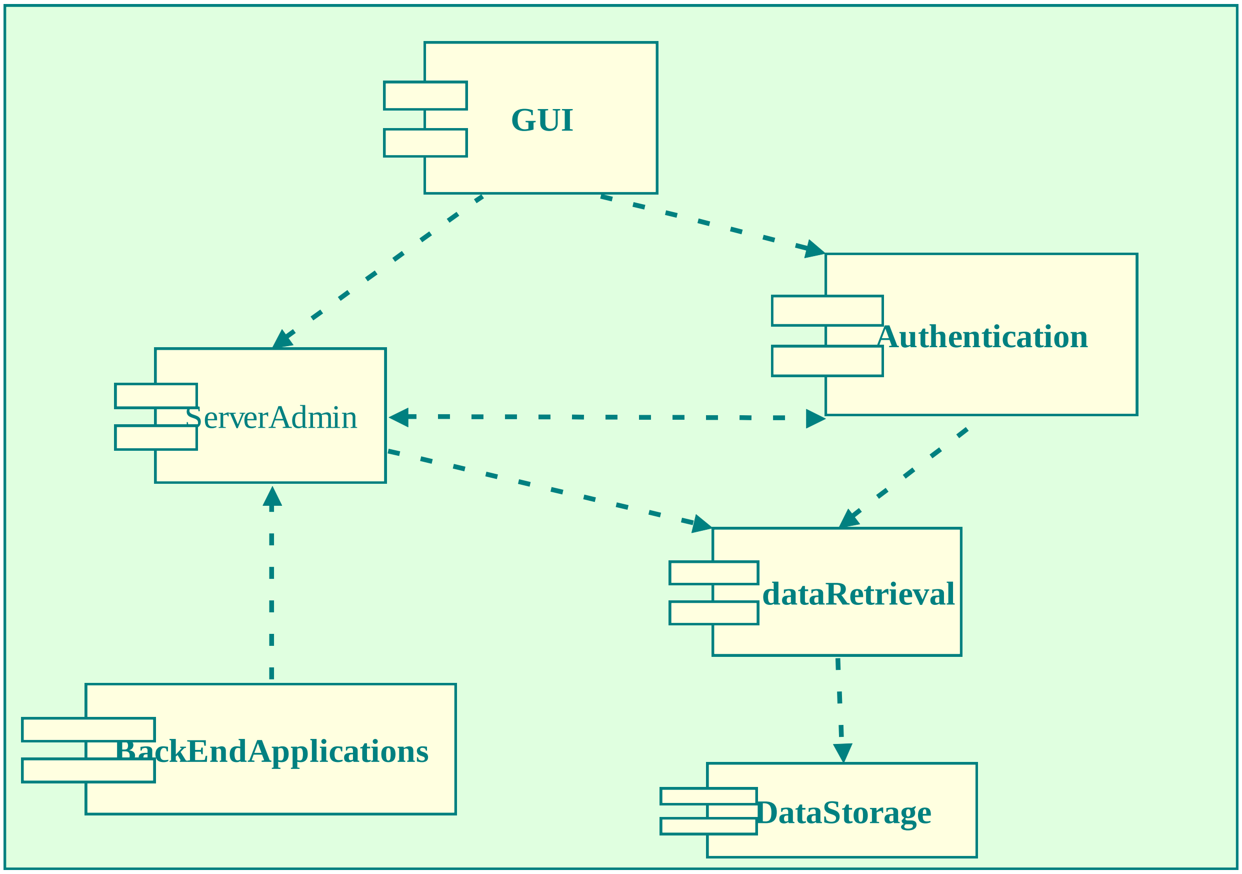


Figure 5: Components of the System

Our system can be subcategorized into six components according to major activities performed by the system. The components are namely; Graphical User Interface(GUI), Server Administrator(ServerAdmin), Authentication, Back End Applications(BackEndApplications), Data Retrieval(dataRetrieval) and data Storage(DataStorage).

5.2.1. Graphical User Interface

5.2.1.1. Processing Narrative for GUI

This component comprises all the objects that render the graphical User Interfaces with the appropriate contents. When a client issues an http request to the application server, a corresponding instance of class is issued by the Java Servlet[6] to respond to and process the request. In addition to that, the component is responsible for creation of dynamic HTML webpage using JSP technology before sending them to the client side.

