Bilkent University

Department of Computer Engineering

Senior Design Project

EyeSight

High-Level Design Report

Group Members: Cemil Şişman, Derviş Mehmed Barutcu, A A M Jubaeid Hasan Chowdhury, Mustafa Azyoksul, Onur Mermer

Supervisor: Varol Akman

Jury Members: Fazlı Can, Hamdi Dibeklioğlu

High-Level Design Report
May 22, 2020

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.

TABLE OF CONTENTS

1. Introduction

In this technological era, most of the productions and developments are aimed at the entertainment business. In our opinion, technology should be used to help handicapped people as much as it's used for entertainment purposes.

Using this motto, EyeSight is intended to hold visually impaired people's hand to make their life less challenging. Despite the rapid development of computers, visually impaired people still use a stick to navigate themselves through obstacles. EyeSight is imagined to become a new eye for those people. Using the camera of the smartphones, EyeSight will describe the user's surroundings such as objects, walls, humans etc. to them in realtime and also navigate them to where they want to go, even for small distances such as going from kitchen to the bedroom etc. Users will also be able to register their relatives, friends and family members to EyeSight. The app will recognize human faces and will inform users specifically on who those people are.

EyeSight will have a user friendly UI for visually handicapped people. It will be accomplished by using voice overlay and taking voice inputs from the users which will be used to help them interact with the app. We are determined to ensure that visually impaired users will have no issue in using the app without any help from other people. Because the deadlines of this project are limited, we decided to implement EyeSight just for indoor usage. EyeSight will initially not be suited for outdoors because of the vital risks that might occur.

1.1. Purpose of the System

Our purpose for creating this system is to create an app which can create a sense of awareness by using mobile phone cameras and sensors to gather data about users' surroundings and inform them via sound. This application mainly targets users who have visual impairment thus the system should be accessible to such users with proper UI and voice overlay.

Our intention is to make smartphones an accessible tool for people who have visual impairment as a cane which has more capability and interaction than a normal cane to make the experience of walking more engaging.

1.2. Design Goals

1.2.1. Accessible and Usable System

Since our main target group as users are people with visual impairment, our system's main features should be accessible and easy to use by those users. Our UI elements should be designed mainly for our main target group and voice overlay to give proper feedback and to get voice input from users should be available in the system.

1.2.2. Response Time

Our system includes vision as video and image, sound as input and output data which are used for our main features and processing those data and giving proper feedback to users in real time is our system's main priority. Thus a major design goal for this system design is to provide main features in real time to the user. Processing cost of data transfer and processing the data in-hand is considered to provide higher performance to the user during system design process.

1.2.3. Safety

Our system should have a low margin of error during analysis and returning feedback to avoid any undesirable circumstances that a user with visual impairment could experience. Thus our system should ensure that created data and feedback by analysis is reliable to be used by the user.

1.2.4. Modifiable System

Our system should be designed open to improvement for new features which might be added. In order to ensure that a centralized database and server application is needed which will enable us to implement features which involve data communication. Our subsystems also should be designed as a modular system which involves a central controller subsystem so new features might be added as new subsystems in order to ensure modifiability.

1.2.5. Low Space Requirement vs. Speed and Offline Availability

In order to reduce the size of the application in the device, we will utilize a server to handle vision analysis rather than handling it in the mobile device. Compared to standard computers, smartphones have a limited amount of disk space available to applications. Since we are going to utilize a server for database system to keep records for the users, we will utilize the same server to handle data analysis and return feedback to client applications to reduce the size of applications that are going to be downloaded by the user to their smartphone. That may create an issue for performance criteria of the application but with contemporary internet connection capabilities this may be avoided. This decision will also limit us to use the application only when the internet is available but it may be ignored considering contemporary internet connection capabilities and huge disk space that will be allocated if we are going to handle vision analysis in the local device.

1.3. Definitions, Acronyms, and Abbreviations

HTTP: (Abbv) Hypertext Transfer Protocol. Application-layer protocol for data communication.

