

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**DETAILED DESIGN SPECIFICATION  
CSE 4317: SENIOR DESIGN II  
SPRING 2021**



**ALS EYE TRACKING GROUP  
ALS EYE TRACKING APP**

**ANTHONY VARDARO  
MICHAEL KOSTA  
ZIXIU SU  
THANHTHAO LE**

## REVISION HISTORY

Revision	Date	Author(s)	Description
1	2.21.2021	AV, MK, ZS, TL	document creation and finished

## CONTENTS

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>System Overview</b>	<b>5</b>
<b>3</b>	<b>Eyetracking Hardware Layer Subsystems</b>	<b>6</b>
3.1	Layer Hardware . . . . .	6
3.2	Layer Operating System . . . . .	6
3.3	Layer Software Dependencies . . . . .	6
3.4	Subsystem 1 Interface . . . . .	6
<b>4</b>	<b>Electron Layer Subsystems</b>	<b>8</b>
4.1	Layer Hardware . . . . .	8
4.2	Layer Operating System . . . . .	8
4.3	Layer Software Dependencies . . . . .	8
4.4	Subsystem 1 Electron Main . . . . .	8
4.5	Subsystem 2 Electron Render . . . . .	9
4.6	Subsystem 3 Eyetracking . . . . .	10
4.7	Subsystem 4 Calibrate . . . . .	11
<b>5</b>	<b>Cloud Layer Subsystems</b>	<b>13</b>
5.1	Layer Hardware . . . . .	13
5.2	Layer Operating System . . . . .	13
5.3	Layer Software Dependencies . . . . .	13
5.4	Subsystem 1 Firebase Authentication Web Hosting . . . . .	13
5.5	Subsystem 2 Firebase Realtime Database . . . . .	14

## LIST OF FIGURES

1	ALS Data Flow Diagram . . . . .	5
2	Eyetracking Hardware Subsystem 1 Interface . . . . .	6
3	Electron Subsystem 1 Electron Main . . . . .	8
4	Electron Subsystem 2 Electron Render . . . . .	9
5	Electron Subsystem 3 Eye-Tracking . . . . .	10
6	Electron Subsystem 4 Calibrate . . . . .	11
7	Cloud Subsystem 1 Web Hosting . . . . .	13
8	Cloud Subsystem 2 Firebase Database . . . . .	14

## 1 INTRODUCTION

Our product is an Eye-Tracking Keyboard used to help people with ALS. The Keyboard is used to improve communication with anyone who is only able to move their eyes. They use their eyes to spell out words on a screen, forming sentences.

Originally we were going to have three different layers in our data flow, the Eye-Tracking Hardware, the Keyboard, and the Menu. Instead, we decided to merge the Menu with the Keyboard screen and we decided to add Cloud Services, using this for the database. We also decided to use an Electron App for the Keyboard.

## 2 SYSTEM OVERVIEW

The Eye Tracking Hardware has not changed, it is still the Tobii eye-tracker, the users PC, the calibration feature, and the Interface. This is what is capturing eye data from the user, to convert that into a cursor on the screen. We can use this cursor to point to the appropriate letter or setting the user is trying to press. The Electron is the interface we used to implement the user keyboard. It uses a timer that is synced with the Eye Tracker, when it hovers over a letter for the set time, it will capture the letter. There are other features that are used in conjunction with, like calibration and copy/paste. Finally, the third is the Cloud Services, which are used to store the database, which is also used for the lettering prediction and for storing user data.

