IM3080 Design and Innovation Project

(AY2020/21 Semester 1)

Project Report

Title: FashionAR

Github: https://github.com/Vanessa-24/DIPGrp7

Submitted by: Group 7

Supervisor: Erry Gunawan

Table of Contents

1. Background and Motivation

Augmented Reality (AR) refers to any technology that superimposes digital information onto the user's physical environment in real-time [1]. By overlaying the existing environment with rendered data or images, it elevates the user's visual perception of their surroundings.

In recent years, there has been an increase of AR applications. With AR technology, it improves the buyer's experience, by enabling them to envisage and experience a product or service before actually buying it [2].

The proliferation of online shopping has made AR technology more applicable and relevant to the customer shopping experience. There is an expansive variety of goods and services offered on the Internet. However, with an endless amount of options, it is difficult for the buyer to decide, especially when they are unable to try a product. Thus, through the employment of AR technology and machine learning into our application, our group wants to enhance the buying experience for customers. Focusing on a single product—glasses—our mobile application allows users to freely try on any and every style of glasses across multiple shops online. With our recommendations feature, users are able to receive a personalised curated list based on their face shape and previous ratings.

2. Objective

When it comes to choosing the perfect pair of glasses, the options offered in the market are aplenty. From shape, to size, colour and style of the frames, there are many factors to consider. With the bombardment of different choices, it can be confusing for the average consumer. Thus, this is what our application seeks to address.

FashionAR aims to make the online shopping experience accessible and convenient for the average buyer. Through the use of AR technology and machine learning, our mobile application targets fashion eyewear enthusiasts, providing them an authentic and realistic experience to experiment new forms of styles whenever and wherever with no expenses needed.

Based on the user's ratings and face shape, FashionAR's smart recommendations algorithm suggests the best products to suit the user's taste and preferences. It also presents fresh designs, allowing users to discover alternative options beyond their normal.

Given these unprecedented times, the shift to virtual experiences has made the purchasing of products hygienic, convenient, and responsible.

3. Review of literature/technology

3.1. Android Studio: Main software used for creating our mobile application.

3.2. Google Sceneform: This is used to produce the AR portion of our application. Sceneform SDK makes it straightforward to render realistic three-dimensional (3D) scenes in AR and non-AR apps, without having to learn OpenGL. It includes a high-level scene graph API, a realistic physically based renderer provided by Filament and an Android Studio plugin for importing, viewing and building 3D assets [3].

3.3. Amazon Web Services (AWS) : AWS (EC2) is used to host our web server, written in Python, to do machine learning and face detection algorithms. Thus, relieving our local devices of additional space, enabling phones with limited central processing units (CPU) to render and process quickly.

3.4. Firebase: For authentication of Google accounts log-ins and a database for collection of users' information (ratings and face shape).

3.5. Jupyter Notebook (Python): We used Python programming language and some data science libraries, such as pandas, numpy, sklearn, etc. to write our machine learning algorithm for our face detection and recommendations system.

3.6. Autodesk Maya: Maya, a 3D modelling software is used to reposition and resize the glasses models to fit our canonical face mesh. Additionally, we used the software to add textures and colours to our models, producing more realistic and appealing designs.

3.7. GitHub: We utilised GitHub as a collaborative tool to help us manage our work and code efficiently. We simply had to push our codes to Github whenever we finished our own parts, then anyone in the group could pull the latest version to test out the application. We also created different branches out of the master for our respective subgroups, so we could work effectively on different aspects of our application.

4. Design and implementation

4.1. Design Considerations

i. **Usability:** The app is designed to be user friendly with a straightforward and minimal interface, allowing users to navigate easily within the application without much interference.

ii. **Compatibility:** The use of AR software in the application may prevent older Android models or devices with no ARCore from downloading it. Also, phones with older hardware may likely experience delays in application's performance.

iii. **Consistency:** The overall design of FashionAR is kept visually consistent with a logical structure and clearly defined icons, similar to most camera

applications, which gives users a sense of familiarity and control as they navigate and use our application.

iv. **Performance:** We made use of asynchronous programming to create a more responsive interface for users. Furthermore, by hosting our machine learning algorithms on the cloud, it saves space and reduces processing time.

v. **Navigation:** While designing our interface, we took into consideration the different number of ways and navigation requirements users may have as they interact with our application. Hence, we developed an interface that is intuitive and fulfilled the desired goals of users with certain profiles.