TCP/IP: (Abbv) Transmission Control Protocol/ Internet Protocol. Transport-layer protocol for data communication.

1.4. Overview

Eyesight is an object detection system which aims to be used by the people with visual impairment. It will be implemented as a smartphone app for android to make it accessible to use. Main feature of the system is to provide a constant stream of data about the environment to the user which has visual impairment to increase awareness of the user about its environment with minimal effort. In order to decrease the effort of the user about usage of the app properly, a user interface for a person with visual impairment will be prepared which includes interaction with smartphone screen, volume buttons on the phone and proper audio feedback together with text-to-speech features. In addition to the main data feed provided via voice by the app, a navigation system for indoors will be implemented to help users to navigate and give specific room informations to inform the user about the environment. In order to give more information about the people in the environment, specifically to recognize a family member, the data of family members with their face images will be kept in the app to recognize them and provide that information to the user. During all the user interaction and feedback to the user both audio feedback and text-to-speech features will be implemented as an audio overlay system in the app to make the app easy to use.

Object detection feature should work with a smartphone with a camera that can record a video. In order to gain the sense of distance, a separate sensor should be used since proximity sensors of the smartphones have very short ranges for our use. Gyroscopes of smartphones will be used to ensure that the user holds the phone in a suitable position to gather visual data properly. App should give the object information together with any other obstacle user faces in their way to give the sense of indoor environment in audio format. Navigation feature in the system should work with specific symbols in the environment that are prepared and registered to the app beforehand so it can process the location information of the user, give the data about the room they are in and navigate them through their destination.

2. Current Software Architecture

There is an app called Seeing AI [1] that is developed by Microsoft. It has a similar objective with our app EyeSight but overall architecture has major differences. Firstly, unlike EyeSight, Seeing AI doesn't provide a voice interface which allows visually impaired users to interact with the app without being in need of others' help. Another difference is that Seeing AI requires its users to take a photo and then analyzes this photo to produce a voice output. EyeSight will do that in real time with the camera of the phone. Seeing AI provides some extra features such as converting text to speech and reading documents which are very helpful. However, these features are hard to provide in real time. Hence we are not including those types of features in EyeSight. Also, Seeing AI has separate tabs called "Product" and "Person". You need to go to those tabs to allow the app to detect the corresponding objects. In EyeSight, there is such separation. Our detection algorithm will distinguish the objects and people.

3. Proposed Software Architecture

3.1. Overview

Proposed software architecture for the system is designed in a way to accommodate our design goals and criteria in an effective manner. Whole system is composed of a client app which is designed to work on a smartphone, a server app which listens for new requests that are sent from client apps and a database that is connected to the server application to hold records of user data in a centralized structure. Specification details regarding system architecture are explained in corresponding sections below.