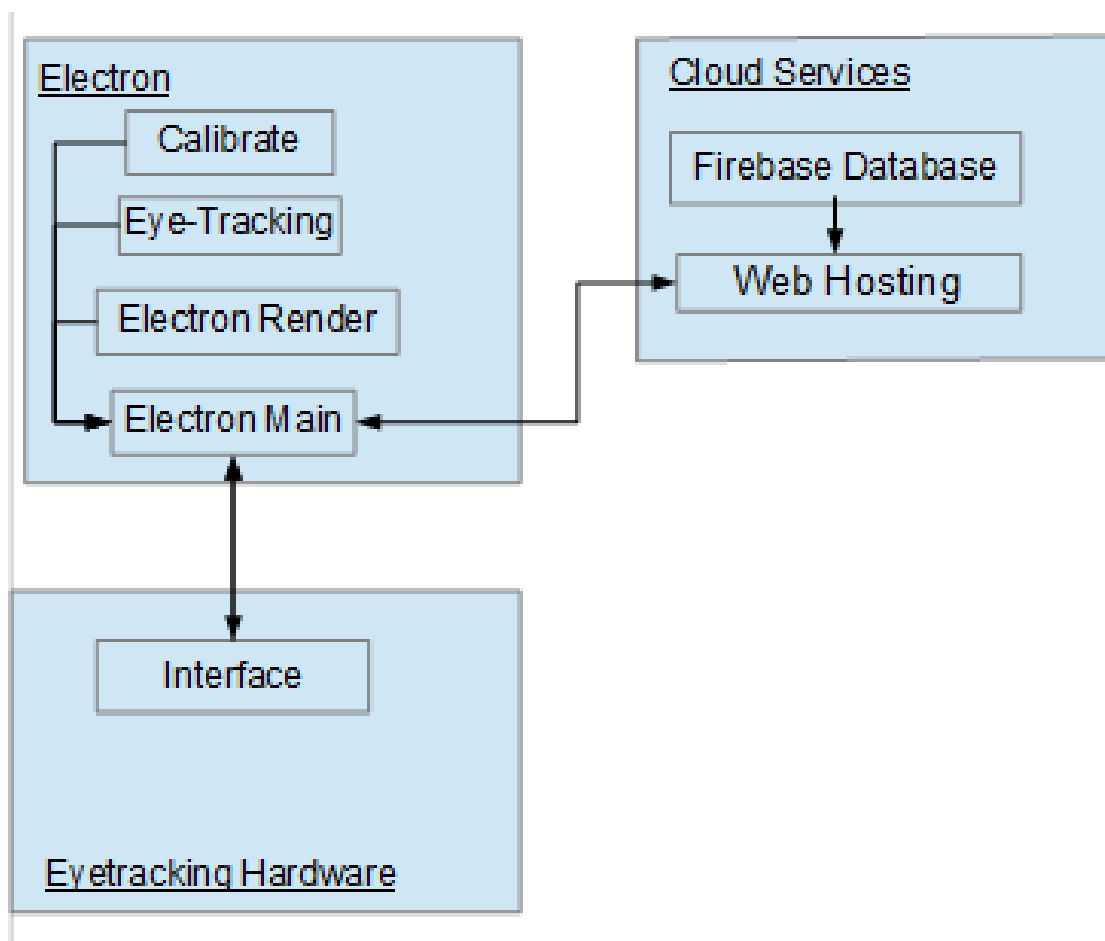


Figure 1: ALS Data Flow Diagram

### 3 EYETRACKING HARDWARE LAYER SUBSYSTEMS

This layer includes essential hardware components and their drivers/operating systems, setting up a basic working environment for our virtual keyboard app.

#### 3.1 LAYER HARDWARE

This layer contains a computer that runs Windows operating system with a Tobii Eye Tracker 5 device and a display screen.

#### 3.2 LAYER OPERATING SYSTEM

The operating system required by this layer is Windows 10 (x64).

#### 3.3 LAYER SOFTWARE DEPENDENCIES

The only Dependency required is the Tobii Experience.

#### 3.4 SUBSYSTEM 1 INTERFACE

The first subsystem in this layer is the Tobii Eye Tracker 5 and its dependent software. Our app uses Tobii as an input device to replace the need of a physical keyboard. The support of the Tobii device, most importantly calibration, is provided by Tobii Experience software.

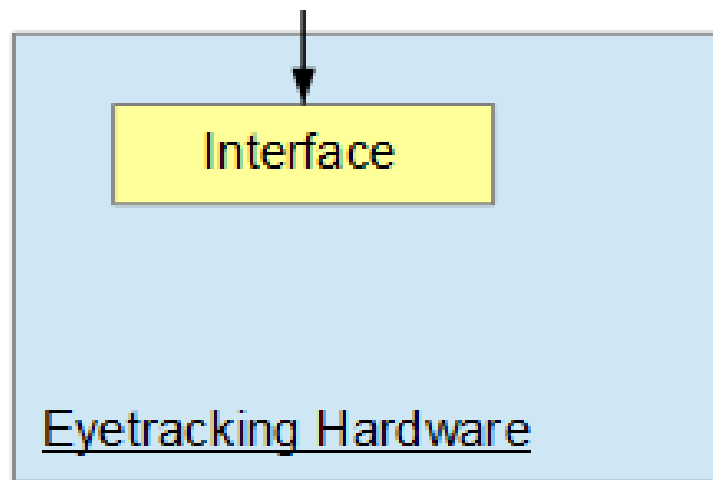


Figure 2: Eyetracking Hardware Subsystem 1 Interface

##### 3.4.1 SUBSYSTEM HARDWARE

This subsystem involves a Tobii Eye Tracker 5 device.

