EyeXam Final Report

Shunan Chen, 1001131376 (Apper)
Yi Li, 1001251860 (Programmer)
Xinyi Lin, 1001091132 (Programmer)

Word count: 2263

1. Introduction

1.1 What & Why

EyeXam is an iPad application designed for testing and recording visual acuity. We create this app to reproduce the eye testing procedure on the iPad and make it more convenient than traditional eye exams. The traditional way of eye examination is conducted by hanging the paper-based eye chart (usually the eye chart full of English letters, called 'the Snellen Chart') on the wall, users standing in front of the eye chart at a certain distance and trying to read the optotypes on the eye chart until they cannot see anything smaller. Such test requires a large space and is time consuming.

EyeXam reproduce the testing procedure by displaying eye chart optotypes one by one on the iPad screen. Like traditional eye tests, user stands in front of the iPad screen and tell which direction the optotype (the letter E) is facing by pointing to that direction. User will hold Node sensor in hand for motion detection. The Node sensor is connected to the iPad via Bluetooth. Fig 1.1 shows the rough design of the application.

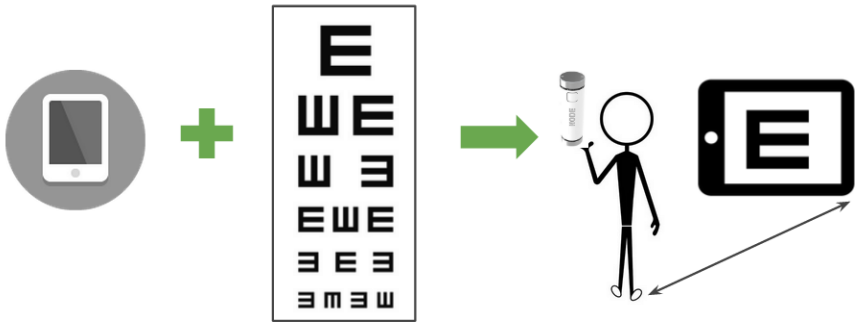

Fig 1.1 Design of EyeXam

The testing eye chart we use is E tumbling chart (also known as E Standard Logarithm Eyesight Table, depicted in Fig 1.2), which is different from the Snellen chart that most North American countries use. This kind of eye chart follows the National Standard of the People's Republic of China (No. GB 11533-2011) and is proved to provide same testing accuracy as Snellen chart (Chaplin & Bradford, 2011). The testing distance could be changed from 2 to 5 meters, and the sizes of optotypes will scale down according to distance changes. Therefore users could choose the comfortable distance according to the space.

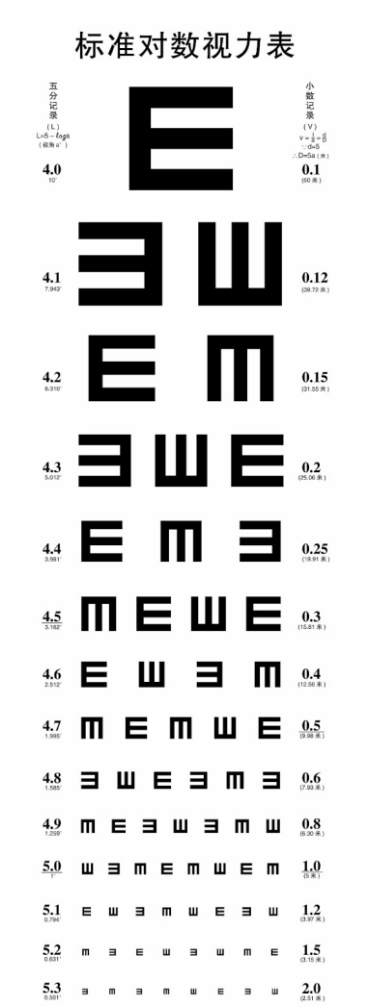

Fig 1.2 The E tumbling chart

1.2 Apper Context

As a graduate student of knowledge media design (KMD) with biomedical undergraduate background, I consider EyeXam bringing both positive influences. As a healthcare app, EyeXam creates a new way of eye testing and save people's time. It change the testing and recording method of traditional eye tests. With the help of EyeXam, users could complete eye exams by themselves and keep the testing record in the software database. In the future, the application may connect to the Internet and users could be able to share testing results with their family doctors. I believe EyeXam could contribute more to the healthcare field and could become a great helper to people's lives.

On the other hand, EyeXam explores more ways of interaction between people and mobile devices by connecting sensor with iPad. As a KMD graduate, I'm excited to see that user could communicate with EyeXam with their movements and the iPad could give real-time responses to user gestures. All in all, the application have improved eye testing procedure, as well as brought new possibilities for interactive design.

2. Overall Design

The block diagram of our overall design is depicted in Fig 2.1.

