Online Voting System

Object Design Document

<1.0>

<23.12.2018>

Berkay Günay

Şuayb Talha Özçelik

Ögeday Sunar

Tuncer Oğuzkurt

Prepared for

SE301 Software Engineering



Table of Contents

[Introduction 1](#_Toc471092201)

[Object Design Trade-Offs 1](#_Toc471092202)

[Interface Documentation Guidelines 2](#_Toc471092203)

[Definitions, Acronyms, and Abbreviations 2](#_Toc471092204)

[References 2](#_Toc471092205)

[Packages 3](#_Toc471092206)

[Models 3](#_Toc471092207)

[Controllers 3](#_Toc471092208)

[Views 4](#_Toc471092209)

[Class Interfaces 5](#_Toc471092210)

OBJECT DESIGN DOCUMENT

# **Introduction**

While designing the application to the users to try to make the fastest way to design, we have stayed between the beautiful design.Database to repeat the data in a short time with the data to keep the data in some data, although we kept some database.

In our application integrated with Firebase, we transformed our code structures into a firebase format. We have tried for warning messages and feedback so that the users can think about errors and to understand their mistakes.

**Object Design Trade-Offs**

**Reliability**

We have designed the software with firebase for users can be use the application usually and safely. We have tested all the codes we have written. After each update or any changing, we have added to our project with testing. We have seen and corrected our mistakes easily because we are progressing regularly.

**Expandability**

While designing our system, we paid attention to extensibility. Thanks to its modular structure, our system is ready for new technologies that can be added later.

**Programmability**

In our project, we have created a simple and functional application using with android studio and firebase. The reason we use the Firebase is that simple and safety. In addition, android studio is supported to Firebase. In this way, we have designed the database part very easily and we have written our code according to these designs.

**Maintainability**

When we completed our project, we provide the necessary technical support. Users can reach our application on google play store and users can start to use from the download. We designed our system under stable and make easy to use. Of course, we will provide technical support for the admin management panel.

**Compatibility**

We have choice to use java, which is the languages we all know better in order to choose which platform will do the application. So, we have decided to design it in Android Studio. After that, the reason of we use firebase is that it supports to android studio. In this way, we have revealed a long-term software.

**Adaptability**

The application we have developed is designed to adapt to changing requirements and changing requests. We designed our system for current conditions however when the requirements and conditions changing on time our system can change and update with our database.

**Availability**

Availability is the ratio of time a system or component is functional to the total time it is required or completion of the transaction in the expected time. The online voting system we developed is constantly online and we designed to allow users access anytime.

## **Interface Documentation Guidelines**

**Top Navigation**

Our developed application has menu bar button. Users can pass the screens with the menu bar shortcuts.

**Icons**

We designed our system under stable and easy learn so we designed simple images for the describe functionality.

**Buttons**

While we were creating the buttons which we use in the application, we have used drawable and layout so with the xml source codes design which are offered by android studio. We have used particular colors during coloring. (All of the buttons are in the same color.)

## **Definitions, Acronyms, and Abbreviations**

**Acronyms**

Some button names and ID’s that developers can understand.

**Abbreviation**

There is no any abbreviation.

## 

## **References**

https://doodle.com/free-online-voting

https://www.easypolls.net

**Packages**

**Class**

* ApplyElections.java
* Contact.java
* Developers.java
* Election.java
* ElectionsObject.java
* Elections.java
* Leaders.java
* Login.java
* MainPage.java
* Model.java
* MyAdapter.java
* MyFirebaseIntanceIDService.java
* MyFirebaseMessagingServices.java
* MyPublishedElections.java
* ViewMyVote.java
* viewmyvotings.java

**ApplyElections.java**

**Fields**

-apply

-b

-ElectionName

-mAuth

-menuL5

-Option1

-Option2

-Option3

-Question

-sdf

-sdf2

-sdf3

-spinner

-tarih

**Constructor**

ApplyElections() // In apply elections java class user creates his own elections with needed attributes.

