Survey4All

System Design Document

v0.5

15.12.2019

Berkay YILMAZ

Gözde GÖKYOKUŞ

Umut YILDIZ

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc27341479)

[1.1. Purpose of the System 1](#_Toc27341480)

[1.2. Design Goals 1](#_Toc27341481)

[1.3. Definitions, Acronyms, and Abbreviations 2](#_Toc27341482)

[1.4. References 3](#_Toc27341483)

[2. Current Software Architecture 3](#_Toc27341484)

[3. Proposed Software Architecture 3](#_Toc27341485)

[3.1. Overview 4](#_Toc27341486)

[3.2. System Decomposition 5](#_Toc27341487)

[3.3. Hardware Software Mapping 7](#_Toc27341488)

[3.4. Persistent Data Management 7](#_Toc27341489)

[3.5. Access Control and Security 8](#_Toc27341490)

[3.6. Global Software Control 9](#_Toc27341491)

[3.7. Boundary Conditions 9](#_Toc27341492)

[4. Subsystem Services 10](#_Toc27341493)

[5. References 10](#_Toc27341494)

SYSTEM DESIGN DOCUMENT

# Introduction

Survey4All is an survey management system developed fort he web platform using the Technologies like Angular 7 for the front end and Firebase firestore as the backend. It designed to become a collection of components that communicate flexibly with each other. Angular helps us to design visually beautiful pages for the survey4all and firestore gives us the flexiblity to store our data freely and connect it however we want. The website itself have some proffesionaly working websites on out day like; surveymonkey.com , microsurvey.com and surveey.com so on. The main difference between thos ewebsites and survey4all is, Survey4All is a survey management tool for everybody to see. So we wanted to use layered architecture on our project. Last of all our desing goals will be summarized as follows;

* Survey4All should be easy to use by all kind of users in term of usabilty
* Survey4All should be working correctly with all of it functions.
* Survey4All should be reliable in terms of saving the survey data without any fault and giving the correct results to the users all the time
* Survey4All’s codes must be Reuseable whick wil improve the efficieny and also should be flexible in terms of adding or removing parts of codes from it.
* Last of all Survey4All must be supportable for mobile devices which means it must be responsive to the screen size.

## Purpose of the System

The main purpose of the Survey4All is provide an online-platform that everyone can create a survey about something they wonder or need and also for the wanderers , a platform where they can give their opinion about their interests and may be find get some data about a survey they wonder. Its a cost free survey management tool for everyone.

## Design Goals

***Usability***

Since the end-user who will be using the system could be anyone in any age , it is essential for the system to be intuitive and easy to use by all kind of users.

***Multiple Users***

Survey4all should support a lot of simultaneous connections at the same time and also should be operative even if same task which efects data’s on the system done at the same time

***Reliability***

Survey4All must work %100 of the time. Survey4all is using the server support from the firebase which a product of a reliable company Google.On the code level survey4all’s all functions and operations should be tested relatively and must be crash/error free.

***Reusabilty***

Survey4All must have clear algorithms that could be developed further, the variable names should be understandable and unique, most importantly the source code and design patters should be reusable şn further development.

***Implementation***

There are no constraints on the hardware platform. There are no constraints imposed by the maintenance team. The design methodology is obtained as Agile Development Approach. System runs with firestore’s own query system in NoSQL which is more flexible than regular SQL query systems.It controllerd with the lambda notation and most of the data return as Document form. Html,Scss,Typescript languages is used in Survey4all as a part of Angular, Bootstrap ,Firebase Tools. Query based statements are handled by Firebase Firestore in NoSQL styke.

***Legal***

Survey4All does not have any license or any relative form of licensing. Its a student group project.

## Definitions, Acronyms, and Abbreviations

* **SDD** : System Design Document
* **Survey4All** : Survey4all is a survey management Website that let users to create share or fill surveys.
* **Firestore** : Firestore is a NoSQL based cloud database developed by Google.
* **Angular**: Angular is a structural framework for dynamic web apps.
* **Bootstrap**: Bootstrap is the most popular CSS Framework for developing responsive and mobile-first websites.
* **Admin**: System actor that administrates the system.
* **Subsystem**: Collection of classes, associations, operations and events closely related to each other.