3.2. Subsystem Decomposition

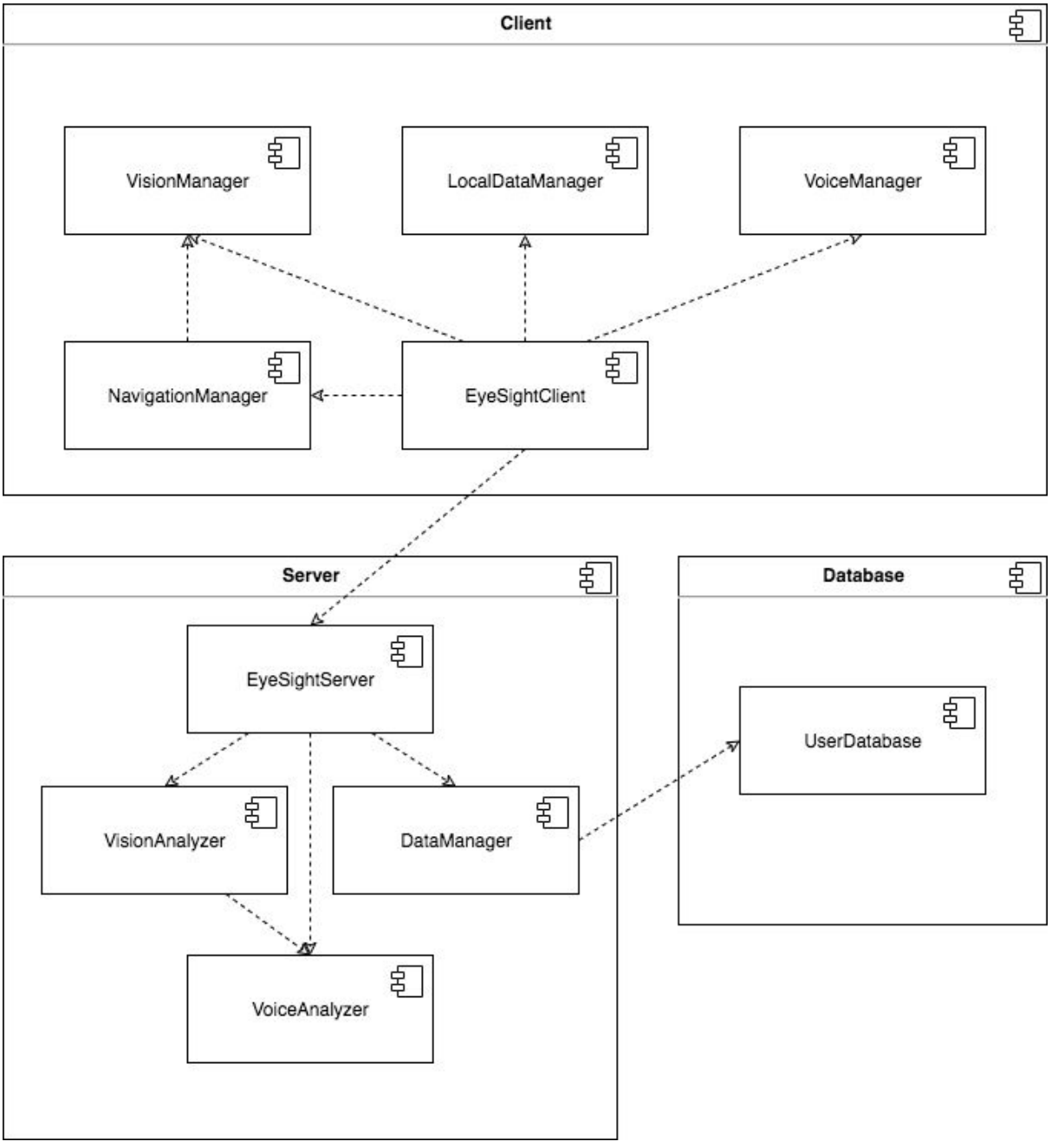

Figure 1: Decomposition Diagram

We have three subsystem components which are Client, Server and Database. Since we have three different subsystems, we can easily make changes which provides flexibility to us.

Client subsystem is the system that users can interact with and make requests for the Server subsystem. Server subsystem handles the different requests from the client subsystem such as visual requests to be analyzed and sent back to the client. Database subsystem is for holding the user information that we need for the application such as google account basic information, phone numbers and pictures that application requests.

3.3. Hardware/Software Mapping

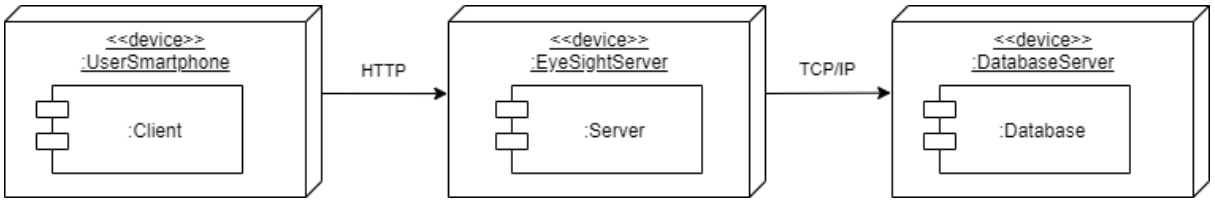

Figure 2: Hardware/Software Mapping

In our system there are 2 main applications assigned as the client application, which runs on individual users' smartphones, and server application which runs on the server machine (EyeSightServer). Client application manages main interactions with users, outputs the feedback to users via voice overlay and makes requests to server applications for vision analysis or to get user data from the database. Server application handles vision analysis requests on the server machine and returns feedback as text which is going to be turned into speech in client application. Server application also handles get requests for user data by sending query to database management system. We will use MySQL as the database management system to hold records of the data of users.

