Bilkent University
Department of Computer Engineering

# Senior Design Project

## FAVEO

# High Level Design Report

Zafer Çınar 21601514
Engin Deniz Kopan 21301826
Enes Varol 21604086
Enes Yıldırım 21602725

Supervisor: Uğur Doğrusöz
Jury Members: Çiğdem Gündüz Demir, Hamdi Dibeklioğlu

December 31, 2019

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

# 1. Introduction

There are not enough mobile applications for visually impaired people's usage. This lack of service occurs especially in education area. FAVEO is an online quiz application for android that utilizes text to speech [1], Voice Control [2], TalkBack[3] and image processing to help visually impaired people's studying. In more detail they can solve questions, test their knowledge and also prepare their questions and share with others.

## 1.1 Purpose of the System

System's main purpose will be solving questions as well as preparing questions. Faveo will be an Android application for visually impaired people. Users can test their knowledge and practice topics with the help of the application. If they like users can create custom quizzes.

## 1.2 Design Goals

### 1.2.1 Performance

- System have to provide fast and accurate responses to users' requests(approximately around 1 seconds).
- System should be able to respond multiple requests at the same time(approximately around thousand request simultaneously).

### 1.2.2 Extensibility

- System should be extensible for additional categories of questions and also for development of iOS version.

### 1.2.3 Usability

- The system will be implemented in Turkish.
- Application will have accessibility features.

### 1.2.4 Security

- The system shall prevent access to the system by unauthorized parties.

### 1.2.5 Supportability

- The API, platform and data sources which is planning to be used in FAVEO, are open to future updates. So FAVEO should adapt to these possible changes with ease.

## 1.3. Definitions, Acronyms and Abbreviations

- **Client:** User End of the application. For FAVEO, it is the android application.
- **Server:** The part of the application which returns response to the Client's requests. Responsible for data management and API interactions. In FAVEO, distributed servers pretend server role.
- **API:** Application programming interface.
- **HTTP:** Hyper-Text Transfer Protocol.
- **URL:** Uniform Resource Locator.
- **JSON:** JavaScript Object Notation.
- **Python:** Widely used, high-level, interpreted, general purpose programming language.
- **Android OS:** Android is a mobile operating system developed by Google.
- **GUI:** Graphical User Interface.
- **DB:** Database.
- **SDK:** Standard Development Kit, tools and libraries by Android Studio.
- **MVC:** Model-View-Controller.
- **OCR:** Object character recognition.

## 1.4 Overview

The main features of FAVEO will be solving questions as well as preparing questions. User will choose a category from the fixed list to solve questions and they can solve with the help of TalkBack or Voice Control. Questions will be read by text to speech. To upload a question user can use upload image option that extracts the text from image and convert it to question. User can make changes on result of image processing or they can upload a voice questions.Questions can be true/false or multiple choice. If they like parents can create custom quiz which will be accessible only by their account and can be modified.

# 2. Current Software Architecture

In this section of the report software architectures of Kahoot[4], QuizUp[5] and Trivia Crack[6] will be explained.

## 2.1 Kahoot

Kahoot is a platform where users can create quiz and host it for other users to join and compete with each others. Basically it has the following features:

- Create- User create a learning game or trivia quiz on any topic, in any language.
- Host- Show quiz questions on screen to competitor.
- Play- Join to the host game and read the questions from phone or screen and solve it.

## 2.2 QuizUP

QuizUp is a platform where users can upload questions to the system and solve questions from different topics while competing in pairs. Basically it has the following features:

- Upload- User can upload questions to the system.
- Play- User can compete with others and solve questions from different topics.
- Ranking- System will rank users according to their result on questions.

## 2.3 Trivia Crack

Trivia Crack is a platform where users compete in multiple categories which determined by the system. Basically it has the following features:

- Spin- Spin the wheel to get a question from a topic.
- Get Character- User gets a character for every correct answer.
- Challenge- User can challenge other user for a character.

# 3. Proposed Software Architecture

## 3.1 Overview

From system architectures  “Three Tier” architecture is chosen for FAVEO and it has Presentation, Logic and Data tiers. System’s architecture will be discussed in detail in following sections. Firstly, subsystem structure will be discussed. Secondly, details of hardware and software mapping will be explained. Furthermore, data which will be stored by the system will be explained in “Persistent Data Management” and security of data will be discussed in “Access Control and Security”. Finally system and user interaction then boundary conditions will be discussed.