5.2.1.2. GUI Interface Description

The inputs to this component are the viewable webpage requests from the client side. On the other hand the outputs are the dynamically/statically created webpages to be displayed on the client side.

5.2.1.3. GUI Processing Detail

The complete step-by-step procedural activities related to this component are as follows;

1. User/client requests a page from the system through internet
2. Server Admin captures the request.
3. After processing administration tasks according to the type of request, Server Admin delegates the presentation of solution page(s) to the GUI component to create appropriate internet page.
4. The GUI presents the created page to the Server Admin to send it to the requester.

5.2.2. Data Storage

5.2.2.1. Processing Narrative for Data Storage

This component is responsible for creating and storing data objects. Therefore it makes frequently requested data available instead of querying into the database frequently. It uses the JDBC connector to get data from the database and create corresponding objects with attributes and methods to access the data easily.

5.2.2.2. Data Storage Interface Description

It receives data requests from the dataRetrieval component as an input. Then it translates these into SQL commands and processes them using JDBC connector. The obtained result is put into an object. The object becomes available for future use.

5.2.2.3. Data Storage Processing Detail

It works as follows

1. It receives a request of data from dataRetrieval component
2. It0 issues the command through JDBC connector
3. The received response from the run queries and creates a corresponding object.

5.2.3. ServerAdministrator(ServerAdmin)

5.2.3.1. Processing Narrative for ServerAdministrator

ServerAdmin is “a junction” between requests and responses. It receives HTTP requests from the client side and delegate the requests to respect servlets to process the requests. In addition to that, it collects the ready responses and sends them to the appropriate clients. It works closely with authentication component to authenticate the income requests before delegating them to the corresponding back end applications to process them.

5.2.3.2. ServerAdministrator Interface Description

It receives data packets online in form of HTTP protocols as an input. Using back end programs the packets are processed, the required information is extracted and the necessary steps taken into actions. It outputs HTTP responses and sends them to the clients via the internet.

5.2.3.3. ServerAdministrator Processing Detail

It works as follows

1. It receives a request from clients through HTTP.
2. It checks the validity of the request.
3. According to the type the request it assigns the request to a corresponding back end program.

4. When the request is processed it sends to the corresponding client

5.2.4. Authentication

5.2.4.1. Processing Narrative for Authentication

This component is responsible for checking the critical requests with the permission of the clients. For example if a client tries to log on into the system Authentication checks if he is a registered user of the system according to the user identification and password. This is also the same when user wants to access some data. An election commission officer may be granted to view the voter profile while a voter cannot be granted the access the profile of other voters.

5.2.4.2. Authentication Interface Description

It receives commands as well as data from the ServerAdmin to help authenticate the process in question. The output is either request granted or denied. The output is directed to the ServerAdmin. It interacts with DataRetrieval in order to get data from the data storage component.

5.2.4.3. Authentication Processing Detail

It works as follows

1. It receives authentication request from ServerAdmin along with data.

2. It using the given data and that in the database it processes authentication.

3. It returns a grant or a denial response.

5.2.5. Back End Applications

5.2.5.1. Processing Narrative for Back End Applications

This includes technologies to handle different tasks and instantiate and serve different tasks delegated by ServerAdmin. The technologies involved include XML-parsers, JSP, Servlets, and the JVM.

5.2.5.2. Back End Applications Interface Description

In general the server task can be considered as an input to the back- end server. The output is the result of the back end server according to the requirements of the ServerAdmin.

5.2.5.3. Back End Applications Processing Detail

It works as follows

1. ServerAdmin triggers a job to the appropriate back- end application.
2. ServerAdmin provides appropriate input to the application.
3. The application processes the job
4. The application returns response to the ServerAdmin.

5.2.6. Data Retrieval

5.2.6.1. Processing Narrative for Data Retrieval

This component is responsible for accessing data from and storing data to the database. It acts as a bridge between the applications and the database objects.

It uses the JDBC connector to process the data queries in form of SQL commands.