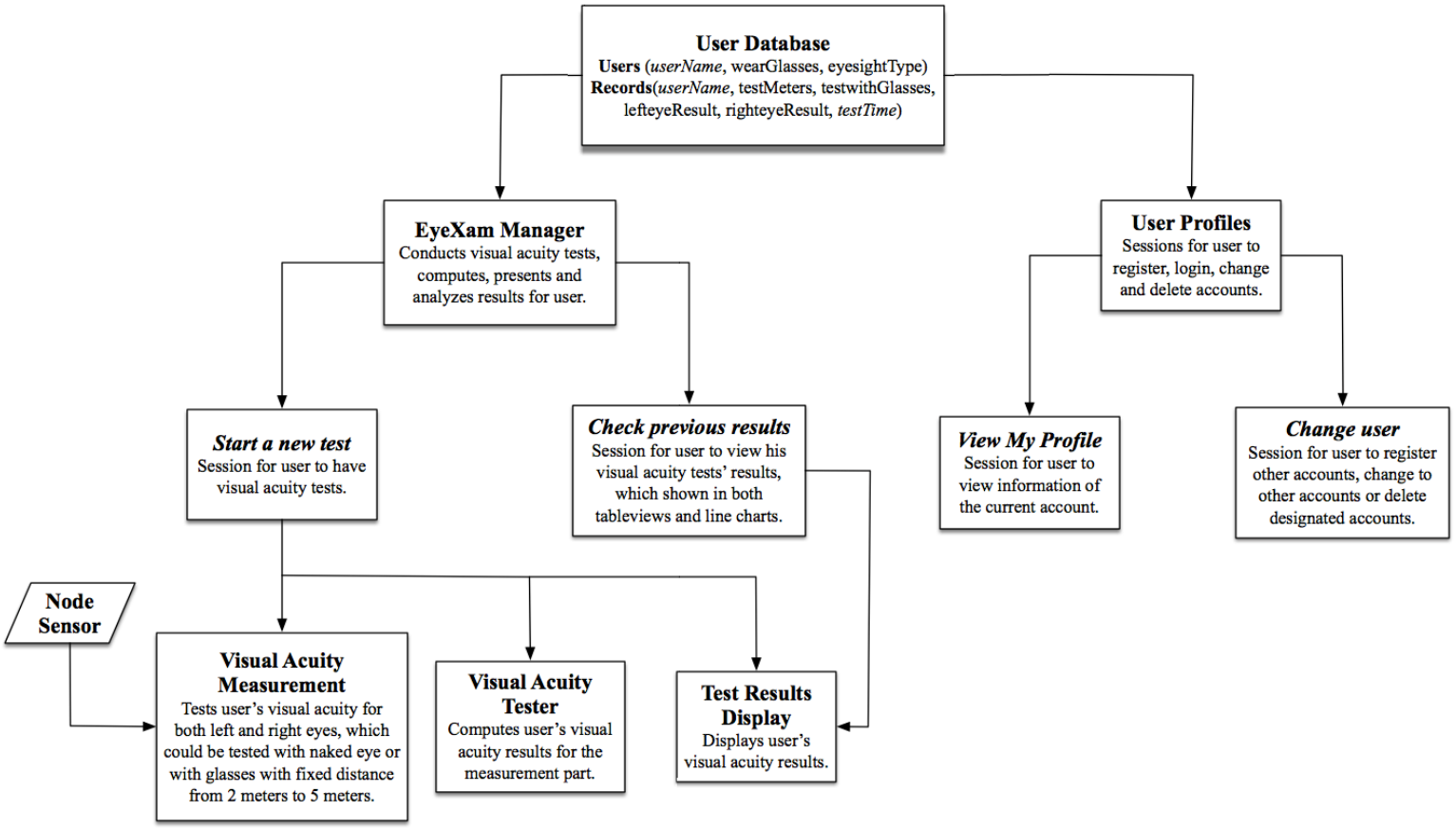

Fig 2.1 Block diagram for overall architecture

In EyeXam, there are two parts of stored information in database for users corresponding to two main functional parts of EyeXam which are *EyeXam Manager* and *User Profiles* respectively.

EyeXam Manager is responsible to conduct a new eye test and manage test's results, which consists of two functions *Start a new test* and *Check previous results*. With these two, users could both have a complete test and check their own result histories. Particularly, Node sensor is used in *Visual Acuity Measurement* to detect user movements and identify their pointing directions.

User Profiles is built for users to manage their accounts. It includes two functions *View my profile* and *Change user*. *View my profile* is designed for viewing user profiles and *Change user* allows multi-user accessing, and users could add or delete accounts via this function.

Details concerned with each block are shown in the block diagram.

3. Screen Shots and Functionality

To illustrate the functionality of our application, we would like to create a use case in which user Xinyi is going to complete an entire eye test with EyeXam. Suppose she has created and logged into her account, she would be redirected to her homepage (shown in Fig 3.1).