**All Methods**

onCreate() //This method creates the page

**Contact**

**Fields**

**-**textView76

-toolbar76

**Constructor**

Contact()

**All Methods**

onCreate() // This method creates the page

onOptionsItemSelected() // It has selected item which choose what to do

**Developers**

**Fields**

-toolbar3

**Constructor**

Developers()

**All Methods**

onCreate() // This method creates the page

onOptionsItemSelected() // It has selected item which choose what to do

**Election**

**Fields**

-Anonym

-Anonymi

-b

-Channel\_1\_ID

-Channel\_2\_ID

-database

-id

-idelection

-mAuth

-menu5L

-myRef

-notificationManagerCompat

-opt2

-opt3

-radio1

-radio2

-radio3

-radioButton

-radioGroup

-selection

-textView

-value

-valuecount

-voteboolean

**Constructor**

Election()

**All Methods**

createNotification() // When the vote received to us, app create a notification.

selection() // It takes previuous java class attributes (electionid).

sendOnChannel1() // The channel which is sended the notification.

onCreate() // This method creates the page

**ElectionObject**

**Fields**

-des // which is the finish date of election

-id // election id

-image // which shows election added from admin or user

-name // election name

**Constructor**

ElectionObject(java.lang.String name, java.lang.String des, int image, java.lang.String id)

**All Methods**

clone()

equals()

finalize()

getClass()

hashCode()

notify()

notifyAll()

toString()

wait()

**Elections**

**Nested Classes**

Elections.ElectionssAdapter

Elections.PopularAdapter

Elections.ResultsAdapter

**Fields**

-adapter

-adapterPopular

-art

-butonlar

-categories

-choosencategory

-cinema

-culture

-database

-history

-id

-k

-listOfElections

-listView

-mAuth

-myRef

-myRef1

-myRef5

-popularElections

-popularlistView

-populars

-resultAdapter

-resultListView

-resultsElections

-science

-Selection

-sport

-thereisno

-toolbarelections

-u

-urunara

**Constructor**

Elections()

**All Methods**

initList() // which is about search bar for empty string

onBackPressed() // which is about android devices back button

onCreate() // This method creates the page

onOptionsItemSelected() // It has selected item which choose what to do

searchItem() // It search a election from all elections.

selector() // It takes previuous java class attributes.(electionid)

sortPopulars() // It sort the popular elections

**ExpandableListAdapter**

**Fields**

-context

-listDataHeader

-listHashMap

**Constructor**

[ExpandableListAdapter](file:///C:\com\company\talha\vote\ExpandableListAdapter.html#ExpandableListAdapter-android.content.Context-java.util.List-java.util.HashMap-)(android.content.Context context, java.util.List listDataHeader, java.util.HashMap<[ViewMyVote](file:///C:\com\company\talha\vote\ViewMyVote.html),java.util.List> listHashMap)

**All Methods** //adapter methods add,delete,modify a item in the listview

addItem()

getChild()

getChildId()

getChildrenCount()

getChildView()

getGroup()

getGroupCount()

getGroupId()

getGroupView()

hasStableIds()

isChildSelectable()

**Leaders**

**Fields**

-name

-surname

-voteCount // This is sum of votes

**Constructor**

[Leaders](file:///C:\\com\\company\\talha\\vote\\Leaders.html" \l "Leaders-java.lang.String-java.lang.String-int-)(java.lang.String name, java.lang.String surname, int voteCount)

**All Methods**

getName()

getSurname()

getVoteCount()

setName()

setSurname()

setVoteCount()

**Login**

**Fields**

-a

-mAuth

**Constructor**

Login()

**All Methods**

isEmailValid() // It checks to email

openActivity15()//This method opens the Register page with in xml codes click setting.

**MainPage**

**Nested Classes**

MainPage.LeaderAdapter

**Fields**

-adres14k

-anasayfa

-butonlar

-category

-database

-dbUser2

-id

-imageButton4

-imageButton5

-imageButton6

-imageView

-leaderAdapter

-leaderlayout

-leaderlist

-leaders1

-leaders2

-mAuth

-menuL2

-menuL3

-myRef

-myRef2

-myRef3

-sdf2

-tarih

-textView1

-toolbar1

**Constructor**

MainPage()

**All Methods**

onBackPressed() // which is about android devices back button

onCreate() // This method creates the page.

onCreateOptionsMenu() // Which is create the option menu.

onOptionsItemSelected() // It has selected item which choose what to do.

sortArrayList() // It sorts to the ArrayList which is the list shows us leaders.