## References

* www.surveymonkey.com
* www. microsurvey.com
* www. surveey.com

# Current Software Architecture

There are several websites that are providing survey management tool for companies or the ones who willng to pay. These websites moslty provide the tool for the customer who pays fort he service.Otherwise the results of any survey is hidden from the users is not customers. On the other hand there is websites that do it for free but most of them only supports for 1 question for fast foward surveys which will be used on the internet simultaniously in the situations like a streamer wants to get feedback from its users on time-based question or similar situations like that. But survey4all is free to use by everyone who wants to create a survey about anything they want or participate and see the reulsts of a topic they interested in. Since the cost of relevant services are not for regular users and the other systems simply case-based , Survey4all can get the place among the other survey management systems.

# Proposed Software Architecture

Survey4All is a system which you can create surveys, distribute them to public view and analyze the results. We wanted to make sure that people can create their own survey and answer the others’ surveys easily, that’s why we designed a user friendly, simple interface. The Survey4All System has 3 actors: Registered User, Unregistered User, Admin. Users can register to the system all free. Without registration, you can view and share surveys but you can not fill a survey, create/delete/edit your survey or favorite surveys. You can edit your profile and see your surveys on your profile page if you are a registered user. Survey and user informations are stored in the database. The data is retrieved from our database provider Firebase by queries.

## Overview

The system will have a layered architecture and there will be 3 layers, which are:

1. Interface
2. Application
3. Storage

Users will be interacting with the system using the interface layer, application layer will have all the functions of the system in it and storage layer will have all the data about the users and surveys. Storage layer will have the registeredUserData, SurveysData and incomingMails. Subsystems retrieve these information from our database. Application layer will have the survey, admin, user(unregistered and registered user) subsystems. User subsystem will have the functions of register, login, searching a on the system, editing the user profile and sending messages to the website admin. User subsystem keeps some of the functions for both unregistered and registered user such as viewing surveys but some of the functions are only for registered users such as favorite survey, delete survey, create survey, edit profile.

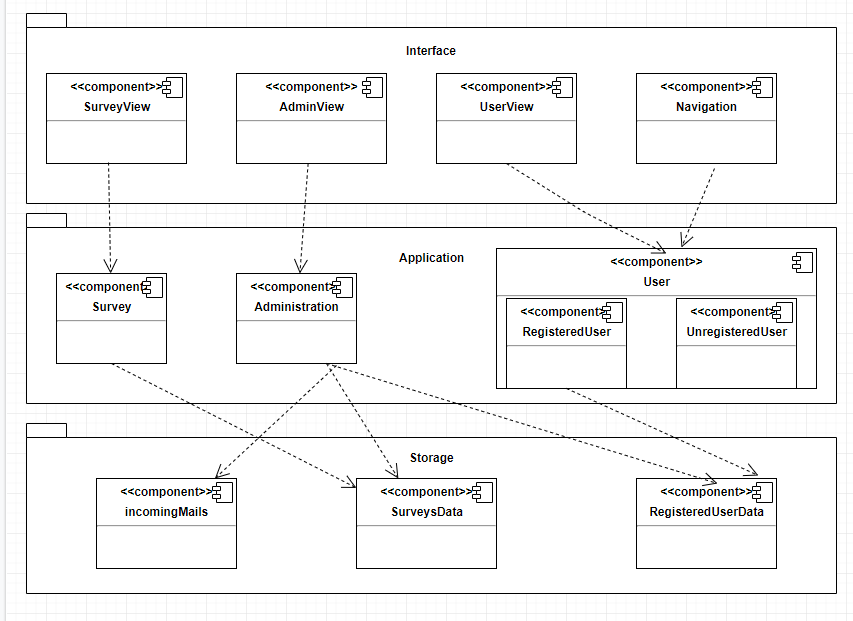
* Registered user subsystem will keep the functions of login, editing profile, messaging, create a survey, fill a survey, favorite a survey.
* Unregistered user subsytem has access to view surveys, fill a survey, search for a survey, share survey.
* Survey subsystem handles the surveys created by registered user, will have the functions of giving a respond to the survey.
* Admin subsystem will have the functions of managing a survey such as approving or

rejecting the auction, managing the user profile such as banning their

account, delete inappropriate surveys. Admin has and managing the contact us messages.

.