3.4. Persistent Data Management

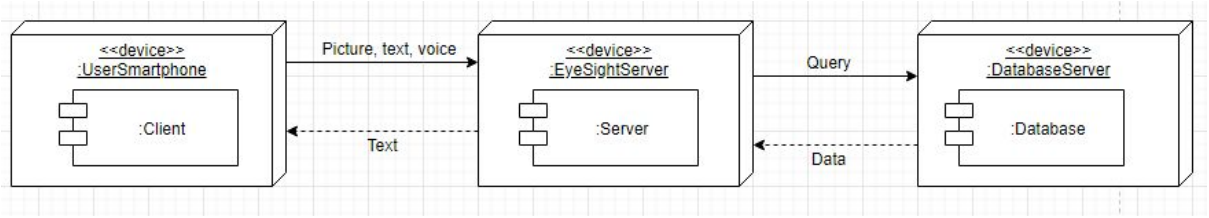

Figure 3: Persistent Data Management

EyeSight is a mobile application that we assumed is used by only one person per phone so we need to hold some of the information for a long time such as account information or profile information. Other than this, as EyeSight processes the camera inputs in real time, we need to send the images to the server system in some way. To achieve it persistently and securely, we will store related information in database systems. We will use the database service that Google provides. We have decided to use it because it is accessible from any devices and because we are developing an android application, using Google's services may be the best choice for us. We are going to use some of the features that database services provide to us. Indexing related information fastens the queries that our application will make.

EyeSight is going to analyze real time inputs from the camera so holding all the data coming from the camera will increase the storage needs. Therefore, we are not going to save the videos that we processed. We will store the results of the analysis for further analysis from them.

3.5. Access Control and Security

To use the EyeSight, the users need to enter their Google account. Since they are using an android phone, the users have a Google account and they can log in with the account that they wish. EyeSight will check the entered information using Google API. The system stores the account information under Google database system. We will not store the account information locally. When they log in, they can add requested information, and these information will be available for each time, when users use the application.

We are careful about users' privacy. We are not going to store any sensitive data related with the users locally. Because EyeSight will use Google's API, we need to meet some security levels required by Google. The related user data will be stored on an external database system, excessing these information needs to meet the same level of security.

3.6. Global Software Control

Our main flow is designed as event-driven control. When the users of EyeSight use the application for the first time, they must log in with their Google account. Logging in with the Google account will ease our job to create a personal account. After they Log in with Their Google account, it is requested to fill in some of the information for the profile like relative name, photo of them and phone numbers. This part is optional but we recommend to fill at least some of the parts. After finishing all these parts users can start using the application.

The communication with the server starts in the beginning of the application to check the profile information and retrieve the related data from the database. The main process begins after the user presses the start button. Since Google' services have limitations, we will implement another model for recognizing the objects, symbols and faces. The two services will work concurrently, we choose this architecture to increase the analysis speed.

As for the profile settings, the users can edit their profile information anytime they want from the settings. They can change the emergency contact, add relatives to be recognized by the EyeSight later, delete who they want. They can change the recognized object count from the settings. If they want they can also log out and log in with a different account.

3.7. Boundary Conditions

3.7.1. Initialization

For the first time using EyeSight, the user needs to download the app and create an account. The user needs to give his/her Google account. The user needs internet connection for signing up. The companion should be the person doing the sign up.

3.7.2. Termination

The user can terminate the app by logging out or quitting the app.