**Model**

**Fields**

-counter

-icon

-isGroupHeader

-title

**Constructor**

[Model](file:///C:\com\company\talha\vote\Model.html#Model-int-java.lang.String-java.lang.String-)(int icon, java.lang.String title, java.lang.String counter)

[Model](file:///C:\com\company\talha\vote\Model.html#Model-java.lang.String-)(java.lang.String title)

**All Methods**

getCounter() //It gets the count.

getIcon() // It gets the icon.

getTitle() // It get the title

isGroupHeader()

setCounter()

setGroupHeader()

setIcon()

setTitle()

**MyAdapter**

**Fields**

-context

-modelsArrayList

**Constructor**

[MyAdapter](file:///C:\com\company\talha\vote\MyAdapter.html#MyAdapter-android.content.Context-java.util.ArrayList-)(android.content.Context context, java.util.ArrayList<[Model](file:///C:\com\company\talha\vote\Model.html)> modelsArrayList)

**All Methods**

getView()

**MyFirebaseInstanceIDService**//Firebase necessary class for notifications

**Constructor**

MyFirebaseInstanceIDService()

**All Methods**

onTokenRefresh()

sendRegistrationToServer()

**MyFirebaseMessagingService//** Firebase necessary class for notifications

**Constructor**

MyFirebaseMessagingService()

**All Methods**

onMessageReceived()

showNotification() // It shows the notifications.

**MyPublishedElections**

**Nested Classes**

MyPublishedElections.PublishedAdaptor

**Fields**

-adapter

-database

-date

-id

-kullaniciElectionId

-l

-listOfElections

-listView

-mAuth

-myRef

-myRef1

-name

-Selection

-type

**Constructor**

MyPublishedElections()

**All Methods**

onCreate() // This method creates the page

selector()//It takes previuous java class attributes.(electionid)

**PopularElections**

Fields

-date

-image

-name

-voteCount

**Constructor**

[PopularElections](file:///C:\com\company\talha\vote\PopularElections.html#PopularElections-java.lang.String-java.lang.String-int-int-)(java.lang.String name, java.lang.String date, int voteCount, int image)

**All Methods**

getDate()

getImage()

getName()

getVoteCount()

setDate()

setImage()

setName()

setVoteCount()

**ProfileMenu**

**Fields**

-mAuth

-models

**Constructor**

ProfileMenu()

**All Methods**

generateData()

onBackPressed() // which is about android devices back button

onCreate() // This method creates the page

onListItemClick() //public method onItemClick() is the method invoked when an item is clicked

**Register**

**Fields**

-a

-b

-circularProgressButton

-confirmpassword

-dbUser

-demDownload1

-mAuth

-name

-runnable

-surname

-time

**Constructor**

Register()

**All Methods**

isEmailValid() // It checks the email validation.

isValidPassword() // It checks the password.

onCreate() // This method creates the page.

**Results**

**Fields**

-image //which shows us is election admin or user

-name

-option1

-option1C //option1 vote count

-option2

-option2C // option2 vote count

-option3

-option3C // option3 vote count

-question

**Constructor**

[Results](file:///C:\com\company\talha\vote\Results.html#Results-java.lang.String-java.lang.String-java.lang.String-java.lang.String-java.lang.String-int-int-int-int-)(java.lang.String name, java.lang.String question, java.lang.String option1, java.lang.String option2, java.lang.String option3, int option1C, int option2C, int option3C, int image)

**All Methods**

getImage()

getName()

getOption1()

getOption1C()

getOption2()

getOption2C()

getOption3()

getOption3C()

getQuestion()

setImage()

setName()

setOption1()

setOption1C()

setOption2()

setOption2C()

setOption3()

setOption3C()

setQuestion()

**UpdateProfile**

**Fields**

-dbUser2

-mAuth

-Updateemail

-Updatename

-Updatesurname

**Constructor**

UpdateProfile()

**All Methods**

onCreate() // This method creates the page

setText() //This method set the name String in EditText.

setText1() //This method set the surname String in EditText.

setText2() //This method set the email String in EditText.