## System Decomposition

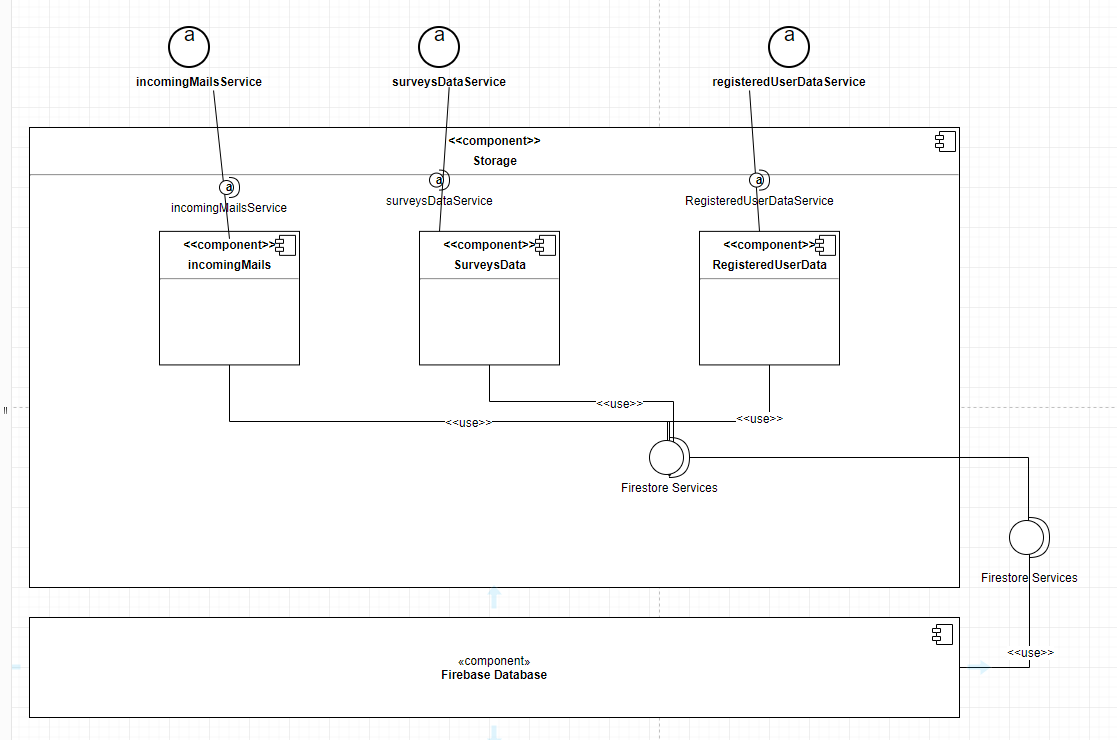


Storage layer holds the RegisteredUserData, SurveysData and incomingMails subsystems. RegisteredUserData has the information (e-mail, name, phone number, user type…) about the Survey4All users which we define as RegisteredUsers and the Admin.