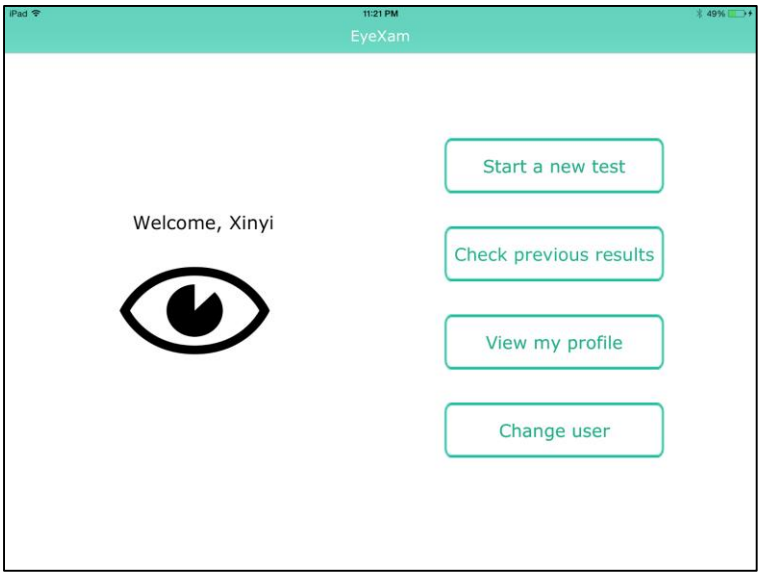

Fig 3.1 Homepage

3.1 Start a new test

3.1.1 Settings for a test

Before a new test, Xinyi needs to choose the settings including eye conditions and preferred distance for it (shown in Fig 3.2). She can set testing distance randomly from 2 to 5 meters.

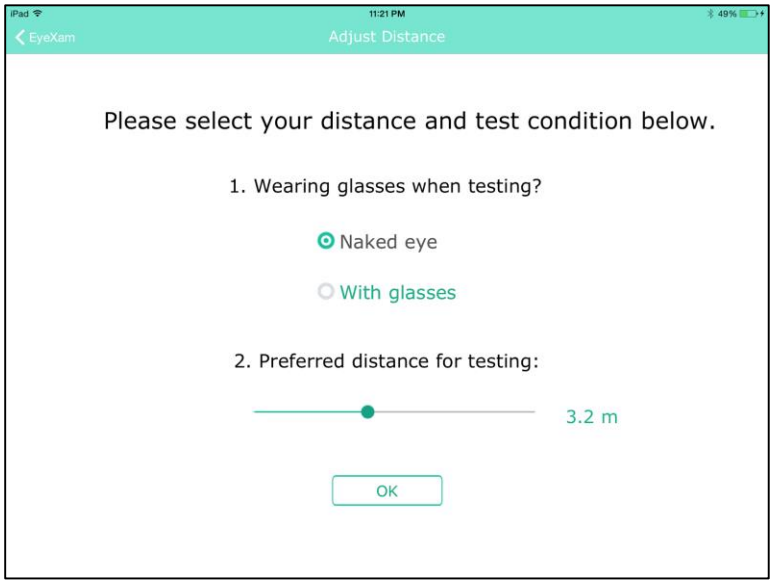

Fig 3.2 Before test

3.1.2 Calibration Phase

Now Xinyi enters the calibration phase and the first thing she needs to do is to connect with Node device through Bluetooth (shown in Fig 3.3).

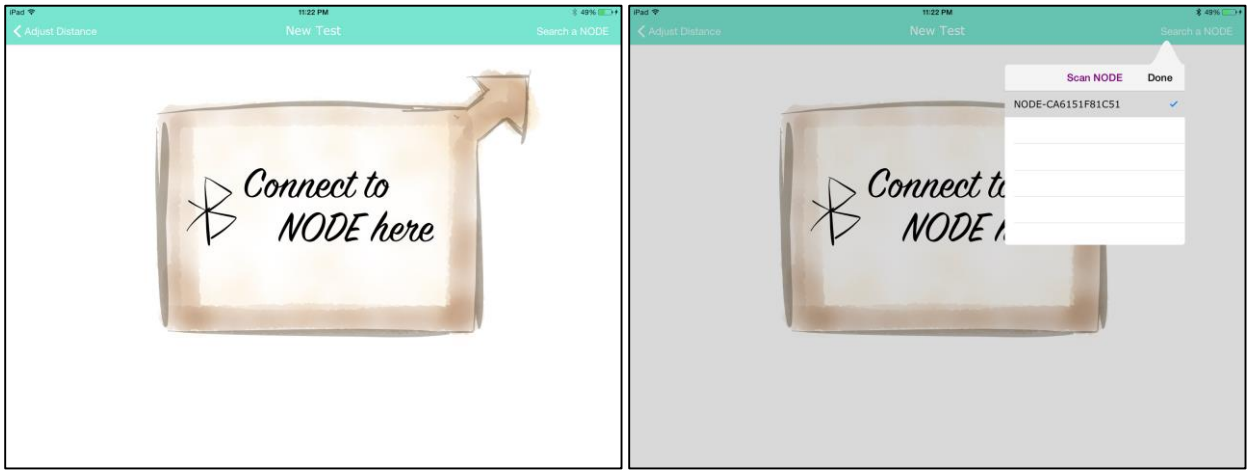

Fig 3.3 Connect Node sensors

Secondly, she is supposed to follow the audio instructions to conduct calibration for the four actions of holding Node device which represent four directions of the optotype showing in the later real test phrase (shown in Fig 3.4).

