**Survey4All**

**Object Design Document**

**23.12.2019**

Berkay Yılmaz

Gözde Gökyokuş

Umut Yıldız

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc28039133)

[1.1. Object Design Trade-offs 1](#_Toc28039134)

[1.2. Interface Documentation Guidelines 1](#_Toc28039135)

[1.3. Definitions, Acronyms, and Abbreviations 2](#_Toc28039136)

[1.4. References 2](#_Toc28039137)

[2. Packages 3](#_Toc28039138)

[3. Class Interfaces 4](#_Toc28039139)

OBJECT DESIGN DOCUMENT

# Introduction

## Object Design Trade-offs

* + 1. Functionality vs Staffing

As our development group have lost a member throughout its development cycle, we had to cut some functionality from the project to deliver it by the beta test date. (Ex: Favorite Survey function)

* + 1. Use vs Build

Survey4All is a system which can be divided into modules. We found some of the modules at the market. Importing modules to your project accelerates you and increases the quality of it as well. With the contribution of them we accomplished to manage most of the functionality. Here are some of the examples we got from the market:

AngularFireModule.initializeApp(environment.firebaseConfig),

AngularFirestoreModule,

AngularFireAuthModule,

BrowserModule,

AppRoutingModule,

BrowserAnimationsModule,

MDBBootstrapModule.forRoot(),

FormsModule,

AngularFireDatabaseModule,

ReactiveFormsModule

* + 1. Delivery Time vs. Functionality

As mentioned in 1.1.1, due to member loss and our busy midterm weeks, we need to get some speed but our development is going successfully on schedule. That’s why we should decrease functionality until the beta tests.

## Interface Documentation Guidelines

We have adopted specific interface documentation guidelines and conventions in order to make communication more uniform among the development team. First, all classes are named with singular nouns or noun phrases, with each word beginning with a non-capital letter. All objects derived from these classes (all class instances) will be named with a similar convention,. Methods are named with verb phrases; fields and parameters are named with noun phrases. Error status is returned via console.log(err) which directly prints them in to the console ,usually in the form of a non-Boolean status variable. Each class prototype will be included at the top of the header file before implementation of that class, and each class’ full interface will be desinged and changed in further development. Furthermore, each appropriate class will have fully pre-condition, post-condition, and invariant information, , including all attributes and methods.

## Definitions, Acronyms, and Abbreviations

- ID : Identification

- Angular : Angular is a TypeScript-based open-source web application framework. Its goal is to augment browser-based applications with Model–View–Controller (MVC) capability and reduce the amount of JavaScript needed to make web applications functional.

- Firebase : Google’s Firebase is a mobile and web application development platform which has several services. Given examples are the services we use:

* Cloud Firestore: Stores data.
* Authentication: Authenticates users simply and securely. Registration and Login Services are easy to implement.
* Hosting: Delivers web assets with speed and securely.
* Cloud Storage: Stores and serves files at Google Cloud. (ex:Profile Picture uploading and editing)

- Angular MDB: is a UI Kit for building responsive apps and websites which has ready to use libraries including plugins, animations, icons, templates as well as modules for CSS and JS.

- Login: to get access to an operating system or application, usually in a computer.

- ODD : Object Design Document

- UI : User Interface

- Server: is a computer,provides services to other computers

- User: a person who use the system

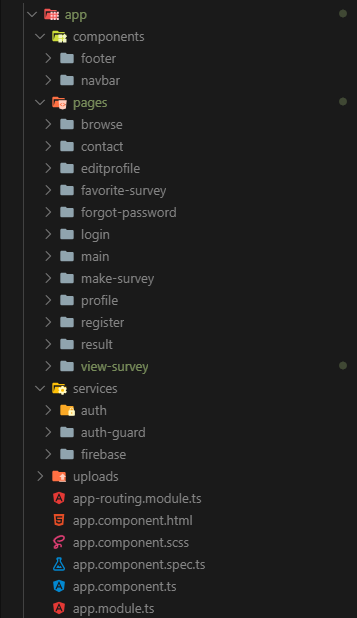
- Exception: Represents errors that occur during application execution.

- Error: The condition of having incorrect or false knowledge.

## References

* <https://angular.io/>
* <https://firebase.google.com/>
* <https://mdbootstrap.com/docs/angular/>

# Packages

The file organization of the code can be seen in the left.Our packages divided into 4 main parts and there are a lot of subsystem under each one of them.

These are as follows;

* Components Package: This package contains the main UI components , that is always visible.
* Footer: This represents the user interface that will be shown at the bottom on all other UI’s
* Navbar: This represents the user interface that will be shown at the top on all other UI’s.
* Pages Package: This package contains all of the UI’s used in Survey4All with all of the backend operations and functionality.