4.2. Final Design (with block diagrams)

Overview

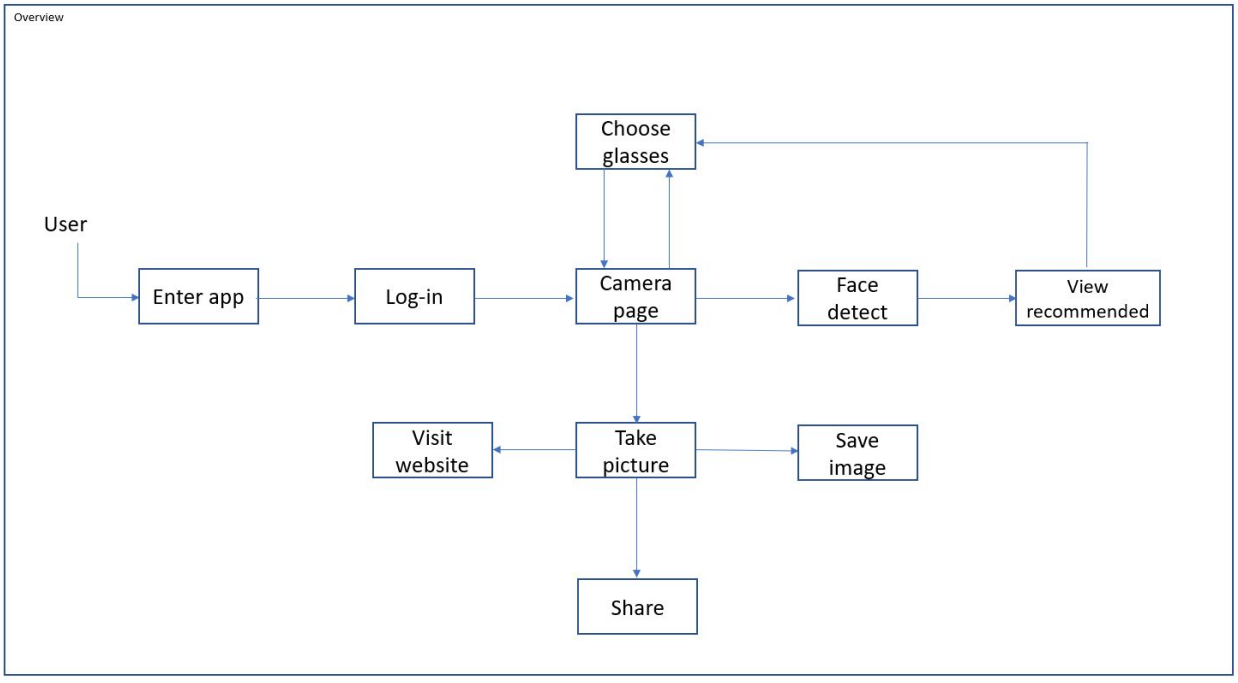


Figure 1: Overview Block Diagram

API, Libraries, Software/Services Used

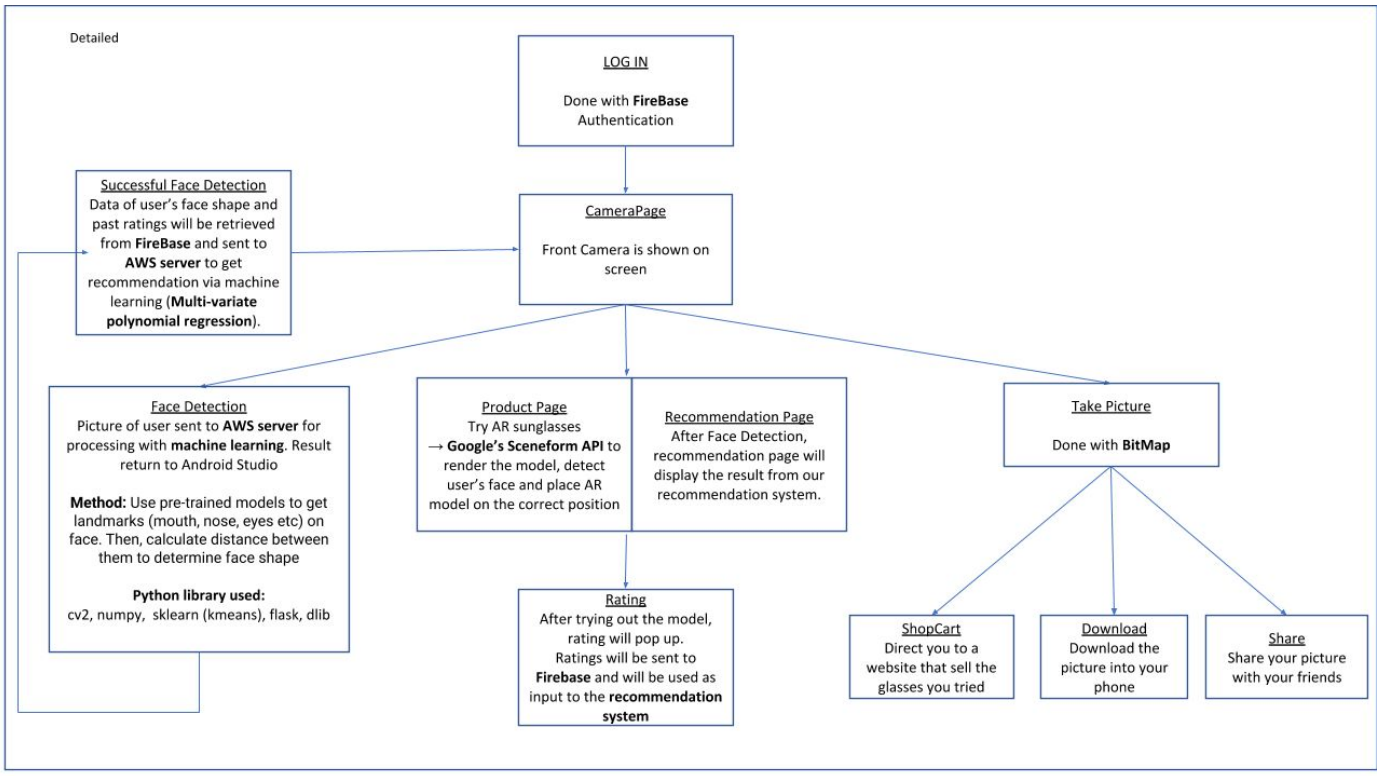


Figure 2: Detailed Block Diagram

4.3. Implementation (with photos)

Log-in

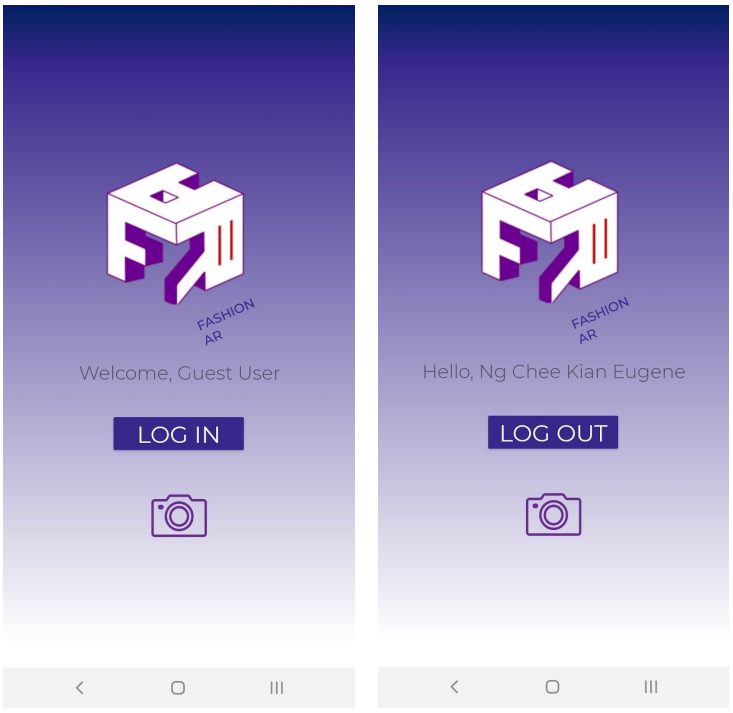


Figure 3: Login Page

To access the full features of our app, the user has to log-in with their gmail account. Their email account information, together with the user's face shape and ratings will be stored on the Firebase server. We implemented firebase into our application, by adding the SDK to the build gradle, and importing the dependencies for Firebase's database and authentication. If the user did not log-in, they would be prompted to log-in in order to proceed.