5.2.6.2. Data Retrieval Interface Description

It receives data requests from the Server admin, authentication and the back end applications. Then it translates these into SQL commands and processes them using JDBC connector. The obtained result is returned as an object. The returned object is extracted to get the required data and reported to the component requested it.

5.2.6.3. Data Retrieval Processing Detail

It works as follows

1. It receives a request of data from Application server components
2. It translates the request into SQL command
3. It issues the command through JDBC connector
4. The received response from the run query is extracted to get the required data
5. The data is sent to the component asked for it.

5.3. Design Rationale

We separated the system into three major modules in order to keep the system simple, minimize cost and increase security level. As it can be seen from the system representation diagram there is much of computational activities rather than just presentation of windows as graphical user interfaces. The presence of one application server minimizes cost in terms of money and the cost of system distribution. All the necessary computations are carried out at the particular center. The presence of backup application server makes sure that the system is available most of the time even in the case the main application server encounters a problem that hinders its functioning.

Data storage is separate because we wanted to separate it completely from direct communication with the clients. Query issuing over the internet can be a threat and sometimes degrades performance. The communication between the application server and the database can be improved by storing the already queried data into the server machine, which we cannot do in the client machine to avoid insecurity.

Before concluding this architecture we had discussed architectures like Single Tier and Two Tier architectures. In Single tier architecture we decided to design an application that runs on a client machine (like a desktop application). However, due to criticality of the system, this cannot be possible because the system can be easily attacked by viruses in the client machine. The 2-tier architecture was totally inappropriate for our system because it requires storage of information in a formatted order for easier access. This is due to the fact that data storage and retrieval is more than 50% of all activities carried out by the system to meet the clients' needs.

6. User Interface Design

6.1. Overview of User Interface

Since the system consists of two parts user interfaces will be different in those two modes. In normal interactive mode there will be common home page interface for all system users and they will use this page for login operation.

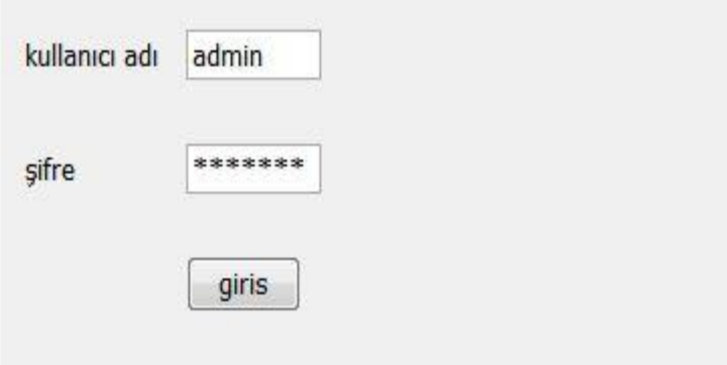
In this mode voters interface will contain the links to view the candidates profiles and past years' election results. EC's interface will include his own profile and he will conduct the election tasks by using this interface. ECA interface will cover the functionalities related with registration of the voters and candidates.

In election mode there will be a major interface that the voting operation is executed. This interface will be used by the voters. And there will be another interface fort the ESS. By using this interface the ESS's will generate a password for the voters used in casting operation and also he can enter the offline votes to the system.

6.2. Screen Images

In this part some of the screen images and their functionalities are described.