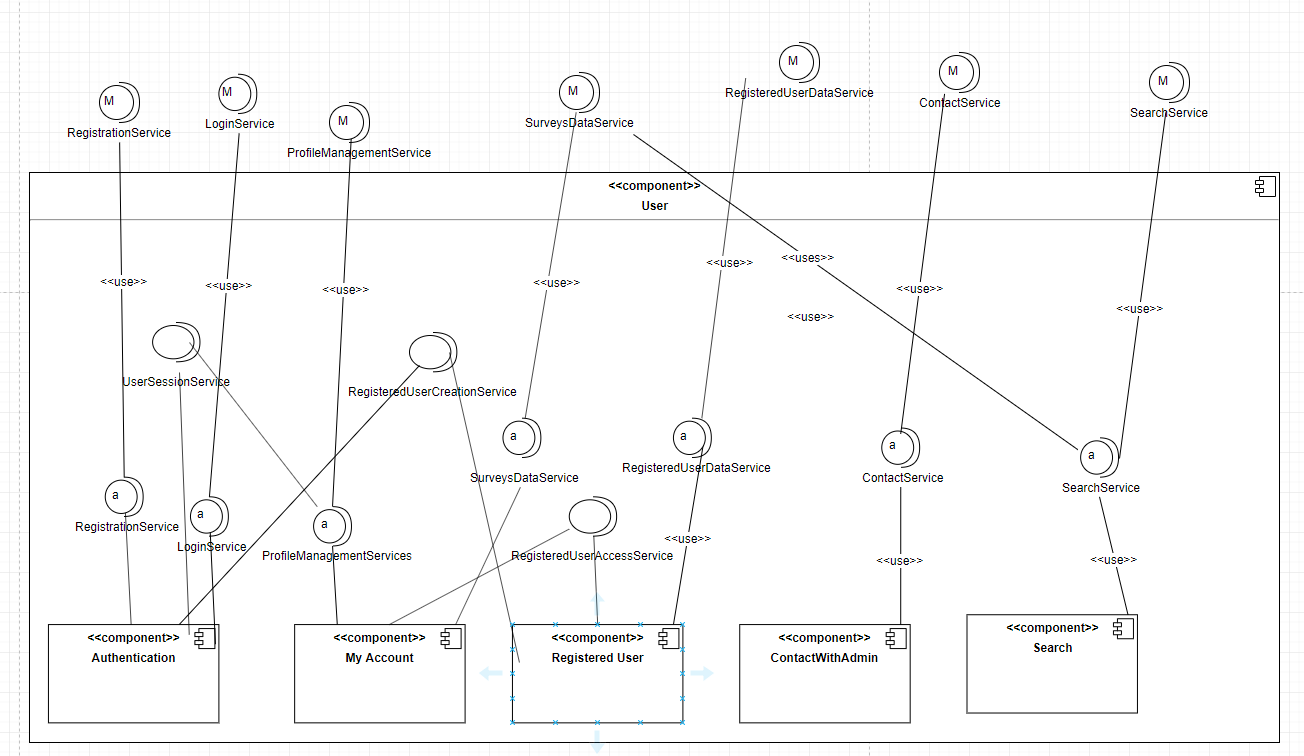
Application layer holds Survey, Administration and User subsystems. User subsystem has 2 different subsystems in it: RegisteredUser, UnregisteredUser. User subsystem is the subsystem that manages all the common services that a user needs. It is about user functionalities like messaging, profile management, search surveys and log-in/log-out services. Administration subsystem is the subsystem that manages the all admin privilages like deleting surveys, viewing contact us page mails, or even banning a user’s account due to his/her usage of the system. Survey subsystem has all the functionalities any survey can proceed such as creation of a survey, deletion of a survey, extraction and analysis of the results of a survey.

As the last of our layers, Interface layer has the boundary objects, all user interface that users interact with. It holds SurveyView, AdminView, Navigation and UserView. SurveyView is the subsystem that controls the ongoing or finished survey services. AdminView is essential for admin to manage common services that stabilizes and holds the control users and surveys for the sake of the system. Navigation is the primary for services like login search and register. It has routing service directly or indirectly (For ex: by pressing a button or navbar tab) UserView has services that giving users to manage simple actions. Users with the userType: user, unregistered users and the admin see different pages because of the authentication service which Firebase provides us.

We use Firebase’s Firestore Database which is a NOSQL database. So in order to get a data we type the query Firestore Service provides us. In the Storage subsystem we collect all necessary data such as incomingMails, RegisteredUserData, surveysData and connect to the Database system via Firestore Service.