## 3.2 Subsystem Decomposition

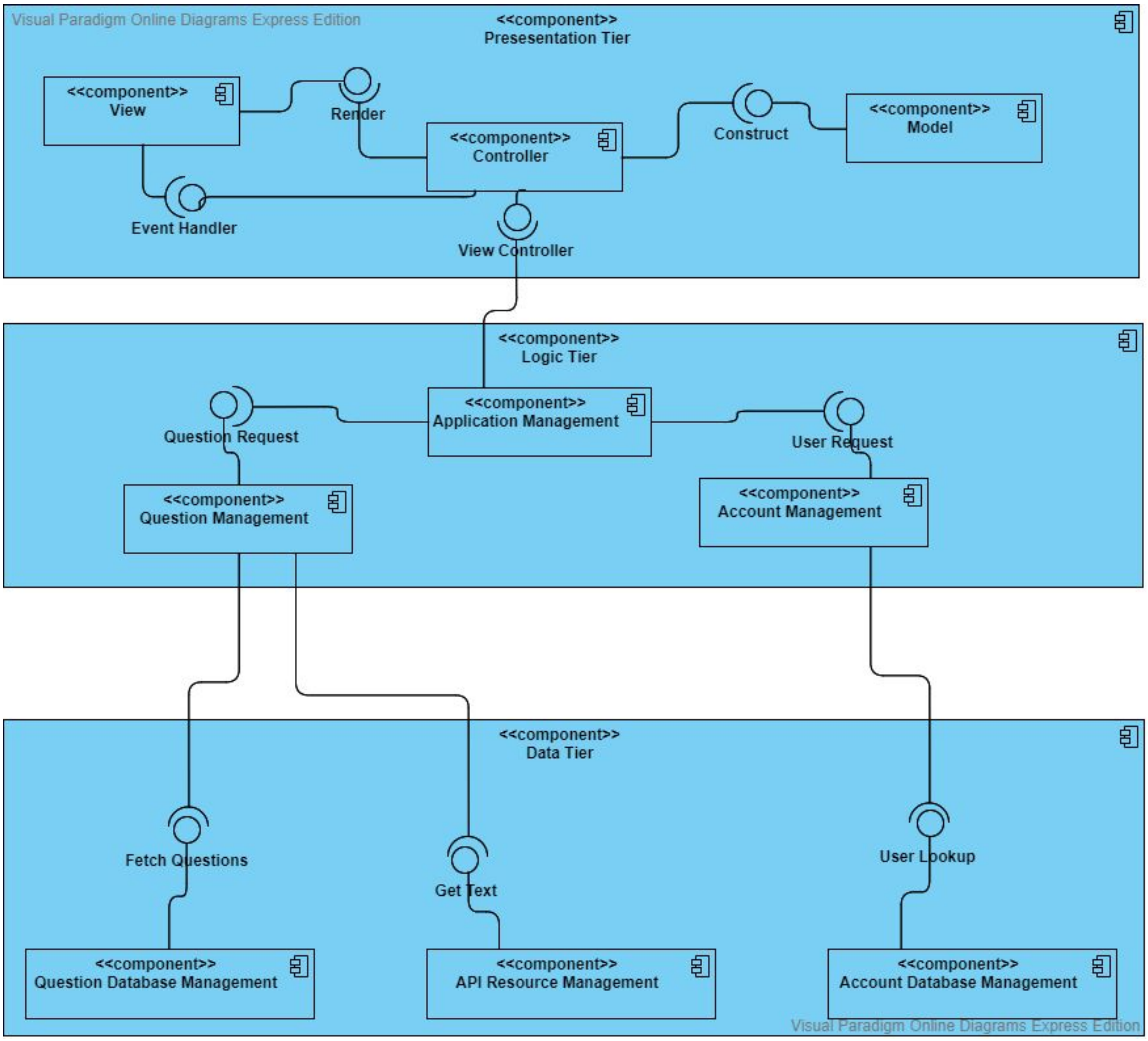

Figure 1: Subsystem Decomposition of FAVEO

Our system is decomposed into 3-tier architecture. Decomposed system consists of 3 different tiers called Presentation Tier, Logic Tier and Data Tier.Our main aim is to achieve minimized coupling and maximized coherence between these components.

Components of the system in detail are as follows:

- **Presentation Tier:** It is also known as client tier. The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results into something that the user can understand easily.
    - View is responsible for the interaction of the application with user.
    - Controller is what manages the transfers between view and model. Moreover, connects presentation tier to the logic tier.
    - Model handles the data/objects that is relevant for the interface.