6.2.1. Login



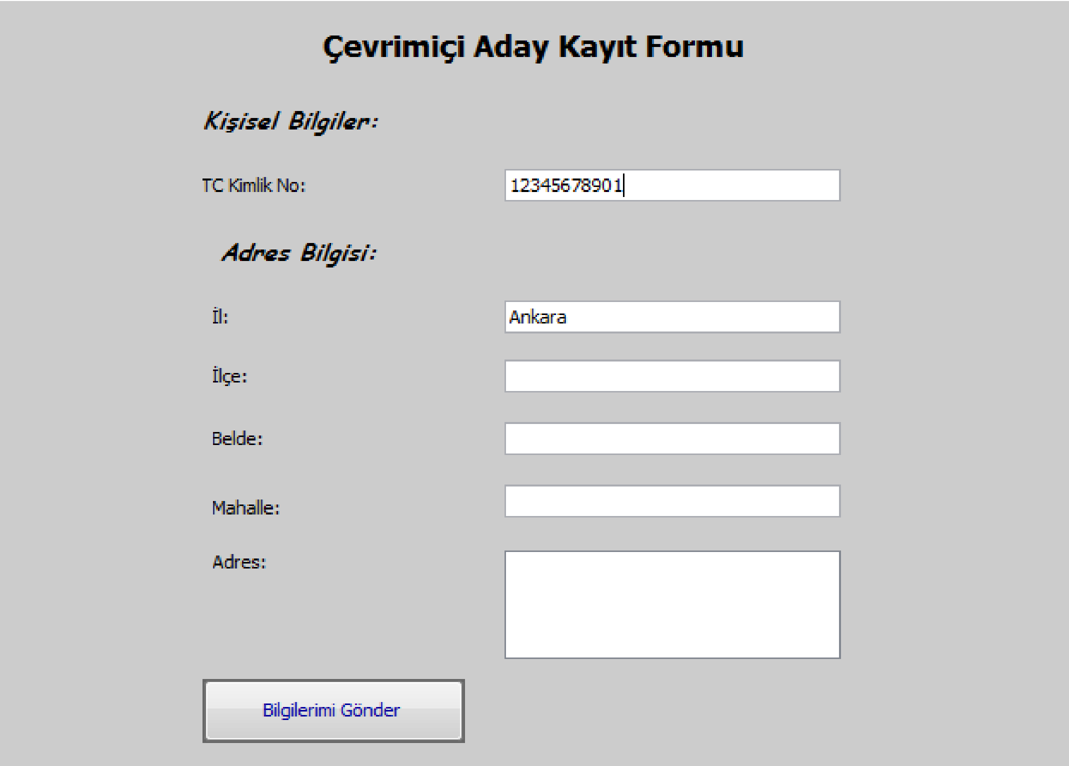
kullanıcı adı admin

şifre *******

Figure 6: User Interface of Login Page

This interface will be used by all of the system users and by entering the userid and password they will be able to use the system. For an incorrect password or userid the system will promote an error message to the users.

6.2.2. User Registration



Çevrimiçi Aday Kayıt Formu

Kişisel Bilgiler:

TC Kimlik No: 12345678901

Adres Bilgisi:

İl: Ankara

İlçe:

Belde:

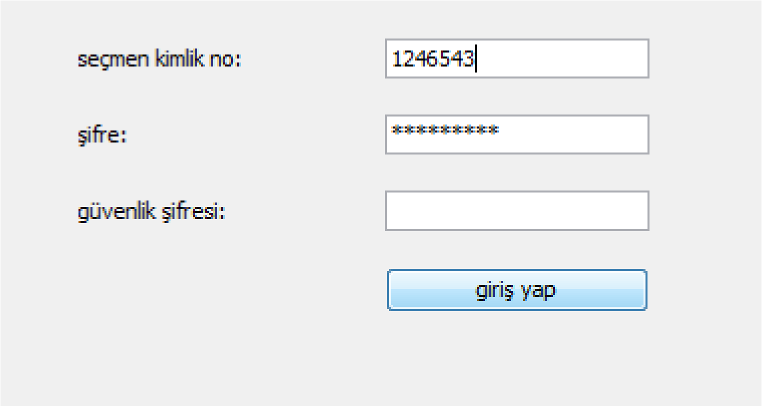
Mahalle:

Adres:

Figure 7: User Interface of Registration Page

This is will be used for the registration of the citizens to this system. We only require TCK of the citizen as personal information. We can get other required personal information such as birthday, sex, father's name, etc from governmental web service by providing only TCK. It will be easier for the user to register. Additionally, citizens must provide their address information. Then the official goes to that address and checks if the citizen is at that address or not. If the citizen is at that address and right to vote, then he will be approved.