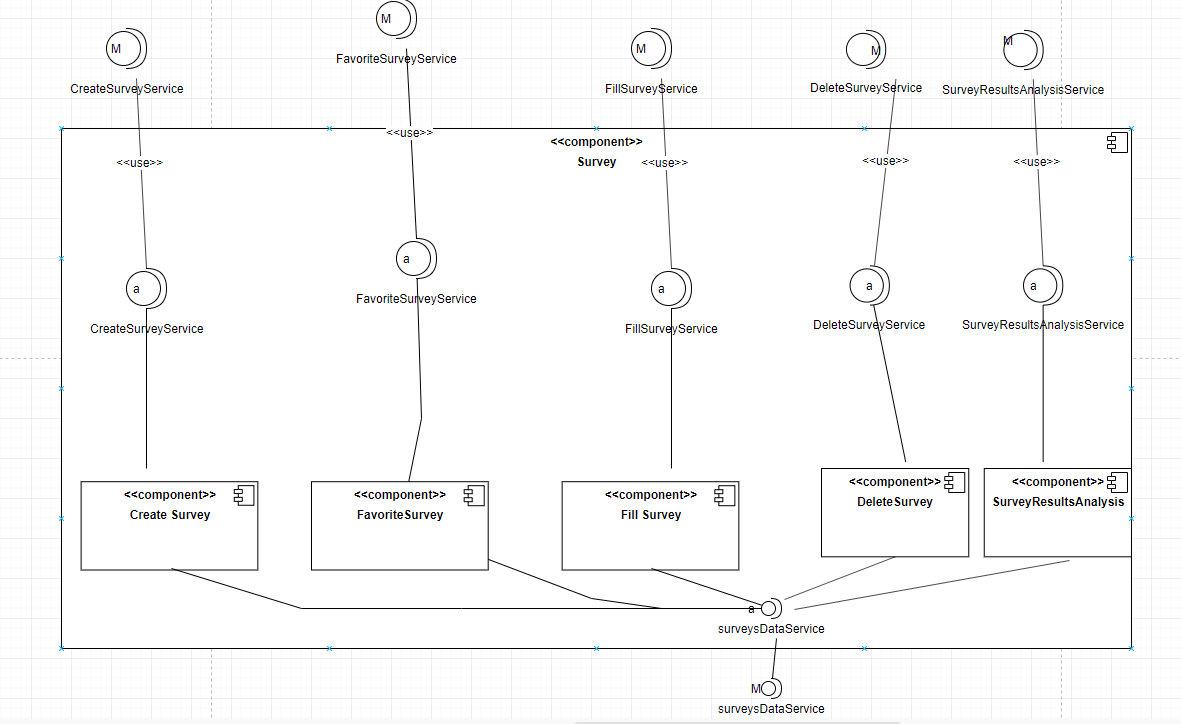


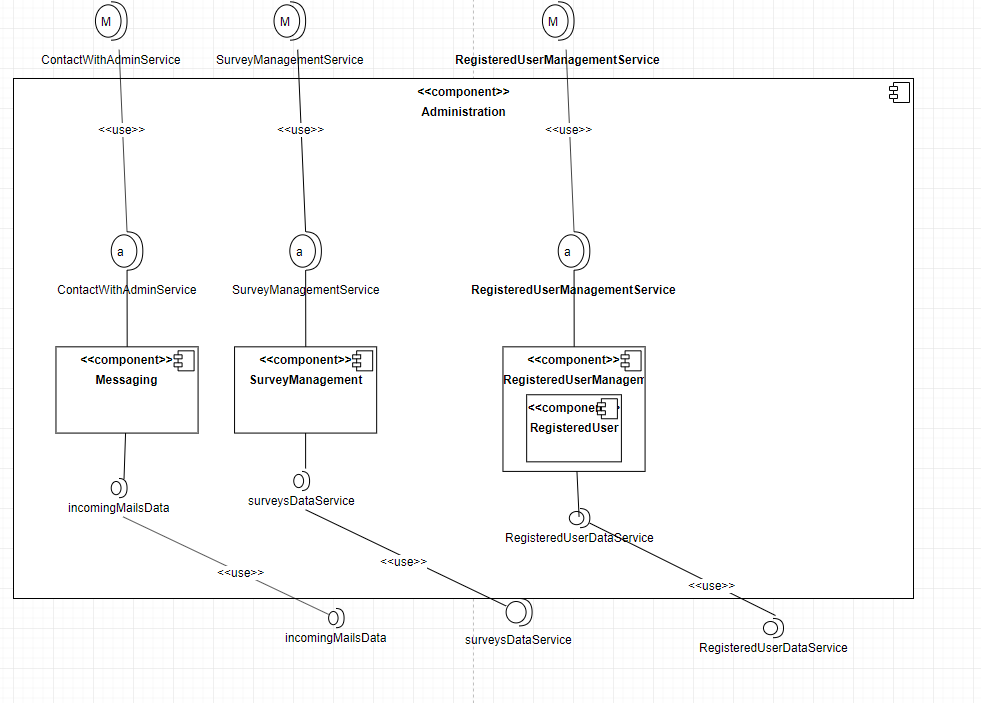
The User subsystem provides registeredUsers and their functions such as registration, login , message admin, edit profile, create survey, favourite survey.