- **Logic Tier:** It is also known as application tier. This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.
    - Application Management is responsible for handling the requests from the user and responding back to the Presentation Tier.
    - Question Management is responsible for retrieving the content of question from the Data Tier and transfer this information to Application Management.
    - Account Management is responsible for sending requests to the Account Database Management for login and signup.

- **Data Tier:** In here, information is stored and retrieved from a database system. The information is then passed back to the logic tier for processing, and then eventually back to the user. It is going to be deployed on server side. Providing secure and reliable access to the database is the main functionality of this layer.
    - Question Database Management is responsible for storing quiz questions and responding to calls coming from logic tier.
    - API Resource Management is responsible for retrieving the text which will be captured from a photo and storing it.
    - Account Database Management is responsible for managing the user profile information.

## 3.3 Hardware/Software Mapping

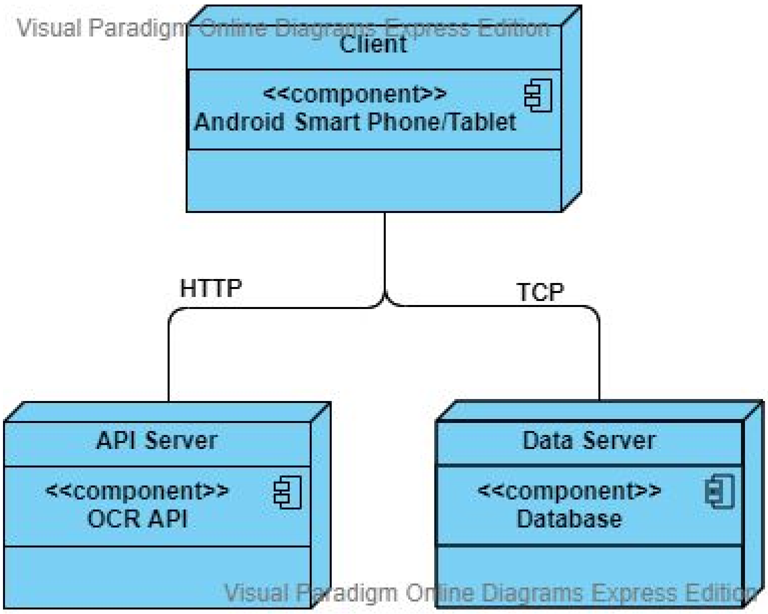
Visual Paradigm Online Diagrams Express Edition
Client
<<component>>
Android Smart Phone/Tablet
HTTP
TCP
API Server
<<component>>
OCR API
Data Server
<<component>>
Database
Visual Paradigm Online Diagrams Express Edition

Figure 2: Deployment diagram of FAVEO

FAVEO distributed to 2 layers: Client which is android application that works on android smart phone/tablet and API/Data Server. Android application will corresponds to the View and Management tiers meanwhile API/Server will corresponds to Data tier. Client layer will send API requests to the API Server for OCR and will send data retrieval requests to the Data Server. Server layers will receive the requests and updates data and responds accordingly. Data tier will developed by using Firebase[7].

## 3.4 Persistent Data Management

User related information email and password will be stored in the system. Moreover, question related information such as question(text or voice) with corresponding multiple choices or true/false choice will be stored in the system.

## 3.5 Access Control and Security

No user information will be stored in the system besides email and password. Moreover, access control and security of these informations will be ensured with testings and evaluations.

## 3.6 Global Software Control

System has a event-driven control to avoid race conditions and operations will be done in proper sequence.

## 3.7 Boundary Conditions

### 3.7.1 Initialization

User access to the system with android implementation. User need to sign up in order to use the system. Application requires internet connection to work properly.

### 3.7.2 Failure

Proper error messages comes up for according failures such as API issues, wrong input, internet connection failure and etc.