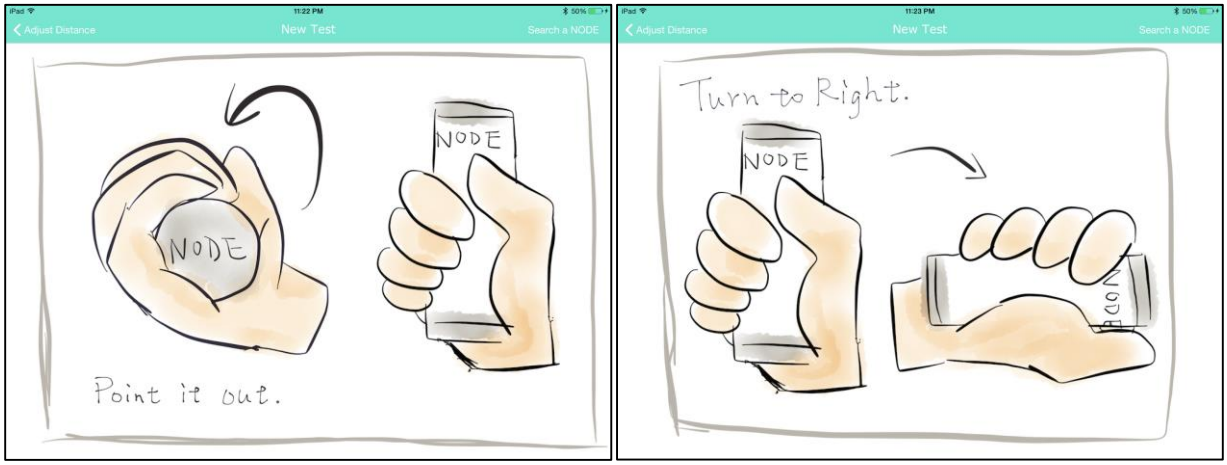

Fig 3.4 Instructions for calibration

After that, Xinyi enters the real test phrase. The size of optotypes will be in proportion to the preferred distance. In the two images below, the sizes of two optotypes correspond to a 2.8-meter and 4.2-meter preferred distance respectively. After pressing the button again which said by the audio instruction, Xinyi will start her real test now.

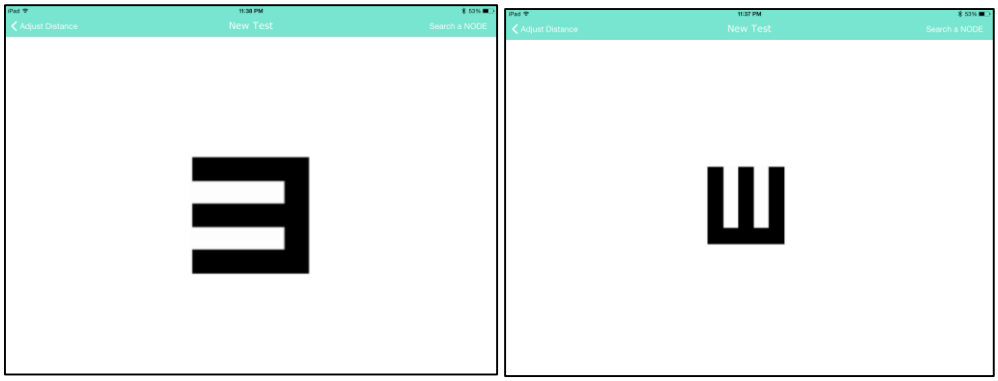

Fig 3.5 Optotypes

The optotypes are shown one after another on the screen, and the sizes of optotype changes according to Xinyi's performance (shown in Fig 3.5). During the test, according to the match between Xinyi's gesture and the direction of current optotype, EyeXam will give a green circle mark as a hint for correct answers, and a red wrong mark for wrong answers shown in the following pictures (shown in Fig 3.6).

