Online Voting(I-Voted)

System Design Document

v0.1

02.12.2018

Şuayb Talha Özçelik

Berkay Günay

Tuncer Oğuzkurt

Ögeday Sunar

Prepared for

SE301 Software Engineering



Table of Contents

1. Introduction 1

1.1. Purpose of the System 1

1.2. Design Goals 1

1.3. Definitions, Acronyms, and Abbreviations 1

1.4. References 1

2. Current Software Architecture 1

3. Proposed Software Architecture 1

3.1. Overview 1

3.2. System Decomposition 1

3.3. Hardware Software Mapping 2

3.4. Persistent Data Management 2

3.5. Access Control and Security 2

3.6. Global Software Control 2

3.7. Boundary Conditions 2

4. Subsystem Services 2

5. References 2

SYSTEM DESIGN DOCUMENT

# Introduction

Design is a work process which has a user perspective and drives development based on your specific customers’ needs. Methods and approaches differ depending on what you are developing. In design part, it offers us both a guide to the customer and a guide to the development process for us. After design part we are able to reduce the time required the implementation.

## Purpose of the System

As we mentioned in Requirement Analysis Document, the aim of the I-voted application provides easy, credible and fair elections for users who want to use a vote to choose a person with the highest number of voters there by declaring the person the winner. The system allows users to vote from within a few clicks and also offers many features such as adding elections, deleting elections which they added, and seeing the elections which they voted.

## Design Goals

The design objectives signify the specified qualities of online voting(I-voted) system and supply a constant set of standards that should be thought of when making design choices. Based on non-functionality requirements the next design targets must achieved as a way to qualify the system as profitable:

* **Security**

We use for save users’ informations, elections information, results, etc. in Firebase database. Firebase is a Google company. When we set non-security settings, Firebase warns us like “These settings are no safe”. Firebase is a security platform for mobile application. Also, we set the rules of databese in Firebase settings which kind of users can write or read data in Firebase. We can set these kind of rules in Firebase.

* **Reliability**

We keep the users vote with users’ id in the elections result table in database. We will Show the users vote in the menu which elections they are voted. If they think that there is a problem, they can complain. We will show the elections result when the elections finish(end date). Also, each voter can vote a election just one time.

* **End User**

All users can vote elections, if they dowload the application.However, Anonym Users can not vote the election which is created by registered users.When they are voting, their phone has to be connected the ethernet.Our application is for Android devices which have Android 5.0 (Lolipop) or greater version of Android.

* **Performance**

Our system can sturdy enough to manage any valid input from the users. Our system can support 100 users to write and read data in firebase database at the same times.Our system has to show the elections results with 2-3 seconds.

Moreover, the other goals of our design I-voted system should accept upgrades and should be implemented on Android Studio platform. The other thing is we have to design our app with easily understandable and useful. Also, the colors, pictures and icons in our app have to be connected each other. Our screen designs have xml codes, our functions have java codes. Our database is Firebase database. User interface should be mobile based.

## Definitions, Acronyms, and Abbreviations

**Anonym-User:** The system user who wants to vote for some elections.

**Registered-User:** The system user who wants to vote for any elections.

**Admin:** The system administrator who will manage all data system data and user controls.

**RAD:** Requirement Analysis Design

**SDD:** System Design Document

**Firebase:** Firebase is a Backend as a Service

**Android Studio:** A platform for Android application development

**JS:** Java Script

**HTML:** Hyper Text Markup Language

## References

1. <https://doodle.com/free-online-voting>
2. https://www.easypolls.net/

# Current Software Architecture

There is an existing system for online voting.The application show all elections and

provide creating new elections.Registered use or visitor user can select category and

participate one of the viewing elections.User’s participate the viewing elections easily and

wiev results.User’s can create new elections and delete their elections.User’s has specific

profile screen.They can view their informations and edit their informations even wiev the

voted elections results.

The problem that we consider is difficulties voting and losses time.User’s could

register and login the online voting sytem and has a profile.The most important point is user

can vote easily and safety with online voting system.User didn’t want vote because of old

system voting take a long time and exahaustive.In our new system ,users saving time and they

vote elections safety.

# Proposed Software Architecture

The voting system is a mobile-based online application. This project will be very useful for

users and voting committees. Our system is easy to use and reliable, has a user-friendly

interface.