# 4. Subsystem Services

## 4.1 Presentation Tier Subsystem

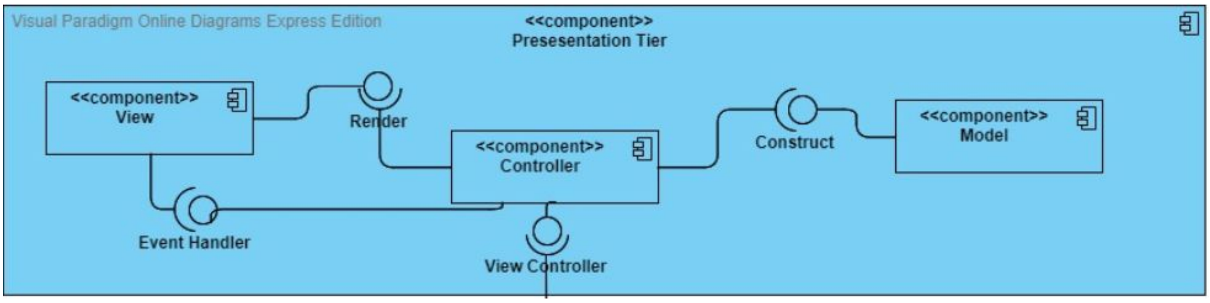

Figure 3: Presentation Tier Subsystem of 3-Tier Architecture

One of the subsystems that the application uses is the Presentation Subsystem. This subsystem, as it can be understood from its name, handles the interface and front end of the application. Just like many other similar purpose subsystems, The Presentation Subsystem adopts the MVC system due to such systems proven applicability and abundant sources with answers for problems that may arise.

The interface managed by the Presentation Subsystem is accessible through the mobile application. The interface consists of menu options, settings options, accessibility settings and all the other content within the application. This interface makes interacting with the  Logic Tier intuitive for the end user as it would give the end user the necessary tools for inputting and outputting content with safety.

Presentation Tier consists of three main components as the following;

### 4.1.1 Model Component

Model component handles the data that is relevant for the interface such as the buttons, the content and saved preferences. These can be called by the controller component when the view component requires content to show or simply some preferences to set in the next window.

### 4.1.2 View Component

View component is what the end user will be looking and interacting with. Although inputs such as button clicks and texts will be handled by the controller, the input will be sent to the controller via the view component in order to make our code scalable in the future. Also, the positioning of the buttons or text-boxes will be handled by the View Component as well. Particularly for our application, View component has to be very adaptable and intuitive in order to provide a friendly and smooth experience for our target audience which is disabled people.

### 4.1.3 Controller Component

The controller component is what manages the transfers between view and model components. If an image has to be displayed, the controller calls that image file from the model and sends it to the view in order to be used. On the other hand, if a button has been pressed, the listener inside the view component will transfer the input to the controller component which will decide what has to be done accordingly.

## 4.2 Logic Tier Subsystem

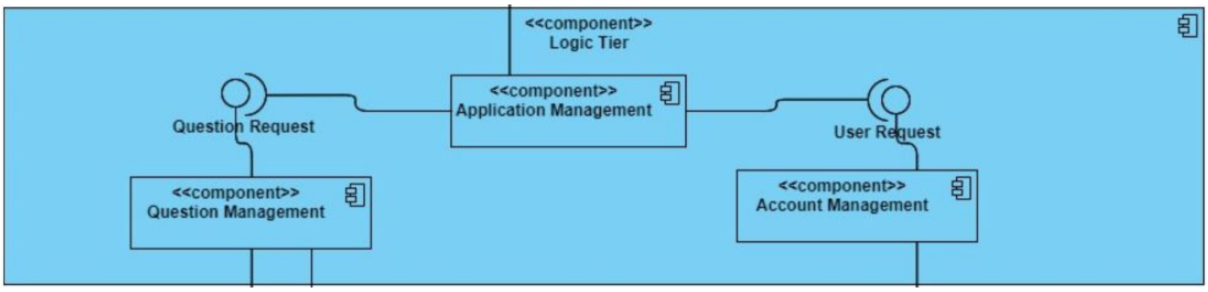

Figure 4: Logic Tier Subsystem of 3-Tier Architecture

Logic Tier is responsible for the main functionality of the system. This subsystem is the connection between the Presentation and Data tiers. Logic Tier is divided into 3 sub-components. These 3 sub-components are as follows:

### 4.2.1 Application Management Component

Application Management component is responsible for handling the requests from the client side. It receives requests from the Presentation Tier, handles these requests and responds back to the Presentation Tier. If the request is related to question or account database, it requests the necessary information from Question Management component or Account Management component and responds back to the Presentation tier.

### 4.2.2 Question Management Component

Question Management component handles question-related tasks. Services like retrieving or uploading questions are addressed by this component. When user enters a new question to be uploaded to the system, the information is transferred to the Question Database Management component and the new question is added to the database. When an existing question will be displayed to the user, this component sends requests to the Question Database Management component, gets the question and transfers the content of the question to Application Management component. This component also retrieves the information from the API Resource Management component.