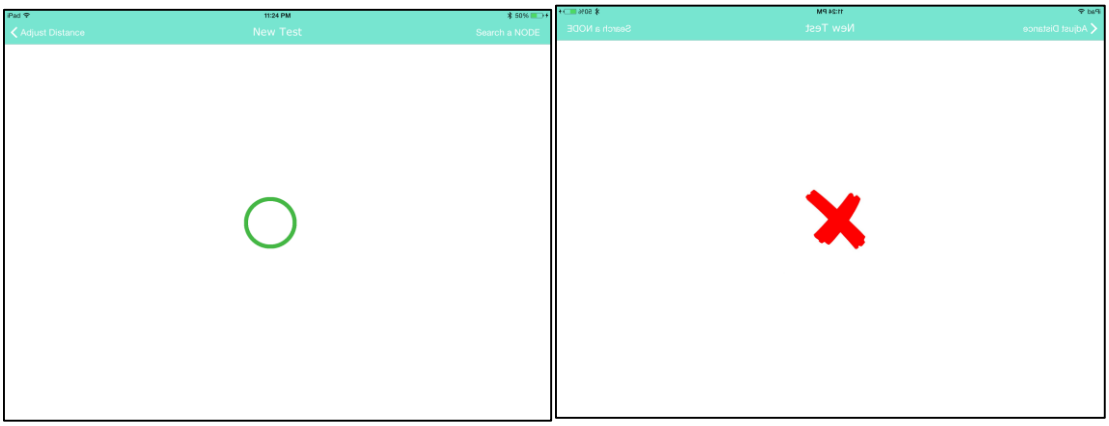

Fig 3.6 Respond to user gestures

If the user point to correct directions, the next optotype will be smaller. If the user makes two mistakes continuously, the optotype will change back to the former size. The test ends when user could tell three directions correct continuously. By default, Xinyi is supposed to get her right eye tested at first and then change to her left eye. After finishing the test for her right eye, there will be an alert window as well as a piece of audio to ask her to change to left eye by pressing button again (shown in Fig 3.7).

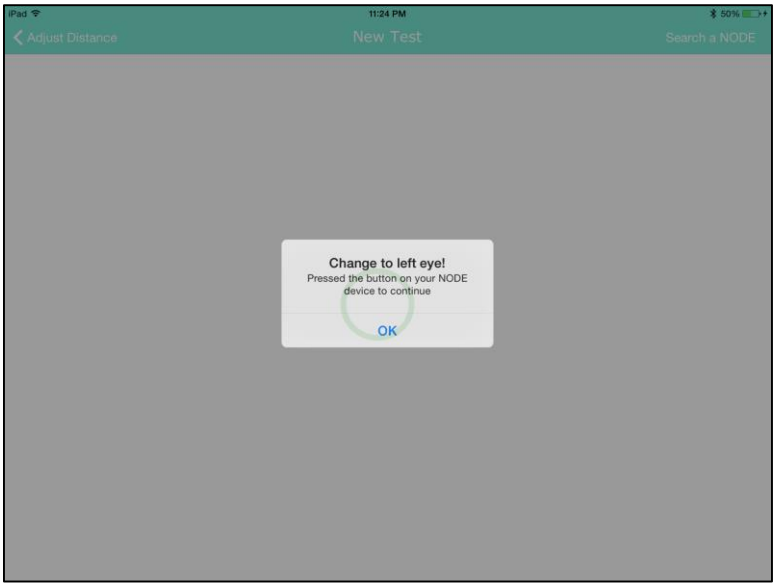

Fig 3.7 Change to the left eye

Finally, when she finishes tests of both eyes, an alert window will show the visual acuity results for this test. And she can choose save or discard the results (shown in Fig 3.8). The result notation for EyeXam is 20/20 vision notation, in which 20/20 indicates the normal eyesight.

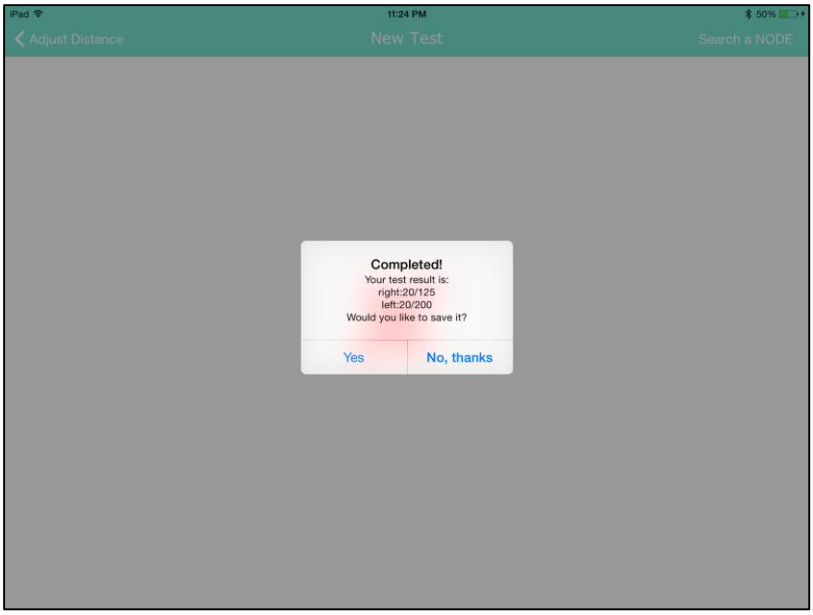

Fig 3.8 Result

3.2 Check previous results

3.2.1 Results displayed in table

Xinyi can check all of her test results in the past and switch between naked-eye and with glasses results by clicking on that toggle button. She could also delete a record by swiping to the left (shown in Fig 3.9).