3.7.3. Failure

The app does not work if there is no internet connection. Also, the internet connection must have sufficient throughout for the app to deliver good enough performance. The app can not detect any object if the environment is too dark. In that case, he/she will be recommended to open the camera flash light. Also, as the application uses a camera and microphone, it consumes a lot of power and the phone might heat up after long time usage. Hence, continuous long time usage is not recommended.

4. Subsystem Services

This section will explain each subsystem of the system.

4.1. Client

The client subsystem will consist of EyeSight Client, vision manager, voice manager, navigation manager and local data manager.

4.1.1. EyeSight Client

EyeSightClient is the main controller subsystem which includes view services, model services and controller unit as a way to control application's executions and handle external events. It interacts with other client subsystems to provide different features of application for user and also communicates with server application to request vision and voice analysis and get user data. This Client subsystem has two divisions:

(i) The network component which interacts with the server to send and receive data.

(ii) All the views of the app.

Network Component: All the data to and from the server goes through this component. It has channels with Vision Manager, Voice Manager, and Navigation Manager to exchange data. A single point of network communication ensures simple and homogeneous communication with less security issues.

SignUp/Login View: Collects sign up or login information to be sent to the server for authentication.

Startup Page View: Collects information about family members or places when the app is used for the first time.

Settings View: Display and update any settings information and store them in the server.

Navigation View: Displays the options for navigation. Takes navigation destination as input.

Main Page View: Displays the main menu.

Main Function View: Opens the camera and starts vision analysis.

Emergency Contact View: Display and update emergency contact information.

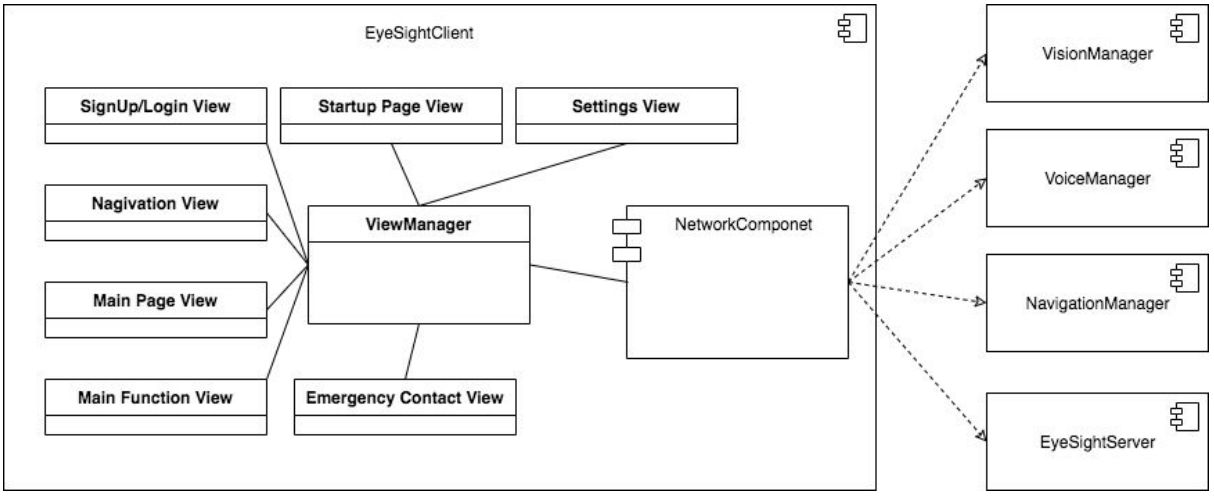

Figure 4: EyeSightClient services and its interaction with other subsystem

4.1.2. Vision Manager

VisionManager collects real time images from the camera and sends it to the server. The VisionManager decides how often to send images to the server and which format to send. The images will be downsized before sending. For different internet speed the images may be downsized differently.

4.1.3. Voice Manager

VoiceManager collects voice input from the microphone if the user is using voice command. It then sends the audio file to the server. Also, all the outputs from VisionAnalyzer are transferred to VoiceAnalyzer for speech synthesis which are then sent to VoiceManager to be played aloud.