6.2.3. Login For Vote


seçmen kimlik no: 1246543

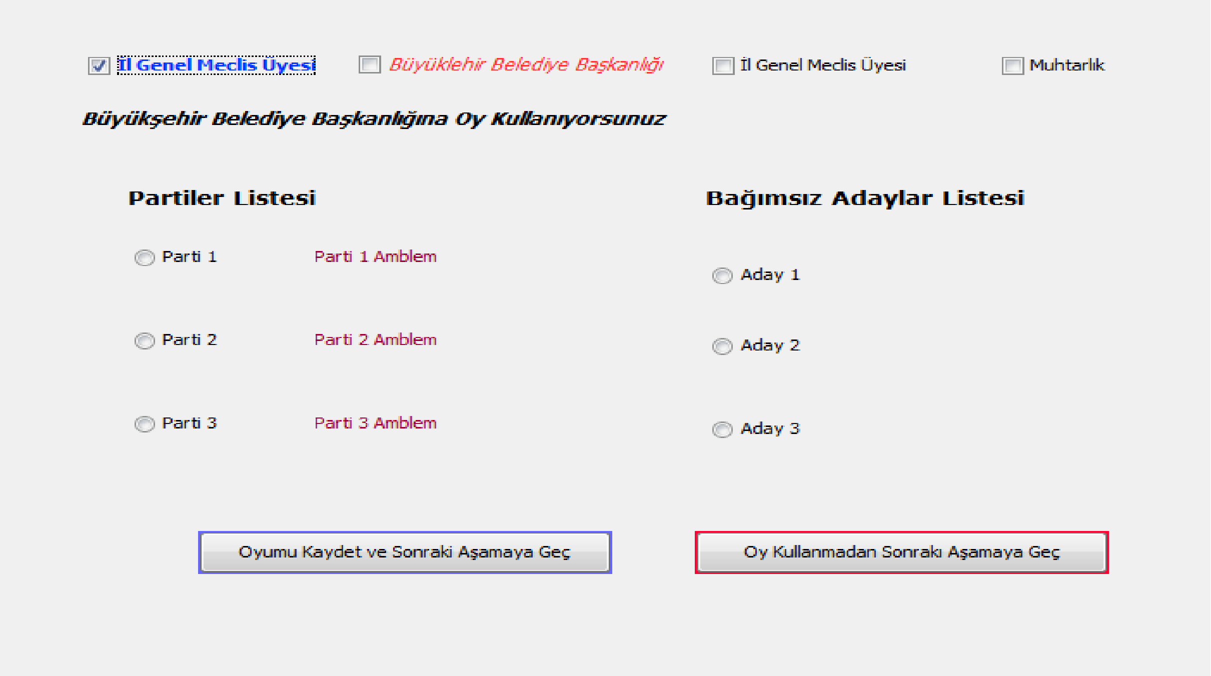
şifre: **********

güvenlik şifresi:

Figure 8: User Interface of Voting Stage Login Page

This interface will be used by the voters during the election mode in voting process. Before casting the vote, the voter must provide his Voter Identity Number, password and security password generated by the ESS. After entering the correct values the voter can reach the voting interface.

6.2.4. Voting


☑ İl Genel Meclis Üyesi □ Büyüklehir Belediye Başkanlığı □ İl Genel Meclis Üyesi □ Muhtarlık

Büyükşehir Belediye Başkanlığına Oy Kullanıyorsunuz

Figure 9: User Interface of Voting Page

After the voter logged in successfully, this interface is used for casting vote. In our system, the user interfaces will be simple and clear since stakeholders of the system have different educational, technological background. The Voter castes for only one candidate type and go to next page for the next type of candidate casting.

6.3. Screen Objects and Actions

Since the users interact with our system through web browser, our objects will be html elements. Some of the main objects and their functionalities are described below:

- **Label**
  The <label> tag defines a label for an input element (Password Field, Text Field).
  In our application, we use labels for every important input element. If the user clicks on the text within the label element, it toggles the input element.
- **Text Field, Password Field**
  When the user fills these fields and sends the form, the server gets filled values and do some transactions and returns results according to given values. We use these objects in order to get required information.
- **Check Box**
  When we want to get only 'Yes – No' or 'True – False' information for the specific question we use check boxes.
- **Radio Button**
  When the user is forced to choose only one option from the list, the radio button is used. The main usage of this object is at voting process. To illustrate, voter chooses only one political party or individual candidate from the list.
- **Submit Button**
  A submit button is used to send form data to a server. The data is sent to the page specified in the form's action attribute.[3]
- **Hyperlink**
  A hyperlink (or link) is a word, group of words, or image that you can click on to jump to a new document or a new section within the current document. Hyperlink's difference from Submit Button is, it does not send any field's values, it's aim is only to redirect to some other page.