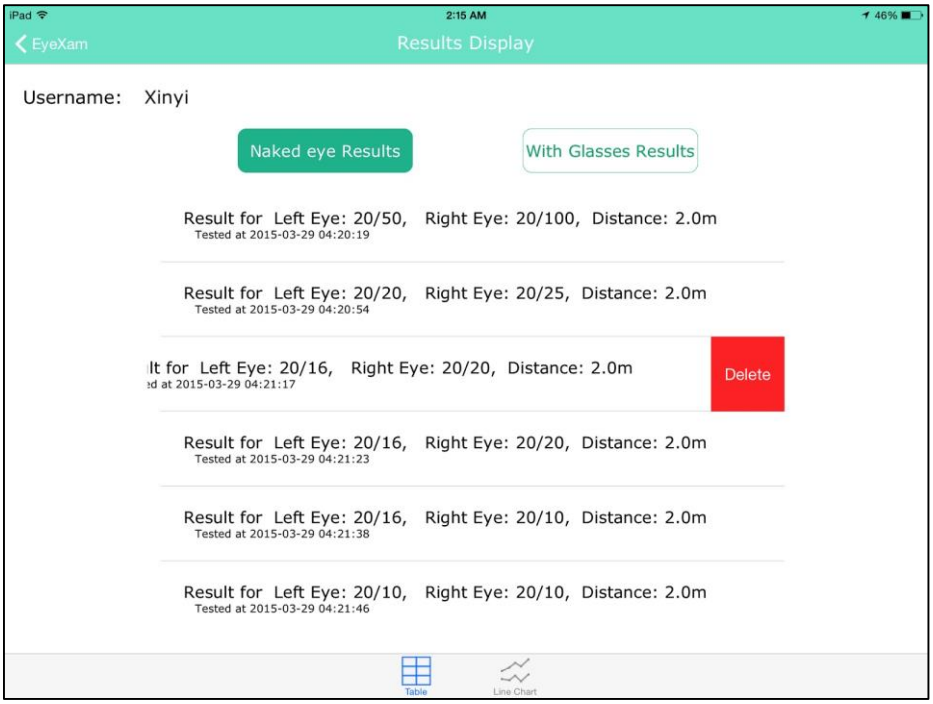

Fig 3.9 History results

3.2.2 Results displayed in line chart

In order to achieve a better visual effect on showing the trend of user's visual acuity, results will also be displayed in the form of line chart showing below. Xinyi could get more information of a specific line by tapping on a line(shown in Fig 3.10). Those four lines respectively represent as follows:

Red line: left eye in With-Glasses test;

Red dashed line: right eye in With Glasses test;

Green line: left eye in Naked-eye test;

Green dashed line: right eye in Naked-eye test.

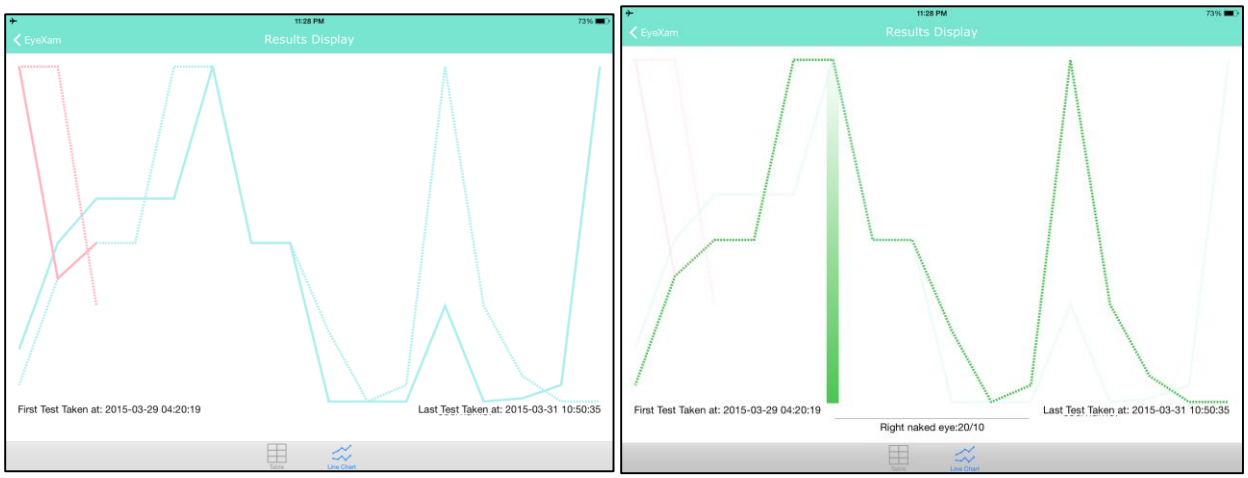

Fig 3.10 Line chart for results

3.3 View My Profile

Xinyi can view the information she filled in when she signed up (shown in Fig 3.11).