In short, in our system, the system allows users to participate in open voting, to vote, to be

rejected, approved or deleted by the management, to see the results of the voting results and

to see and to participate in popular voting. .

In addition, the system allows users to update their profiles, change their passwords, freeze

their accounts, and delete user accounts by the administrator.

## Overview

During the system design modeling of online voting(I-voted), we divided our system into subsystems. This provides us a strong coherence. Our subsystems are; Admin Management System, Registered User Management System, Anonym User Management System.

## 3.2 System Decomposition

The decomposition shows the existence of the following subsystems:

* Admin management subsystem
* Anonym-User management subsystem
* Registered-User management subsystem
* Database subsystem

**Admin Management Subsystem**

This subsystem is managing admin accounts. It offers perform creating an election, deleting an election, approving an election. Admin is the only actor who has permission to access add election and approves functions. This subsystem uses login services of the Registered User management subsystem.

The operetions provided by this subsystem are:

* Login()
* Add Election()
* Delete User()
* Delete Election()
* Logout()

**Registered-User Management Subsystem**

This subsytem is managing registered users’ functions, offers registered side to its functions. This subsystem manages registered users’ voting elections. Also this subsystem manages registered users’ applied elections, updating profile. After, logging step.

The operetions provided by this subsystem are:

* Login()
* Apply Elections()
* Update Profile()
* Search Elections()
* View Leadership()
* Vote()
* VoteRegisteredUsersElections()
* Logout()

**Anonym-User Management Subsystem**

This subsystem is menaging Anonym-Users. It offers perform voting for some elections, register to the system, view elections, view election results, view categories, search and view leardership table. When Anonym-User decide to register to system, this subsystem uses all functions of the Registered-User manegement subsystem.

The operations provided by this subsystem are:

* Register()
* Vote()
* View Elections()
* View Elections Results()
* Search()
* View Leadership()
* View Categories()

**Election Management Subsystem**

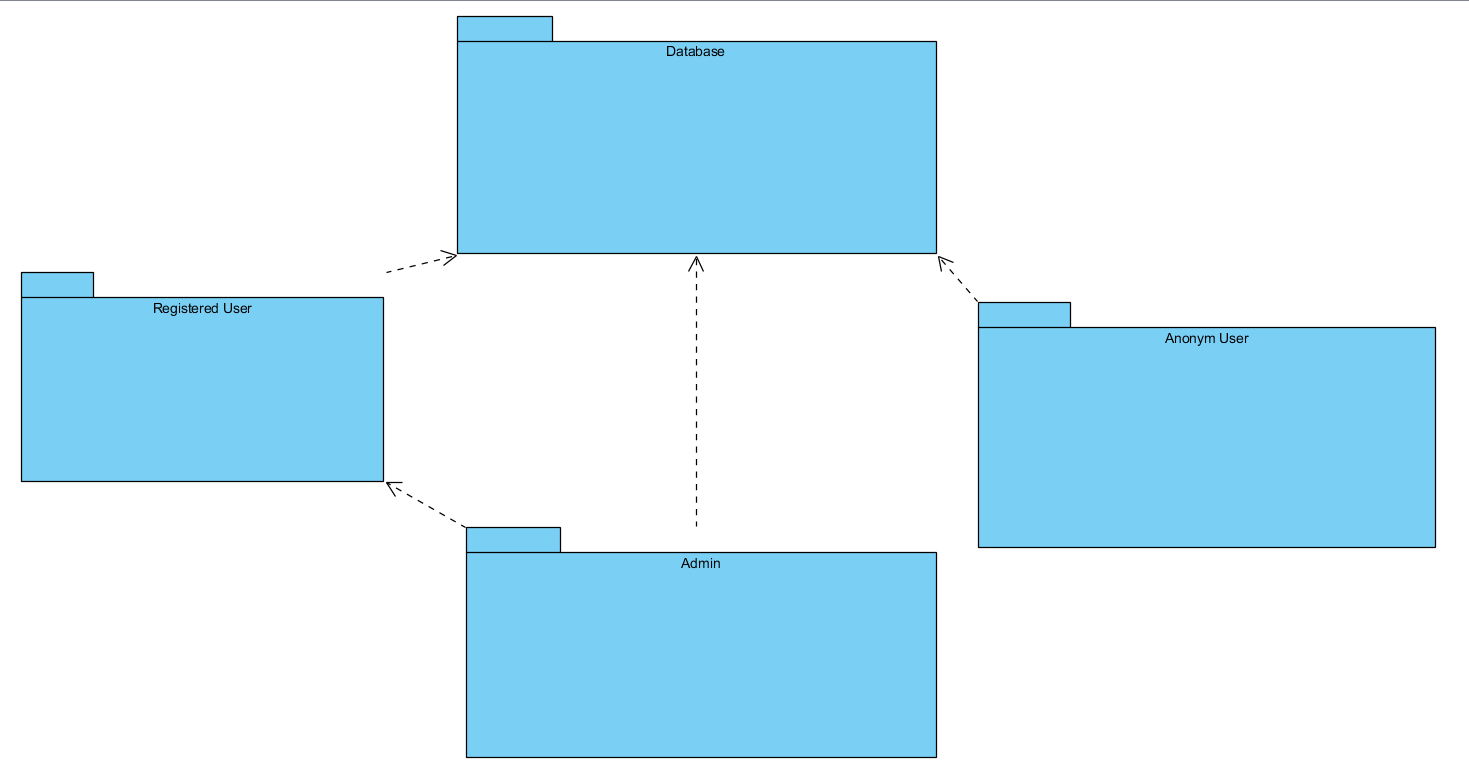
The most important part of this system is election. The election is a subsystem for this system because all users voting is connected to this subsystem. Election subsystem is managing many functions such as view the result of voting, vote for all users, creating an election etc. This system uses all functions of the Election management subsystem.