7. Libraries and Tools

For system design the following tools will be used during the implementation process.

- UML: The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems

- J2EE: J2EE (Java 2 Platform, Enterprise Edition) is a Java platform designed for the mainframe-scale computing typical of large enterprises. Sun Microsystems (together with industry partners such as IBM) designed J2EE to simplify application development in a thin client tiered environment.

- Ajax: Ajax (sometimes called Asynchronous JavaScript and XML) is a way of programming for the Web that gets rid of the hourglass. Data, content, and design are merged together into a seamless whole.[4]

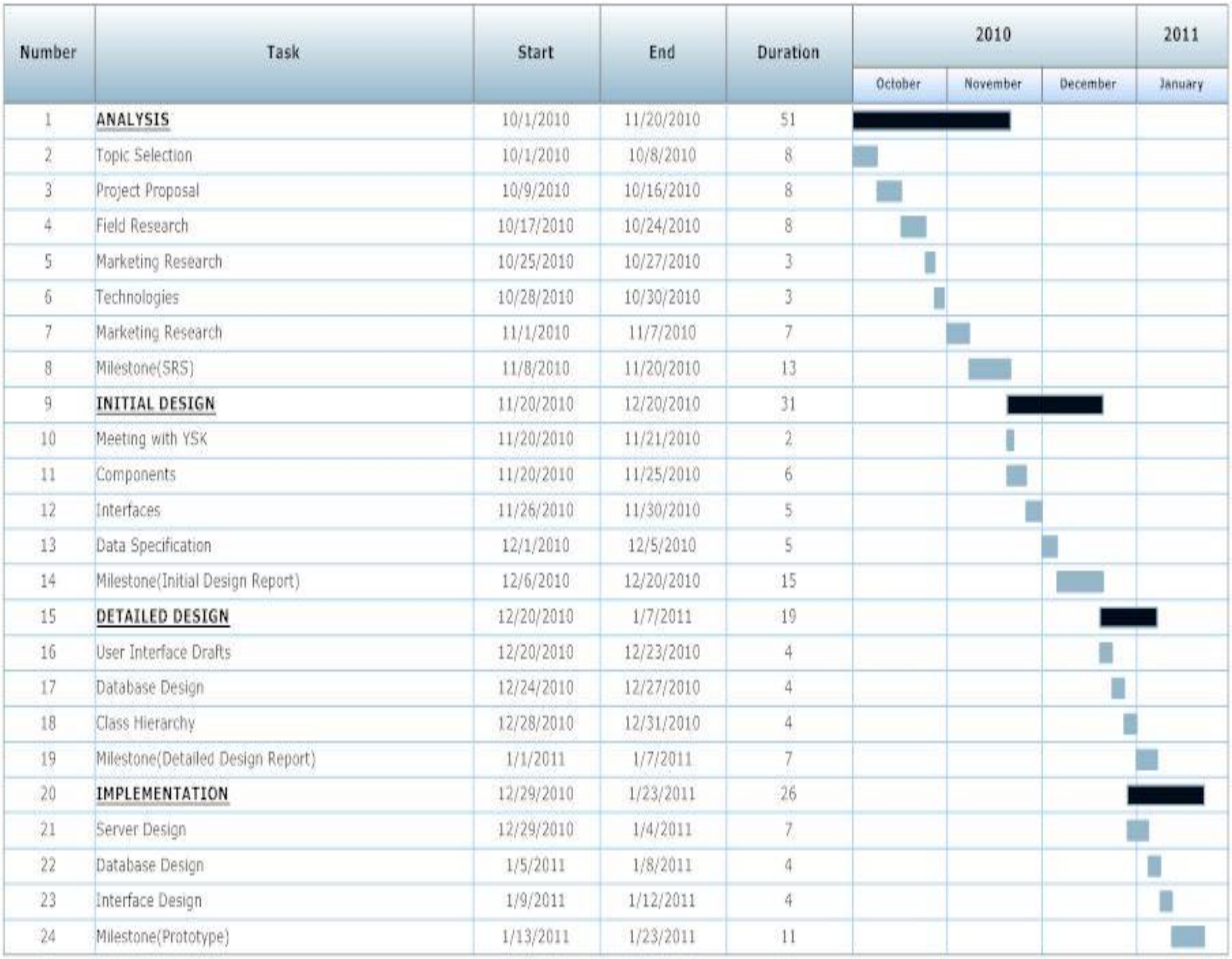
- DB2: DB2 is a family of relational database management system (RDBMS) products from IBM that serve a number of different operating system platforms. According to IBM, DB2 leads in terms of database market share and performance.