The Survey subsystem has following functions: Create survey, delete survey, fill survey,

favorite survey, survey results analysis service”





The Administration Subsystem handles Admin’s functionalities, such as management of registeredUsers (banning their accounts, deleting inappropriate surveys etc.), survey information, viewing received messages from RegisteredUsers or UnRegisteredUsers on contact us form.

Here is the interface layer diagrams containing:

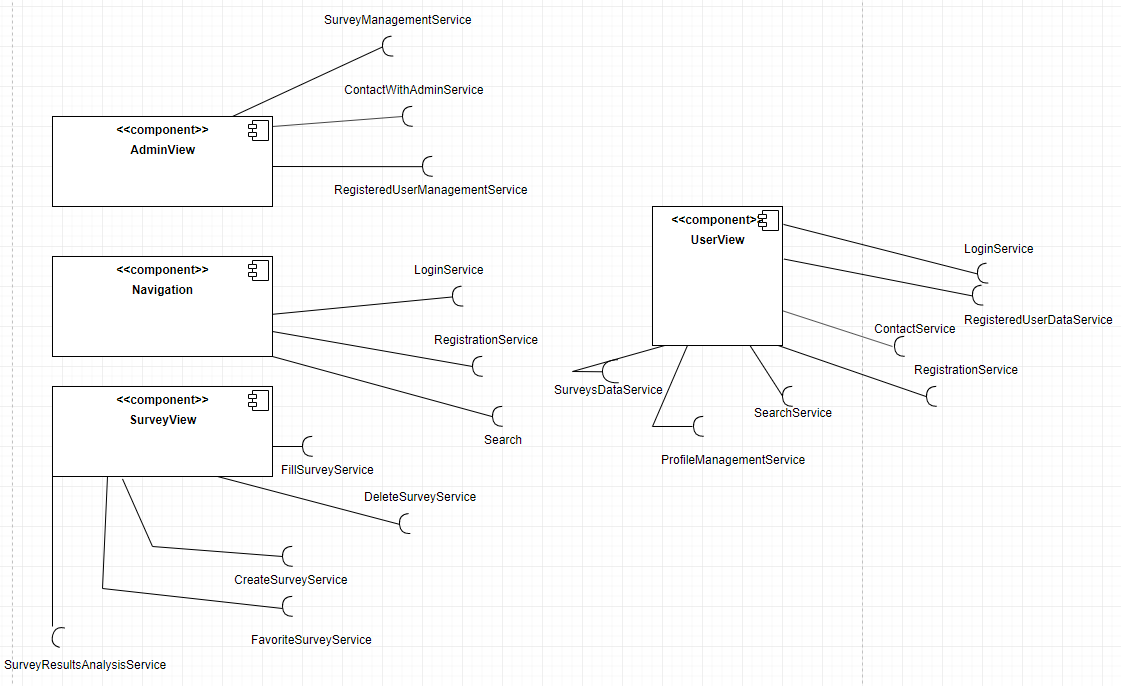
-User View

-AdminView

-SurveyView

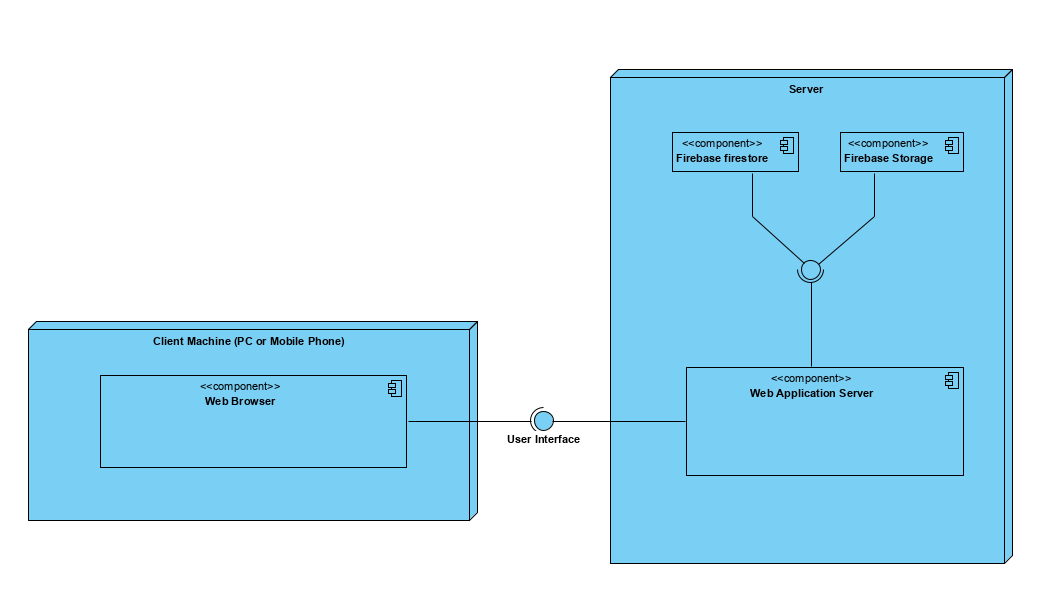
-Navigation

User type selects which interface will be shown.



Our system is a web application that is written with Angular, and interface layer holds the view users can interact with. It holds 3 View subsystems for each application layer subsystem. It also holds the Navigation subsystem which holds the pages like homepage, register and login, browse surveys, contact us.