The operations provided by this subsystem are:

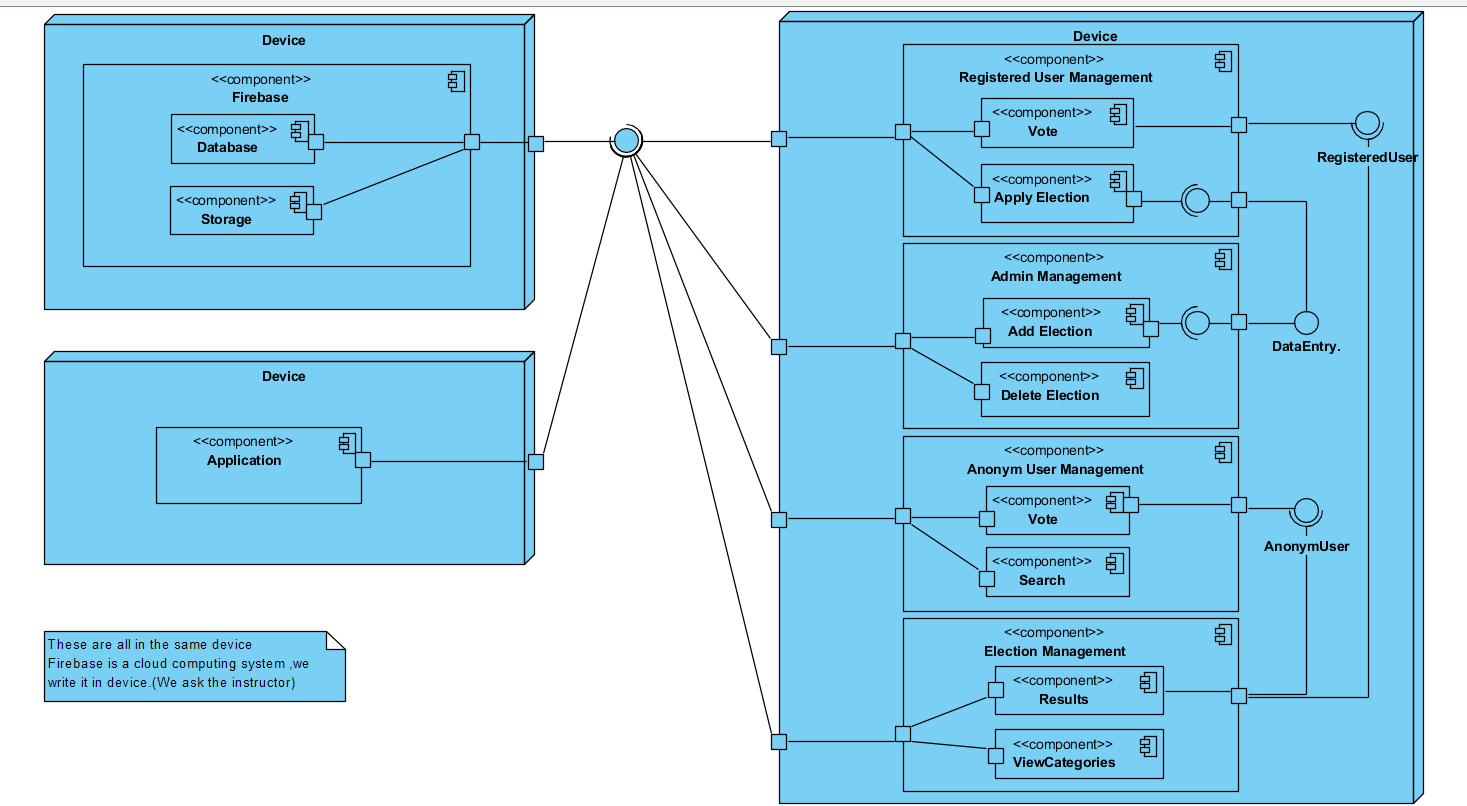
* Vote()
* ViewElectionResult()
* ViewElections()
* ViewCategories()
* SelectCategories()
* ApplyanElection()
* ViewMyPublishedElection()
* ViewMyVotingElection()