### 4.2.3 Account Management Component

Account Management component handles account-related tasks. Services like user authentication and creating new account are addressed by this component. When the user enters the credentials of a new account to be created, the credentials are transferred to the Account Database Management component by this component and a new account is created. When the user enters the credentials of an existing account for logging in, the information is transferred to the Account Database Component, receives the result and transfers this result to the Application Management component.

## 4.3 Data Tier Subsystem

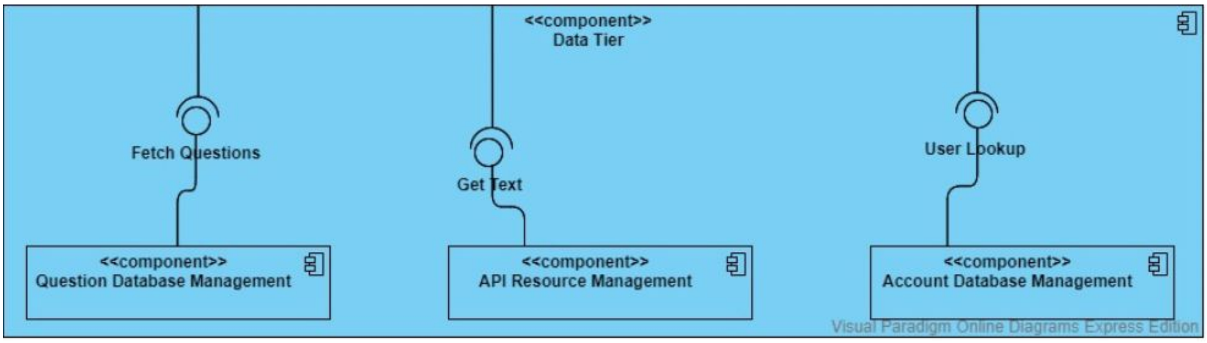

Figure 5: DataTier Subsystem of 3-Tier Architecture

Data Tier Subsystem is going to be deployed on server side. It is responsible for data management and data communication. Data Tier is secure since it can be accessed only from Logic Tier Subsystem. In this way, every modification or action happening in Data Tier must have a corresponding request made by the Logic Tier. Additionally, Data Tier is divided into 3 sub-components so as to achieve different kinds of functionality for different cases in the Database Management System. Those 3 components are as follows:

### 4.3.1 Question Database Management

Question Database Management is responsible for storing quiz questions and responding to calls coming from Logic Tier. When the user attempt to upload a question to the system by voice or text, detailed information such as question itself, one correct answer and two incorrect answers, category of the question and difficulty level will be inserted into the database table. All of that information coming from Logic Tier will be fetched by Data Tier.

### 4.3.2 API Resource Management

API Resource Management is responsible for retrieving the text which will be captured from a photo and storing it. There should be a single scenario for API Resource Management to be called.

- When the user try to upload a question by an image from his/her device, there will be request to our python module that extracts the text from an image on the server.

### 4.3.3 Account Database Management

Account Database Management is responsible for managing the user-specific data accesses and data modifications. Since this layer is secure, it can be accessed only if there exists a request made by the Logic Tier. For instance, when a new user decides to play our application, (s)he will be asked to login first. During the creation of a new account, user-specific data will be inserted into the corresponding database table in our database management system.

# 5. New Knowledge Acquired and Learning Strategies Used

Android Studio and android program structure, basic implementations of Talkback,Voice Control,Text-to-speech and sending image with post request learned via video tutorials from Youtube, question and answers from Stackoverflow and source codes from github. OCR with PyTesseract[8] and creating API with Flask[9] learned via web documentations.

# 6.Reference

[1]Text To Speech Reader

https://ttsreader.com. [Accessed: 9-Oct-2019].

[2] Voice Control:Everything you need to know

https://www.imore.com/voice-control-everything-you-need-know. [Accessed: 9-Oct-2019].

[3] Android'i TalkBack ile kullanmaya başlama

https://support.google.com/accessibility/android/answer/6283677?hl=tr. [Accessed: 9-Oct-2019].

[4] Kahoot! https://kahoot.com. [Accessed: 14-Oct-2019].

[5] QuizUp https://www.quizup.com/en. [Accessed: 14-Oct-2019].

[6] Trivia Crack https://www.triviacrack.com. [Accessed: 14-Oct-2019].

[7] Firebase https://firebase.google.com. [Accessed: 31-Dec-2019].

[8] pytesseract PyPI https://pypi.org/project/pytesseract/. [Accessed: 25-Nov-2019].

[9] Flask https://www.palletsprojects.com/p/flask/. [Accessed: 25-Nov-2019].