**User**

**Fields**

-email

-name

-surname

**Constructor**

User()

[User](file:///C:\com\company\talha\vote\User.html#User-java.lang.String-java.lang.String-java.lang.String-)(java.lang.String email, java.lang.String name, java.lang.String surname)

**All Methods**

getEposta()

getIsim()

getSoyad()

setEposta()

setIsim()

setSoyad()

**ViewMyPublishedElections**

**Fields**

-database

-id

-myRef

-radio1

-radio2

-radio3

-textView

**Constructor**

ViewMyPublishedElections()

**All Methods**

onCreate() // This method creates the page

selection() // It takes previuous java class attributes.(electionid)

**ViewMyVote**

**Fields**

-image

-name

-Option1

-Option2

-Option3

-status

-UsersChoice

**Constructor**

[ViewMyVote](file:///C:\com\company\talha\vote\ViewMyVote.html#ViewMyVote-int-java.lang.String-java.lang.String-java.lang.String-java.lang.String-java.lang.String-int-)(int image, java.lang.String name, java.lang.String option1, java.lang.String option2, java.lang.String option3, java.lang.String usersChoice, int status)

**All Methods**

getName()

getOption1()

getOption2()

getOption3()

getStatus()

getTur()

getUsersChoice()

setName()

setOption1()

setOption2()

setOption3()

setStatus()

setTur()

setUsersChoice()

**viewmyvotings**

**Fields**

-database

-date

-electionname

-expandableListView

-id

-k

-kullaniciVoteId

-l

-listAdapter

-listDataHeader

-listHash

-mAuth

-myRef

-myRef1

-name

-option1

-option2

-option3

-publisher

-publisherdata

-Selection

-status

-statusdata

-type

-viewvotecount

-voteid

-voteidchoice

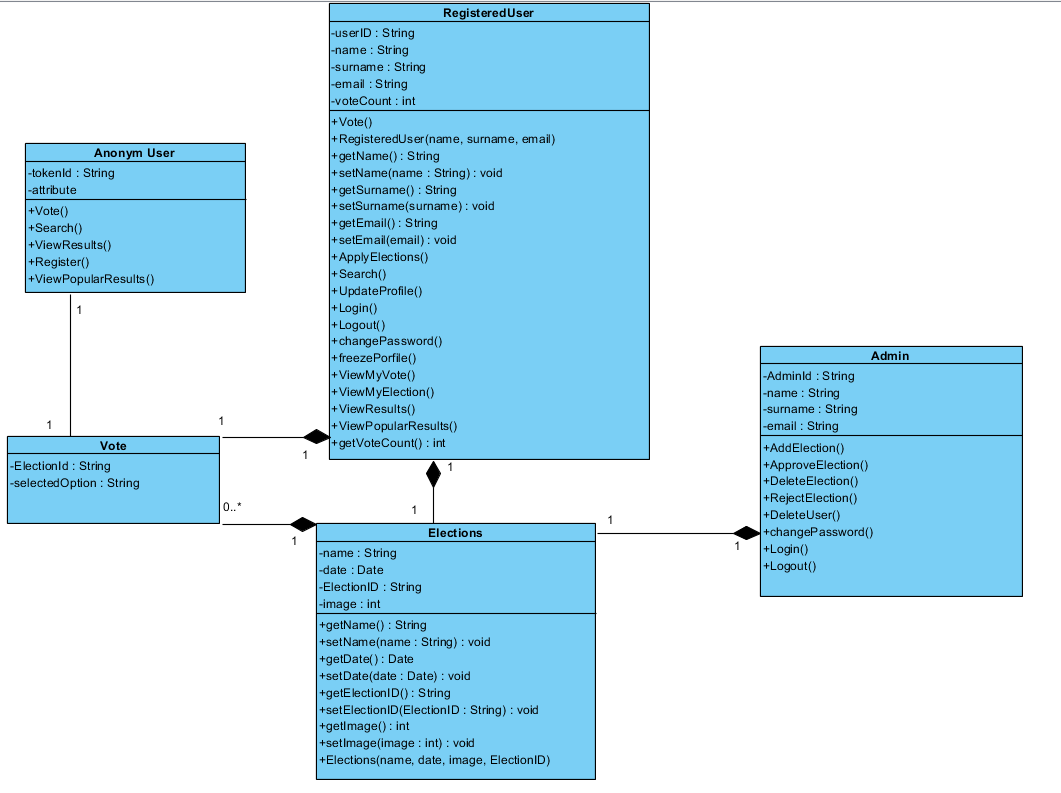
**Constructor**

viewmyvotings()

**All Methods**

onCreate() // This method creates the page

**Class Diagram**



**Database**