##### 3.4.2 SUBSYSTEM OPERATING SYSTEM

None

##### 3.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

For Tobii Eye Tracker to work properly, the Tobii Experience app is required. Download link can be found here: <https://gaming.tobii.com/getstarted/>

##### 3.4.4 SUBSYSTEM PROGRAMMING LANGUAGES

None

### **3.4.5 SUBSYSTEM DATA STRUCTURES**

None

### **3.4.6 SUBSYSTEM DATA PROCESSING**

None

## 4 ELECTRON LAYER SUBSYSTEMS

This layer contains the main program of our virtual keyboard and the components that run on an users machine locally. Our application is developed using the Electron framework.

### 4.1 LAYER HARDWARE

None

### 4.2 LAYER OPERATING SYSTEM

The required operating system for this layer is Windows 10 (x64).

### 4.3 LAYER SOFTWARE DEPENDENCIES

Node.js 14 or greater. Node-Gyp 7 or greater. Foreman 3 or greater.

### 4.4 SUBSYSTEM 1 ELECTRON MAIN

Electron Main is a process that spawns the Electron Render, Eyetracking, and Calibrate child processes. It is considered the entry point of the application. Electron Main is primarily responsible for data flow between processes via IPC.

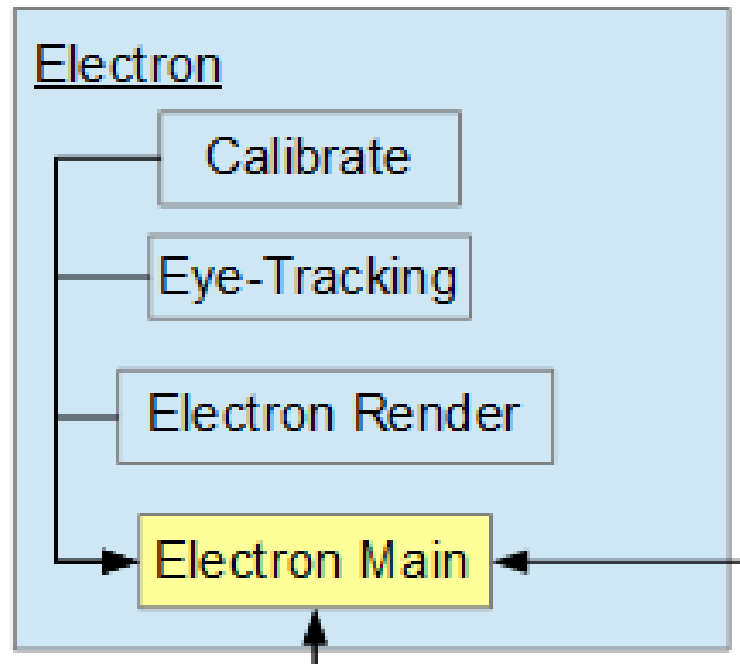


Figure 3: Electron Subsystem 1 Electron Main

#### 4.4.1 SUBSYSTEM HARDWARE

This subsystem contains a computer that runs Windows operating system with a Tobii Eye Tracker 5 device and a display screen.

#### 4.4.2 SUBSYSTEM OPERATING SYSTEM

Windows 10.

#### 4.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

Relevant dependencies are listed in the application package.json.



#### 4.4.4 SUBSYSTEM PROGRAMMING LANGUAGES

Javascript (within Node.js).

#### 4.4.5 SUBSYSTEM DATA STRUCTURES

**TobiiRegion** Class representing the packet sent by Eyetracking process. `TobiiRegion.js`

#### 4.4.6 SUBSYSTEM DATA PROCESSING

The Electron Main process does minimal data processing, as its primarily tasked with pipelining data to child processes.

### 4.5 SUBSYSTEM 2 ELECTRON RENDER

The Electron Main process is the User Interface (UI) of the application. It sends relevant coordinate data to Electron Main, and in return Electron Main emits events when the user focuses on regions on the UI.