**Database subsystem**

This subsystem will be implemented by relational database management system which is Firebase. All subsystems are related and having service with this subsystem



## Hardware Software Mapping



## Persistent Data Management

Our system will use the firebase system to store data. This system will make it easier and

faster to read and write data. The database will store values ​​such as user input, user

information, and votes. Our database structure appears as follows.

## 3.5 Access Control and Security

The application is a multiuser application so it consists of 3 types of users which are registered user, anonym user and administrator. Because of this, the application will provide different interfaces for each user type.

First, the administrator will connect to the system with the membership interface, and will do the administrator's duties on the dating system, such as adding, updating, editing, deleting etc. By the way, registration is not necessary for the administrator through the system website, information will be entered manually into the database at the beginning of the system and the administrator will be the authority that will access the database directly. As a summary, the administrator does not have to register because it is initially registered in the database and the system.

The system will store all the information in the database and in the login processes again the system will use them by collecting data from the database. The information in the database will use both the confirmation and the use of the system for users.

Anonym user mustn’t log into the system these users can search, view leadership and vote systems elections.If they want to vote users’ elections they have to be registered users so they have to register to the system.

Registered user must log into the system to update profile, apply elections an vote users’ elections.

After these steps, the registered user will then connect to the system with your information such as e-mail and password. During the login procedure, the data in the user database table will be obtained and compared to the data entered by the Registered User.After log into the system Registered User can apply elections this means Registered User send elections with needed informations to admin for confirmation. Admin add this election, If the election confirmed by the admin.

During registration, field filling does not require access to the database, while completion of the process requires the data to be written to the database, which requires read and write access to the database. In that case, the required database fields will be blocked and simultaneous access of multiple email will be denied.

For some situation like updating or deleting information it is necessary to update one of the tables in the database in its phase of completion and therefore must be handled with more care since several users(Registered Users, Anonym Users and Admins) can be the cause of updating the table at the same time. This will also be avoided by blocking.

Finally, viewing the information or lists again requires read-only access to the database. Therefore, multi-user access does not impose problems and new restrictions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Actors/  Classes | Registered User | Authentication | Vote | Election |
| Admin | deleteUser() | Login()  Logout() |  | addElection()  deleteElection()  confirmElection() |
| Registered User | updateProfile()  freezeProfile() | Register()  Login()  Logout() | ViewMyVoting() | ApplyElection()  Complain()  searchElection()  Vote()  View() |
| Anonym User |  | Register() |  | searchElection()  Vote()  View() |

## Global Software Control

**External Control Flow (Between Subsystems):**

I-Voted system defined by the mobile application with a simple feature. Firebase DB system requests request submission of user data. Because the system is multi-user, simultaneous executions can occur. However, the control flow of a single user has a predefined form. After the login step, the system has a mobile page structure in the form by links or buttons.The services uses asynchronous callbacks to communicate with each other.

**Internal Control:**

• **Callbacks Between Subsystems**

The services use asynchronous callbacks for interprocess communication. Every single service uses internal methods to notify the other subsystems of its own status, its needs and abilities.

• **Avoiding Deadlocks**

Callbacks must not block the caller. So only status information is updated in the callee. All other internal work within a single service must be done by other worker threads.