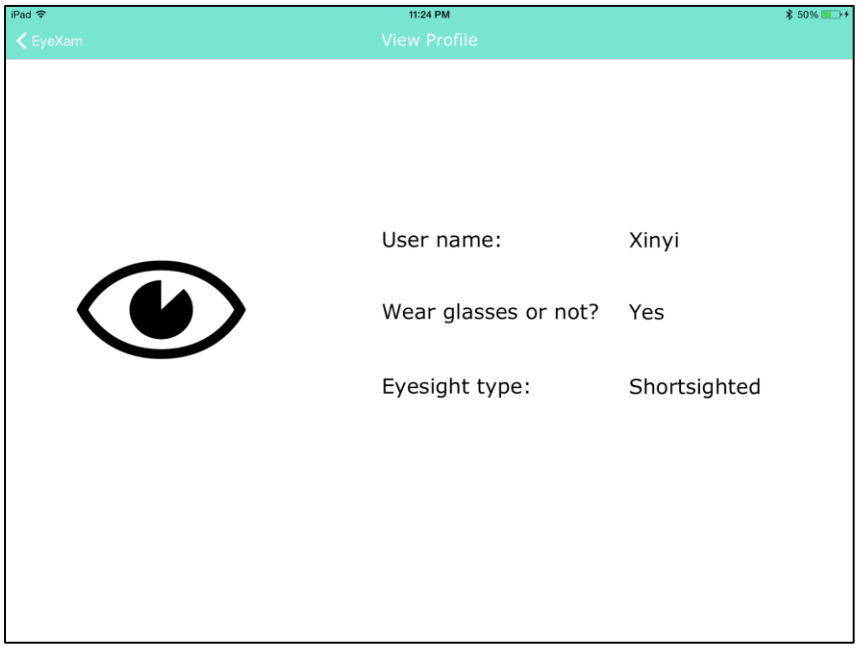

Fig 3.11 View user profile

3.4 Change User

3.4.1 All Accounts

In this screen, Xinyi could view all the accounts on her device (shown in Fig 3.12).

Fig 3.12 All accounts

3.4.2 Change Account

Xinyi could change to another accounts by pressing on the table cell of that account (shown in Fig 3.13).

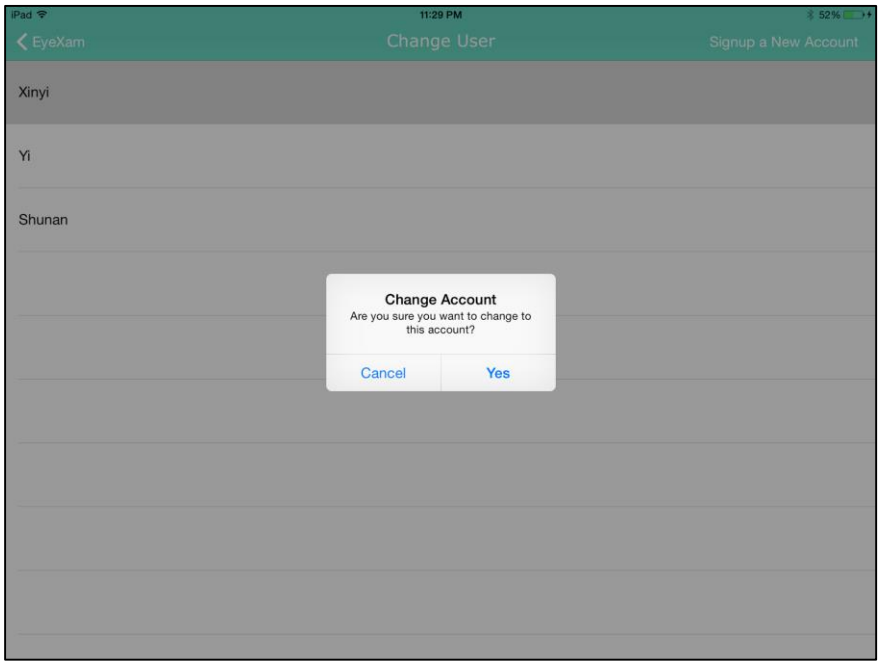

Fig 3.13 Switch between accounts

3.4.3 Delete Account

As a root user, who was the first account signed up in this device, Xinyi could delete any other accounts by swiping left and hit delete (shown in Fig 3.14).

Fig 3.14 Delete an account

3.4.4 Sign up a new Account

Xinyi can sign up an new account by clicking *Signup a new account* on the top right corner. After signing up, it will be redirected to the homepage of this new account "Yi".