4.1.4. Navigation Manager

NavigationManager handles navigation with symbols. In this mode the camera is only looking for particular symbols. Hence, the image taken will be processed differently before sending to the server than normal object detection. The Navigation Manager will also need to get navigation information of the user either from persistent data or the database.

4.1.5. Local Data Manager

Local data manager is the data related component in the EyeSight's application side. It manages the local database by providing other parts of the application interfaces to write, read, update and delete data.

4.2. Server

This section of the system consists of EyeSight server, vision analyser, voice analyser and data manager.

4.2.1. EyeSight Server

EyeSight server handles 3 essential API calls made from the mobile application: speech-to-text, vision analysis and text-to-speech. The server also handles standart authentication functionality and persistent, non-local data storage. The server follows the RESTful design architecture to be able to serve many people at the same time with no conflicts.

4.2.2. Vision Analyzer

Vision analyzer is a server side visual processing service. Its purpose is to process images to find useful information inside frames and return the information the system thinks is important. It is a machine learning model which is based on a highly functional, pre-trained model and specialized to detect daily life objects and obstacles, objects that are not in their normal positions and special, app-specific signs.

4.2.3. Voice Analyzer

Voice analyzer is a server side sound processing service. It converts the simple texts which are feedbacks from vision analysis process to sound recording as output and recordings of sounds to text files which are input from the user to interact with the client application.

4.2.4. Data Manager

Data manager is the data related component in the EyeSight's server side. It manages the persistent database by providing other services interfaces to write, read, update and delete data.

4.3. Database

The database subsystem has only a user database.

4.3.1. User Database

This subsystem mainly includes a database management system which is organized to hold records of user account data with saved family member information. This way we can provide user data to another phone by just logging in that phone instead of limiting the user to use one smartphone and creating a new account for each time. This will also enable us to modify the system for new features which might be added which requires a central database to implement.

5. New Knowledge Acquired and Learning Strategies Used

We have done research on how to store data locally on an Android app. And then we did an efficiency analysis on storing data locally versus storing data in the server. Based on that analysis, we made decisions regarding which data to store in the server and in the local storage. Since some of us are new to developing an Android app via Android Studio, we mainly used Androids' own docs for research purposes [2].

We will provide text-to-speech and speech-to-text features in EyeSight. We plan on using Google Cloud's API for that purpose [3]. Cloud has a well explained documentation which we have used to learn about its requirements, capabilities and limitations. We have also found out about another Google service which is called Firebase [4]. This service is integrated into the app and provides features such as user authentication, storage, a database app and many other services.

For the image detection and classification, we were planning on using Cloud's Vision AI [5]. After doing some research on that, we have decided not to use it because it had a free use limit of 1000 images monthly. After that limit, it required payment. Hence, we will create our own model and do the detection and classification on a separate server.

6. References

[1]: “Seeing AI App from Microsoft,” *Seeing AI App from Microsoft*. [Online]. Available: https://www.microsoft.com/en-us/ai/seeing-ai. [Accessed: 22-May-2020].

[2]: “Documentation : Android Developers,” *Android Developers*. [Online]. Available: https://developer.android.com/docs. [Accessed: 22-May-2020]

[3]: “Cloud Text-to-Speech Documentation | Google Cloud,” *Google*. [Online]. Available: https://cloud.google.com/text-to-speech/docs?hl=tr. [Accessed: 22-May-2020].

[4]: “Documentation | Firebase,” *Google*. [Online]. Available: https://firebase.google.com/docs?gclid=CjwKCAjw8J32BRBCEiwApQEKgS9kFhIlKvglPdbqCoUwcRrYJzyoZvVEkwQuT4t7hZtrEedrvFfgjxoCyFoQAvD_BwE.

[5]: “Vision AI | Derive Image Insights via ML | Cloud Vision API,” *Google*. [Online]. Available: https://cloud.google.com/vision. [Accessed: 22-May-2020].