-Browse: This package represents the browse page, which means a whole page with UI and functions for browsing all of the surveys on the database.

-Contact: This package represents the contact page, which means a whole page with UI and functions for sending a message to Survey4all admins.

-EditProfile: This package represents the profile edit page, which means a whole page with UI and functions for editing their profile info like profile pic,phone etc.

-Favorite-Survey: This package represents the Favorite Surveys page, which means a whole page with UI and functions for the user to let user see and track their favorited surveys.

-ForgotPassword: This package represents the Forgot page, which means a whole page with UI and function for users who forgotted their passwords and want a reset link o their email address.

-Login: This package represents the login page, which means a whole page with which connected to auth-service,for users to log in if they wwant to use survey4all more efficiently(ex: creating a survey requires to login)

-Main: This package represents the Main page, which means the first page a user is going to see, a landing page.

-MakeSurvey: This package represents the Create Survey page, which means a whole page with UI and functions for creating a new survey by a user to share.

-Profile: This package represents the Profile page, which means a whole page with UI for users information stored and can be seen by the user.

-Register: This package represents the register page, which means a whole page with UI connected to auth service for auth operations. This page is a regular registeration form page.

-Result: This package represents the results page, which means a whole page with UI and functions for examining the result of a survey, visualized with the data.

-ViewSurvey: This package represents the Default Survey page, which means a whole page with UI and functions for a survey to be solved or completed by any user.

* Services Package: This package contains the required servies run by the system which includes connection to database , most of the auth operations and CRUD operations.

-Auth Service: This package represents the auth operations. It controls the operations like login, register,resetpassword,logout and it provides the user info to all pages in the system like user is logged in or not , what is the name of the user etc.

-AuthGuard Service: This package controls the limitations of the users , non users and admin. It protects the pages that should not be seen by everyone for example, creating a survey requires a user to be logged in so authguard will redirect user to the main page if the user somehow manage to find the orute fort he make survey page.

-Firebase Service : This service represents the main CRUD operations base. Firebase Service is getting data from the database and sends it to required pages. Reading Creating ,Updating or deleting operations moslty controlled here.

-Shared Service : This service represents a way to share data between unrelated components in the project. Observable based service works in the places like selecting a survey in the brose list and sending the selected data to the view survey page.

* Uploads:This package handles the uploading process on multimedia side. If a user wants to change their profile pic this service will be used.

# Class Interfaces

**Class Browse:**

**-Browse Html:**

This class contains information about browse page UI.

**-Browse TS:**

This class takes each survey datas from the Firebase.

**-SurveyFilter Pipe:**

This class controls the search function and filtering the survey list according to user input.

**Class Contact:**

**-Contact Html:**

This class contains information about Contact page UI.

**-Contact TS:**

This class controls the transmission of user requests and complaints to the Firebase.

**Class EditProfile:**

**-EditProfile Html:**

This class contains the information about EditProfile page UI.

**-EditProfile TS:**

This class enables to change user information by user and it updates the information on Firebase.

**Class FavoriteSurvey:**

**-FavoriteSurvey Html:**

This class contains the information about Favorite Survey page of user UI.

**-FavoriteSurvey TS:**

This class controls the adding the survey to favorite surveys of user.

**Class ForgotPassword:**

**-ForgotPassword Html:**

This class contains the codes of ForgotPassword page UI.

**-ForgotPassword TS:**

This class control the sending email for creating a new password.

**Class Login:**

**-Login Html:**

This class contains the codes of Login page UI.

**-Login TS:**

This class controls the user login function.

**Class Main:**

**-Main Html:**

This class contains the codes of Main page UI.

**-Main TS:**

This class is the main page and it enables to create a way to contact page and create survey page.

**Class MakeSurvey:**

**-MakeSurvey Html:**

This class contains the codes of MakeSurvey page UI.

**-MakeSurvey TS:**

This class create a survey by user.

**Class Profile:**

**-Profile Html:**

This class contains the codes of Profile page UI.

**-Profile TS:**

This class controls the showing profile photo and user information from Firebase.

**Class Register:**

**-Register Html:**

This class contains the codes of Register page UI.

**-Register TS:**

This class controls the registering to Survey4All the unregister user.

**Class Result:**

**-Result Html:**

This class contains the codes of Result page UI and survey statictics graph.

**-Result TS:**

This class controls the survey statictics function.

**Class ViewSurvey:**

**-ViewSurvey Html:**

This class views surveys of user.

**-ViewSurvey TS:**

This class controls the viewing the surveys.

**Class Uploads:**

**Upload TS:**

This class create the upload form including name image url when the image uploading to Firebase.