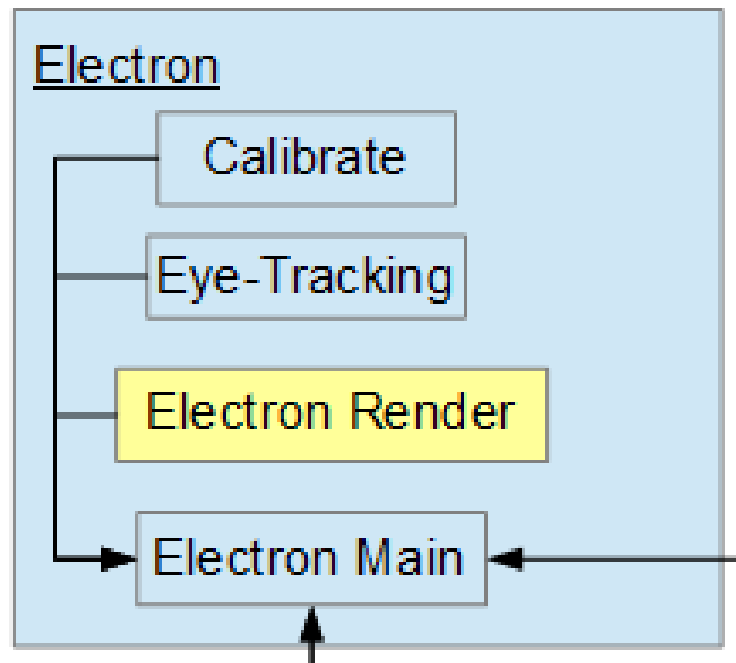


Figure 4: Electron Subsystem 2 Electron Render

#### 4.5.1 SUBSYSTEM HARDWARE

This subsystem contains a computer that runs Windows operating system with a Tobii Eye Tracker 5 device and a display screen.

#### 4.5.2 SUBSYSTEM OPERATING SYSTEM

Windows 10

#### 4.5.3 SUBSYSTEM SOFTWARE DEPENDENCIES

Relevant dependencies are listed in the application package.json.

#### 4.5.4 SUBSYSTEM PROGRAMMING LANGUAGES

Javascript (within Node.js).

#### 4.5.5 SUBSYSTEM DATA STRUCTURES

**Keyboard Wrapper** Virtual keyboard that the user interacts with by focusing on the keyboard keys. KeyboardWrapper.js

#### 4.5.6 SUBSYSTEM DATA PROCESSING

The Electron Main process streams gaze event notifications to Electron Render. The Electron Render processes these events to determine what key is being focused on, how long the user has focused on a key, and computing dwell time.

#### 4.6 SUBSYSTEM 3 EYETRACKING

The Eyetracking process is a child process of Electron Main. It communicates with the Tobii Eyetracking Hardware in real-time. The Electron Main process sends keyboard coordinate data to Eyetracking, and the Eyetracking process emits an event if the user's gaze focuses within a specified coordinate region. It also emits an event if the user's gaze is idle, or the Eyetracker cannot detect the user.

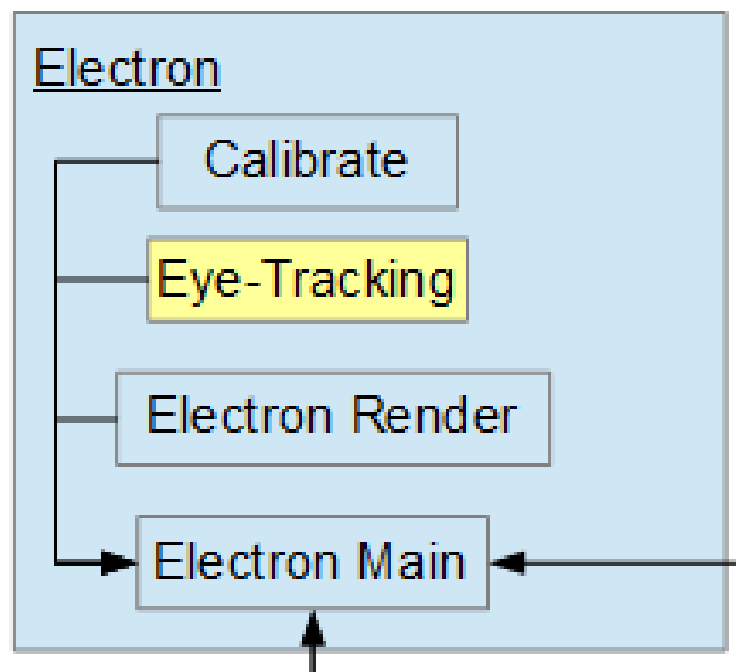


Figure 5: Electron Subsystem 3 Eye-Tracking

##### 4.6.1 SUBSYSTEM HARDWARE

This subsystem contains a computer that runs Windows operating system with a Tobii Eye Tracker 5 device and a display screen.