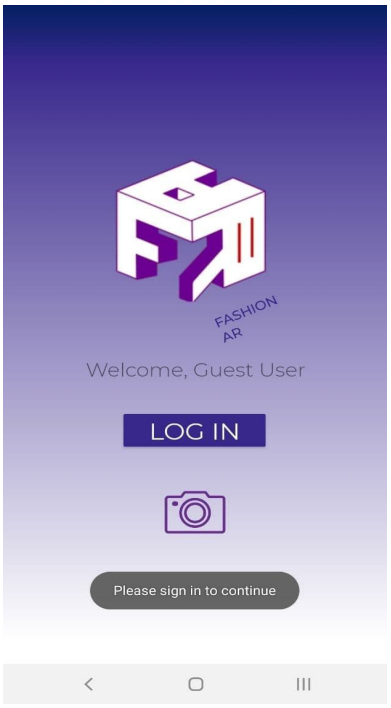


Figure 4: User is prompted to sign in

The function to log in or out is located inside ***MainActivity.java***. Its name is ***GGSignin()/GGSignOut()***.

Bottom Sheet and Tab Layout

Figure 5: Products and Recommendations Fragment respectively

In order to develop the designs of our products and recommendations catalogue, we needed to implement the material design library dependency in our build.gradle [4]. We used two material components: bottom sheets and tabs.

When a user clicks the products icon, the glasses catalogue will slide up from the bottom of the screen. This is the bottom sheet which acts as the main frame to display the models. We chose to use a persistent bottom sheet, as it allows the user to interact with both the main interface and sheet itself while it stays on-screen [5].

To switch between the products and recommendations catalogue, we used the material component—tabs—to create two fragments to house the respective models [6]. With each fragment, there is a corresponding class and XML layout, i.e. ***ProductsFragment*** class with ***fragment_products.xml*** and ***RecommendationsFragment*** class with ***fragment_recommendations.xml***

We also needed to create a separate class, ***ViewPagerAdapter.Java*** to be able to toggle between tabs easily with a swipe of a finger.

The bottom sheet and tabs logic is implemented in ***CameraPage.java***.

AR Models

We sourced our glasses from 3D assets websites such as Sketchfab. Using a 3D modelling software such as Autodesk Maya, we resized our models to fit our canonical face mesh. Additionally, modifying our models by adding texture, basing them off actual products available on the market. This allows us to produce a sufficient variety of models for our catalogue. In order to import the models to Android Studio, we converted our 3D objects to the appropriate formats i.e. .fbx

and .fba. After an extensive testing and tweaking process, we ensured that all forty of our models were rendered correctly and placed properly on the face.

To implement AR technology into our application, we utilised Sceneform API which plots several landmarks on a user's face to accurately identify the location and different features of their face. The rendered model will be then overlaid on the face. This is done with real time tracking to ensure that AR models would continuously be placed properly on the user's face. We have included pictures of our glasses in the products tab, so users can tap on any pair to try it on. This is implemented programmatically by the model renderable function, ***loadMdl()***, located inside class ***CameraPage.java*.**