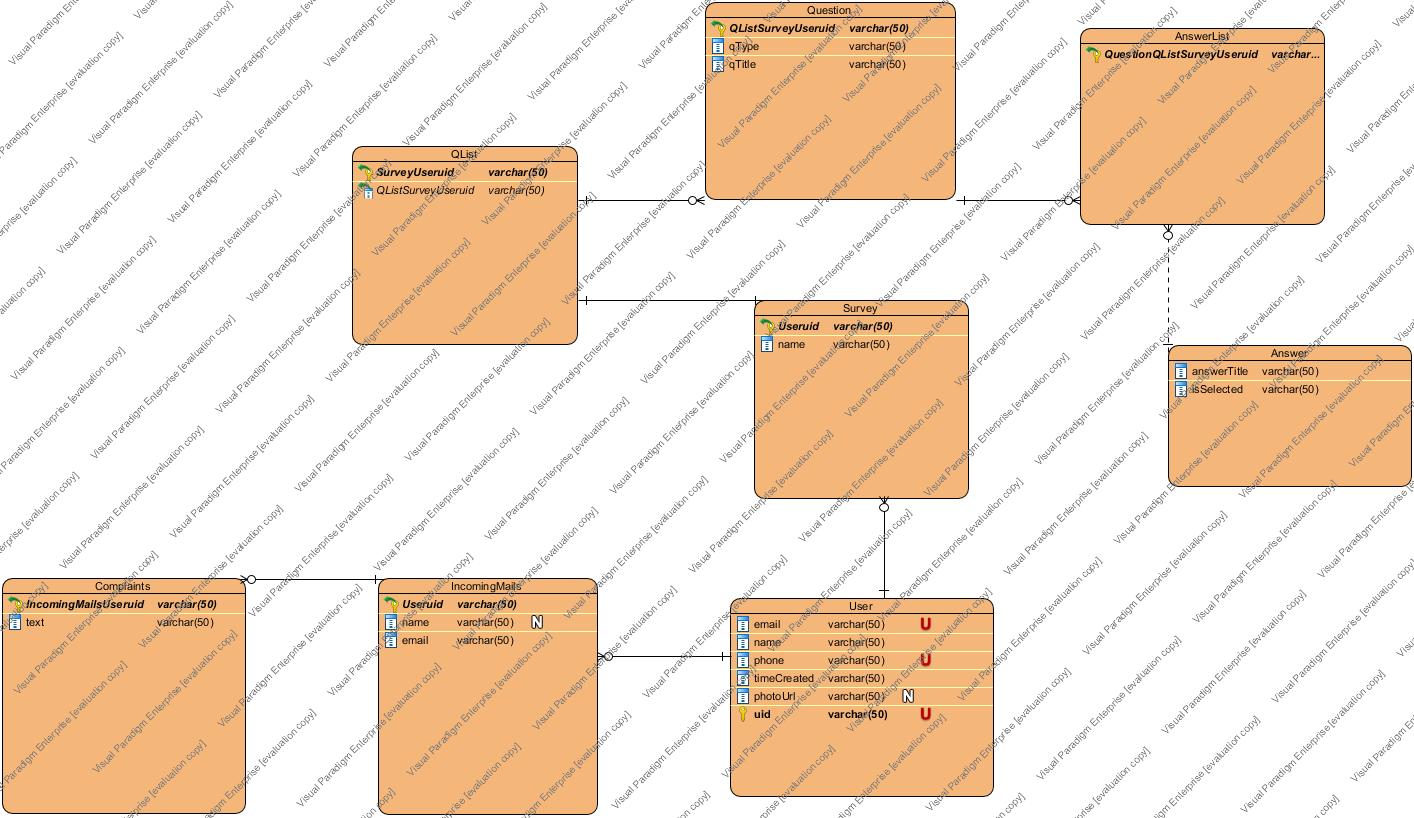
## Hardware Software Mapping



## Persistent Data Management

The system must needs persistent storage of users and their information , questions which created by user and answers which selected by user, their surveys, their complaints and their relationship each other and the data must be accessible from different users and every request should be returned as fast as possible.Also users can be controlled and we needs Real-Time Database that’s why using a Real-Time Database is must.

Security and growing system storage are the biggest concerns and there should not be much work with the server side.



The data scheme for the database is mainly made up of its users , its survey created by registered users , its questions and its answers and its complaints. The table ‘User’ holds information of user. The table ‘Survey’ holds name of survey and list of questions which created by registered user. The table ‘Question’ holds types of question and title of question and a list of answers for each question.The table ‘IncomingMails’ holds name and email of user who will submit a complaint and a list of complaints which submitted by user.

Database will provide query-independent access to data for the higher layers for encapsulation.

## Access Control and Security

|  |  |  |  |
| --- | --- | --- | --- |
|  | Administration | User | Survey |
| Admin | banUser()  viewUserInformation()  viewUserList()  viewComplaints() | login()  logout() | deleteSurvey()  shareSurvey() |
| RegisteredUser |  | login()  logout()  forgotPassword()  searchUser() | favoriteSurvey()  deleteSurvey()  unfavoriteSurvey()  searchSurvey()  fillSurvey()  shareSurvey()  createSurvey()  editSurvey()  checkSurveyStatictics()  checkSurvey() |
| UnRegisteredUser |  | register()  searchUser()  forgotPassword() | fillSurvey()  shareSurvey()  searchSurvey()  checkSurvey() |

Users must provide an at least 8 characters long password to register. The database shall hold the password information hashed. As the system is not designed to hold sensitive information for users, no other encryption is required.