##### 4.6.2 SUBSYSTEM OPERATING SYSTEM

Windows 10

##### 4.6.3 SUBSYSTEM SOFTWARE DEPENDENCIES

Node-gyp 7 or greater.

##### 4.6.4 SUBSYSTEM PROGRAMMING LANGUAGES

C++ and Javascript (within Node.js).

#### 4.6.5 SUBSYSTEM DATA STRUCTURES

**Screen** Screen is a class that stores metadata of the users screen. It also exposes functions that execute user-defined callbacks when an event is emitted. .Screen.h

#### 4.6.6 SUBSYSTEM DATA PROCESSING

The Eyetracking process collects a stream of raw gaze data from the Tobii Eyetracking hardware, and determines whether the gaze point fits the user specified parameters. If so, it executes a callback function specified by the Electron Main process.

#### 4.7 SUBSYSTEM 4 CALIBRATE

The Calibrate process is a child process of Electron Main. Electron Main sends a message to the Calibrate process to kickstart the calibration routine.

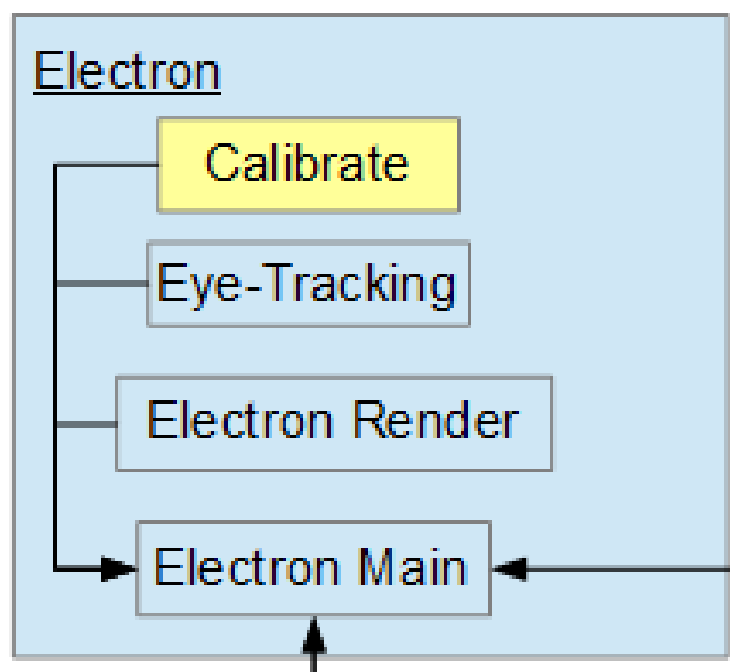


Figure 6: Electron Subsystem 4 Calibrate

##### 4.7.1 SUBSYSTEM HARDWARE

This subsystem contains a computer that runs Windows operating system with a Tobii Eye Tracker 5 device and a display screen.

##### 4.7.2 SUBSYSTEM OPERATING SYSTEM

Windows 10

##### 4.7.3 SUBSYSTEM SOFTWARE DEPENDENCIES

None

##### 4.7.4 SUBSYSTEM PROGRAMMING LANGUAGES

C++ and Javascript (within Node.js).

#### **4.7.5 SUBSYSTEM DATA STRUCTURES**

None

#### **4.7.6 SUBSYSTEM DATA PROCESSING**

The Calibrate process constantly listens for an Idle event emitted by the Eyetracking process. When it detects this event, it asks the user to calibrate their hardware again.

## 5 CLOUD LAYER SUBSYSTEMS

This layer includes all the cloud services required by the local app such as Google Firebase, and support information on our website.