• **Worker Threads for Each** **Service**

Each service has an own thread for communication, which communicates with the admin management system and other services.

**Concurrent Control:**

• **Multithreading :**

The I-Voted Firebase system uses threads, so that a large number of services is able to use the service manager simultaneously. The service manager also handles asynchronous events within the system.

• **Callbacks Between Subsystems** :

The I-Voted system uses asynchronous callbacks for interprocess communication.

**User Interface:** The system user interface will be made through application pages. The control of the next step depends on the user. In addition to this, the flow is implemented within the application pages. Most subsystems have a different application pages.Admin subsystems have web pages. Due to the system event-driven design, subsystems cannot be considered to have their own event loop. However, events are controlled by application pages(web pages for admin).

## Boundary Conditions

Startup: go to application and login

Shut Down: click log out and close application

**Error Conditions:**

**Login :**

* E-mail or password field cannot be blank.
* E-mail must be a e-mail format.
* Password and e-mail don’t match.
* E-mail is wrong or does not exist.
* Password must be more than five characters.

**Registered User Entry:**

* Enter invalid e-mail.
* Registered user information is incorrect or empty.
* Password and again password do not match.

**Add Election:**

* Admin must be registered.
* Text-fields is not blank.
* Complete the necessary information.
* Internet connection must be when admin click the add election button.

**View Popular Election :**

* User can view popular elections until the program allows.

**View Election Results :**

* User can view the results just the elections completed.

**Update Profile:**

* The current password doesn’nt entering truth.
* Confirm passwords do not match.
* Some necassary informations are doens’t pass empty.
* These conditions may cause the system displays warning message.

**Vote Election:**

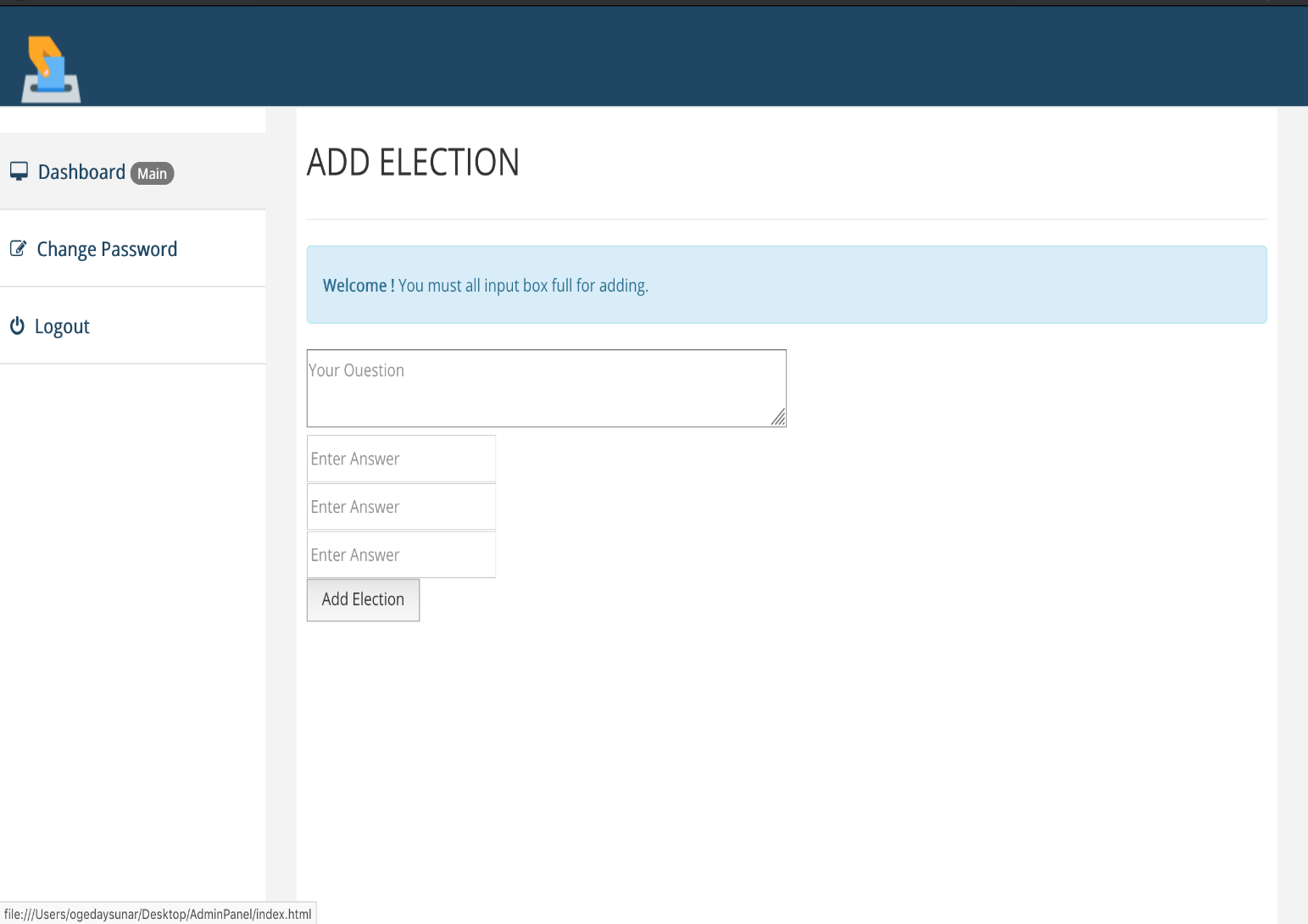
* User can vote election only once and doesn’t change.
* User just select one of the choice for the election.
* The election doesn’t send empty.
* User must complete the election rules.
* These conditions may cause the system displays warning message.