- Eclipse: Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. It is written mostly in Java and can be used to develop applications in Java and, by means of various plug-INS, other programming languages.

- WebSphere: WebSphere is a set of Java-based tools from IBM that allows customers to create and manage sophisticated business Web sites. The central WebSphere tool is the WebSphere Application Server (WAS), an application server that a customer can use to connect Web site users with Java applications or servlets.[5]

8. Time Planning (Gantt Chart)

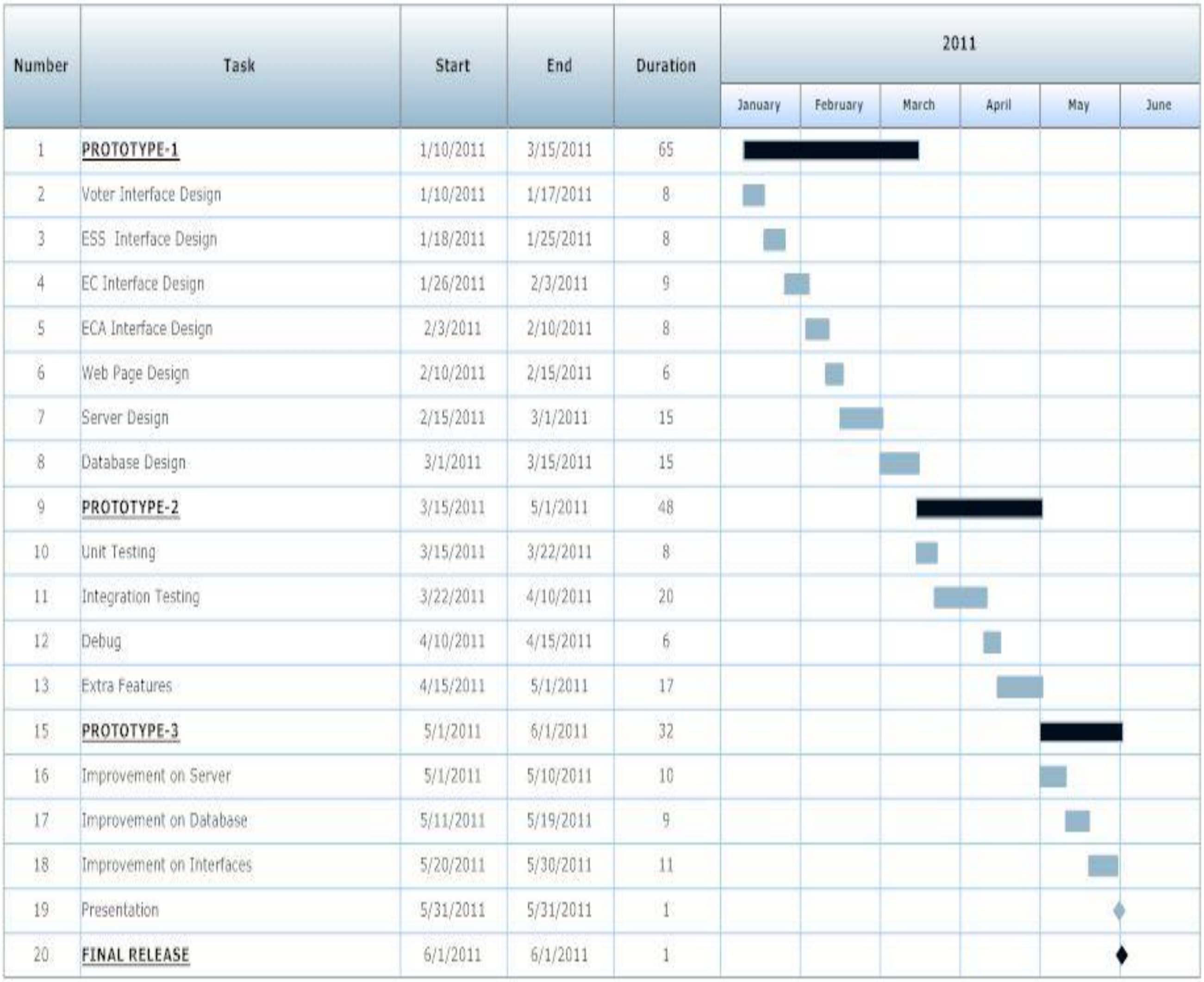
8.1. Term1 Gantt Chart



Number	Task	Start	End	Duration
1	**ANALYSIS**	10/1/2010	11/20/2010	51
2	Topic Selection	10/1/2010	10/8/2010	8
3	Project Proposal	10/9/2010	10/16/2010	8
4	Field Research	10/17/2010	10/24/2010	8
5	Marketing Research	10/25/2010	10/27/2010	3
6	Technologies	10/28/2010	10/30/2010	3
7	Marketing Research	11/1/2010	11/7/2010	7
8	Milestone(SRS)	11/8/2010	11/20/2010	13
9	**INITIAL DESIGN**	11/20/2010	12/20/2010	31
10	Meeting with YSK	11/20/2010	11/21/2010	2
11	Components	11/20/2010	11/25/2010	6
12	Interfaces	11/26/2010	11/30/2010	5
13	Data Specification	12/1/2010	12/5/2010	5
14	Milestone(Initial Design Report)	12/6/2010	12/20/2010	15
15	**DETAILED DESIGN**	12/20/2010	1/7/2011	19
16	User Interface Drafts	12/20/2010	12/23/2010	4
17	Database Design	12/24/2010	12/27/2010	4
18	Class Hierarchy	12/28/2010	12/31/2010	4
19	Milestone(Detailed Design Report)	1/1/2011	1/7/2011	7
20	**IMPLEMENTATION**	12/29/2010	1/23/2011	26