### 5.1 LAYER HARDWARE

None

### 5.2 LAYER OPERATING SYSTEM

None

### 5.3 LAYER SOFTWARE DEPENDENCIES

The only dependency is access to the Internet.

### 5.4 SUBSYSTEM 1 FIREBASE AUTHENTICATION WEB HOSTING

Firebase Authentication is used to manage user identification.

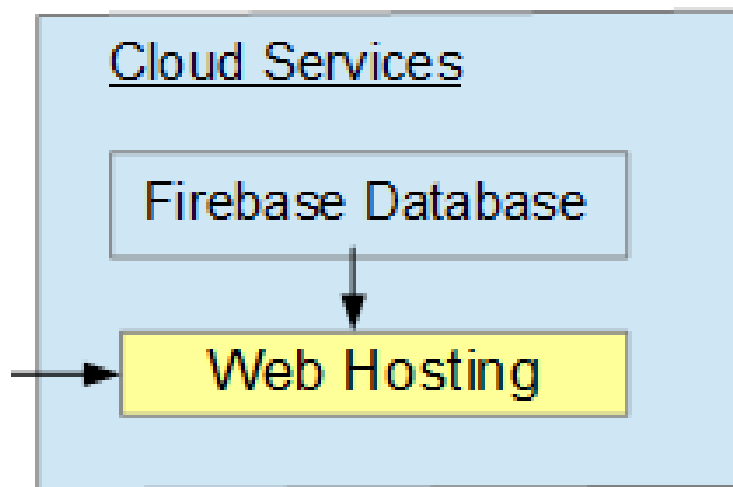


Figure 7: Cloud Subsystem 1 Web Hosting

#### 5.4.1 SUBSYSTEM HARDWARE

None

#### 5.4.2 SUBSYSTEM OPERATING SYSTEM

None

#### 5.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

Firebase Authentication SDK.

#### 5.4.4 SUBSYSTEM PROGRAMMING LANGUAGES

Javascript

#### 5.4.5 SUBSYSTEM DATA STRUCTURES

None

#### 5.4.6 SUBSYSTEM DATA PROCESSING

None

#### 5.5 SUBSYSTEM 2 FIREBASE REALTIME DATABASE

Firebase Realtime Database is used to manage user data storing.

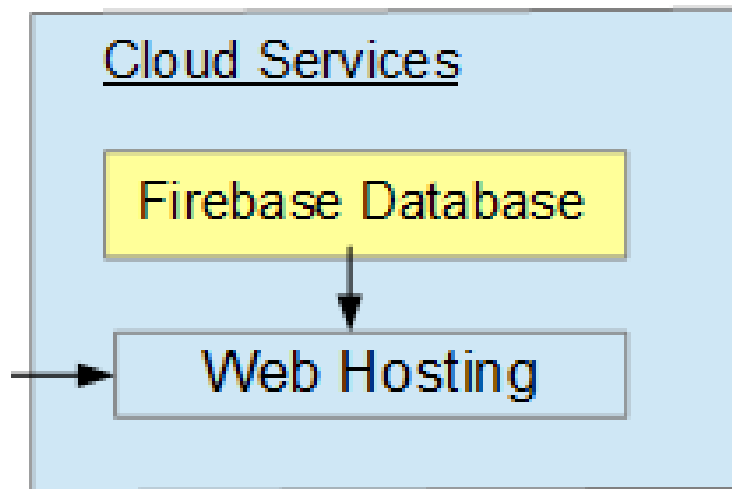


Figure 8: Cloud Subsystem 2 Firebase Database

#### 5.5.1 SUBSYSTEM HARDWARE

None

#### 5.5.2 SUBSYSTEM OPERATING SYSTEM

None

#### 5.5.3 SUBSYSTEM SOFTWARE DEPENDENCIES

Firebase JS SDK is used to communicate with Firebase APIs.

#### 5.5.4 SUBSYSTEM PROGRAMMING LANGUAGES

Mainly Javascript among other languages it supports.

#### 5.5.5 SUBSYSTEM DATA STRUCTURES

Firebase Database is a NoSQL realtime database. Data is structured as a JSON tree. For more information: <https://firebase.google.com/docs/database/web/structure-data>

#### 5.5.6 SUBSYSTEM DATA PROCESSING

Firebase Database is a real time cloud service and works in an asynchronous fashion, read/write procedure may differ from other databases. For more information: <https://firebase.google.com/docs/database/web/read-and-write>