## Global Software Control

Describe how the global software control is implemented. In particular, this section should describe how requests are initiated and how subsystems synchronize. This section should list and address synchronization and concurrency issues.

## Boundary Conditions

Startup:

- First of all , admin is declared with Angular.

Register:

* Email is already used.
* Phone is already used.
* Password is less than 8 characters long.
* Password and password again does not match.
* There are blank fields.

Logging in:

* Email or password are wrong.
* Email or password fields are blank.
* Email or password don’t match each other.
* Password is less than 8 characters long.
* After the logged in , the main page does not appear.
* ForgotPassword button don’t send link.

Profile Management:

* There are blank fields.
* System crashes when user edits its information.
* User uploads wrong file type.

Survey Management:

* Title field is blank.
* User does not select question type.
* Answer field is blank.
* User save blank survey sketch.

Searching:

* User or survey which is searching are not exist .

# Subsystem Services

**4.1 Presentation Layer Subsystem Services**

**-** Takes user input from Interface Layer

**-** Passes user input from Interface Layer to Application Layer

**-** Displays Application Layer Results

**4.2 User Subsystem Services**

**-** Registration Service : User Registration.

**-** Login Service: User Login.

**-** Search Service : Searches users and their survey.

**4.3 Storage Subsystem Services**

**-** IncomingMail Service : sending complaints which user has to admin .

**-** SurveysData Service : Service which store, edit , delete survey information.

**-** RegisteredUser Data Service: Service which store, edit , delete user information.

**-** Firestore Service: Service of Firebase.

**4.4 Survey Subsystem Services**

**-** Create Survey Service: enables to create a survey by user.

- Delete Survey Service : enables to delete a survey by user.

- Edit Survey Service : enables to edit a survey by user.

# References

The following is an example of listing a book in this section. Check the text to see how it is cross referenced (The whole document is based on [1]).

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.