**Contact:**

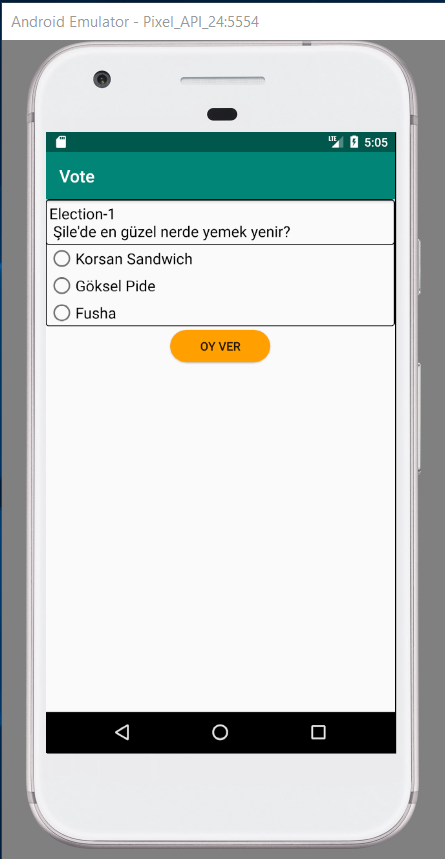
* User doesn’t pass the message box empy.
* User dosen’t exceed the given character limitation.
* These conditions may cause the system displays warning message.

# Subsystem Services

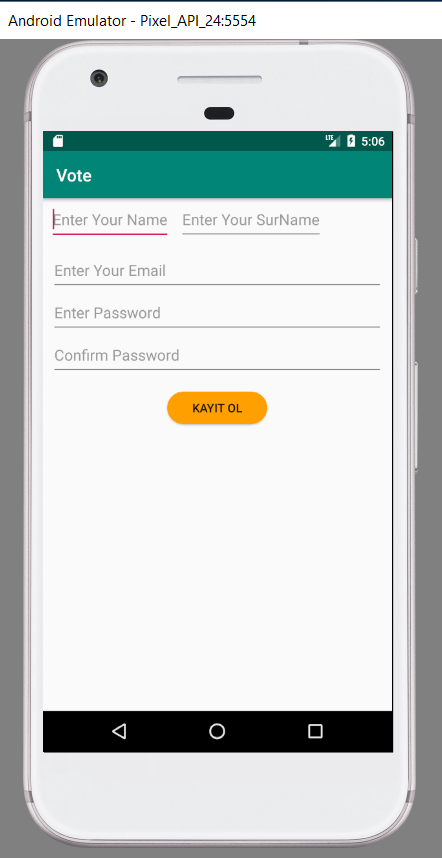
Add Election



Vote



Register



# References

1. <https://doodle.com/free-online-voting>
2. <https://www.easypolls.net/>
3. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.
4. Lecture presentations of the course (the presentations were provided by the Instructor who is Emine Ekin and Deniz Yigitbasi).