21	Server Design	12/29/2010	1/4/2011	7
22	Database Design	1/5/2011	1/8/2011	4
23	Interface Design	1/9/2011	1/12/2011	4
24	Milestone(Prototype)	1/13/2011	1/23/2011	11

Figure 10: Gantt Chart of Term1

8.2. Term2 Gantt Chart



Number	Task	Start	End	Duration
1	**PROTOTYPE-1**	1/10/2011	3/15/2011	65
2	Voter Interface Design	1/10/2011	1/17/2011	8
3	ESS Interface Design	1/18/2011	1/25/2011	8
4	EC Interface Design	1/26/2011	2/3/2011	9
5	ECA Interface Design	2/3/2011	2/10/2011	8
6	Web Page Design	2/10/2011	2/15/2011	6
7	Server Design	2/15/2011	3/1/2011	15
8	Database Design	3/1/2011	3/15/2011	15
9	**PROTOTYPE-2**	3/15/2011	5/1/2011	48
10	Unit Testing	3/15/2011	3/22/2011	8
11	Integration Testing	3/22/2011	4/10/2011	20
12	Debug	4/10/2011	4/15/2011	6
13	Extra Features	4/15/2011	5/1/2011	17
15	**PROTOTYPE-3**	5/1/2011	6/1/2011	32
16	Improvement on Server	5/1/2011	5/10/2011	10
17	Improvement on Database	5/11/2011	5/19/2011	9
18	Improvement on Interfaces	5/20/2011	5/30/2011	11
19	Presentation	5/31/2011	5/31/2011	1
20	**FINAL RELEASE**	6/1/2011	6/1/2011	1

Figure 11: Gantt Chart of Term2

9. Conclusion

This document describes the design levels of the ONEV project conducted by iTeam4. The system architecture of the ONEV and data representations is stated through the document. Furthermore, class diagrams with data flow diagrams and design of the user interfaces are showed in the document in detail. Consequently, this document is prepared to conduct better design approaches to ONEV project at implementation.