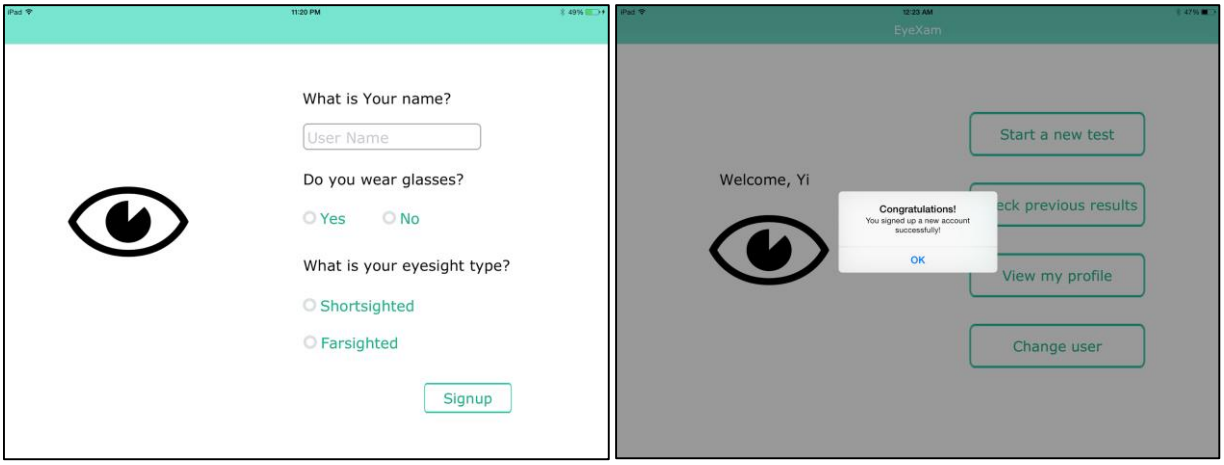

Fig 3.15 Sign up a new account

4. What did you learn - what would you do differently?

4.1 Apper

If I could experience the app development again, I would pay more attention on the basic principles of eye test and gather more research materials to support the testing method used by EyeXam. As a healthcare software, EyeXam need to be theoretically accurate to operate correctly, therefore I need to prove that the testing procedure, result notation and eye chart we use is valid and the testing result of our application could be used to indicate visual acuity. At the beginning, I didn't prepare enough supporting material for the application and the development meets difficulties all the time. Then I did research to the main principles of eye testing and visual acuity to make the application valid. It turns out that EyeXam could present correct results, but I hope to get more prepared next time so that I could spent more time on adjusting UI and navigation.

4.2 Programmers

After all the work we programmers have done, the most significant thing we might want to do differently if we got a second chance is to choose some other device, like iPhone, instead of the Node device. Except the apparent reason that iPhone is more commonly used, honestly, we have been struggling with handling the raw 9-axes values of the Node.

At the beginning, it looked pretty easy for us to calculate those data in order to complete the gesture detection which, however, eventually turned out to be the most crucial part of our app. For Spiral 2, we implemented the gesture detection by employing 3-axes values of the accelerometer. Some thresholds are set to judge the user's actions to four directions. Once it exceeds a certain threshold and meet the conditions of one direction, a user's action would be determined as a valid gesture to

that direction. Nevertheless, this kind of method will set strict demands on a user's actions and then increase the difficulties for users to really use it. After that, in Spiral 4, considering the data we acquire from accelerometer is actually a data stream, we have tried to store a list of data in an added calibration phrase, which allowed a user to calibrate her actions, and then compared it with another list of data in the real test phrase. It should be the most accurate way to get gesture detection done, but because of the operating efficiency, we gave up this method and finally in the final version, we found a compromising approach employing magnetometer rather than accelerometer. In this approach, the calibration phrase is simplified as storing one set of 3-axes values of magnetometer when a user presses the button. It is a tradeoff between the accuracy and efficiency. Then in the final presentation, we found out that the 3-axes values of magnetometer are magnetic field-sensitive which means those values would fluctuate in a huge range when the magnetic field of the testing environment is too complicated, for instance, there are lots of electrical appliance. We have to admit we never thought about this affecting element which directly leads to the calibration failure during the presentation. Although the consequence might be unsatisfying, we do have learned a valuable lesson from this experience.

5. Contribution by Group Members

Shunan Chen (Apper) is responsible for generating the app functions and designing the navigation and user interface of the application, as well as gathering principles for eye testing to keep the testing procedure valid.

Yi Li (Programmer) is mainly responsible for building the app structure, implementing the gesture detection method and refining the overall testing flow.

Xinyi Lin (Programmer) is responsible for implementing the database interfaces, providing the adjusting the size of optotype according to the preferred distance algorithm and beautifying the user interfaces.

6. Future Work

To better improve our app, several modifications would be made for our app:

1. Add settings menu for users to turn off the audio guide.

To help users learn how to start a new test with Node device, we added some audio instructions in the test session. This would be helpful for users' first few attempts on EyeXam. However, once a user is familiar with the testing steps, those audio instructions would be a waste of time. Thus we need to add settings to turn off them.

2. Storing data into remote server.

In the current version, we store all the data in local device. However, if we store them into remote server, users could login their accounts and view result histories in different devices, which would be more convenient.

3. Adding function to measure distance.

EyeXam allows a user to have eye tests with the preferred test distance, which is the distance between her iPad and her ranging from 2 meters to 5 meters measured by herself. Nevertheless, if we could add a functionality to measure this distance in app, it would be more user-friendly.