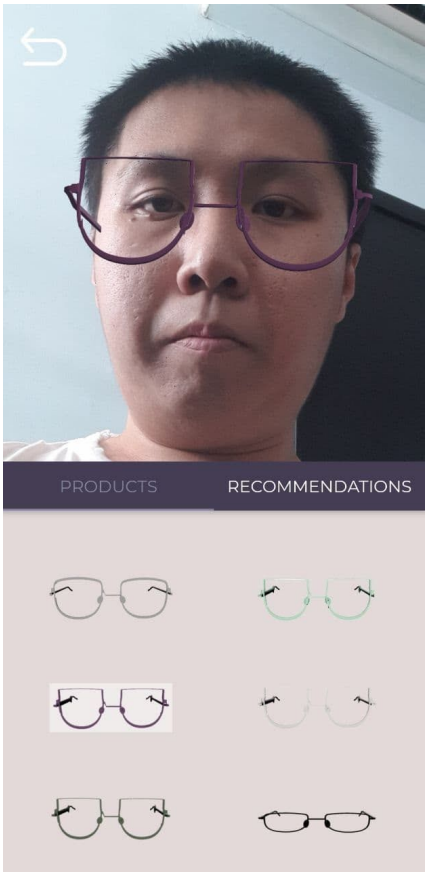


Figure 6 : User trying out the AR model

To improve the user experience (refer to Figure 5), we have added a highlight box that appears when a pair of glasses is selected.

Face Shape Detection

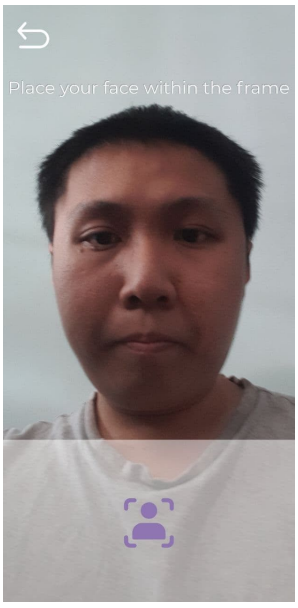


Figure 7: Face Detection Page

An image of the user will be taken and sent to our server on AWS which hosts the code to process the image and deduce the user's face shape. The facial detection process is done by using two pretrained models. The first model (haarcascade_frontalface_default) is for face detection. The second model (shape_predictor_68_face_landmarks) is to obtain facial landmarks i.e. mouth, nose, eyes etc. The classification of face shape is derived by calculating the distance between the different landmarks.

The function in Android Studio is ***faceShapeDetect()*** in ***ScanPage.java***.

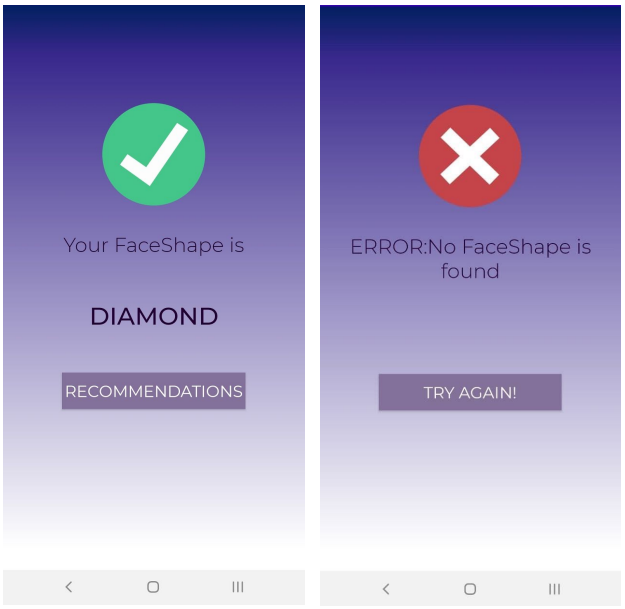


Figure 8: Face Shape Detection Results

Rating

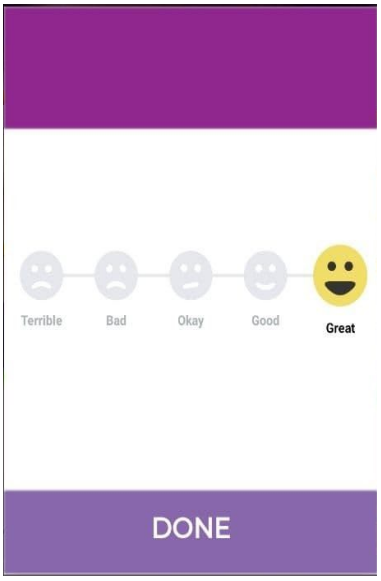


Figure 9: Rating page

After trying a new sunglass model, a rating window will pop up to prod users for their feedback. To verify whether a user has rated a model, we can retrieve data from Firebase to check if the rating is null or not. Whenever a user has given their

rating, the information will be sent to Firebase. This will be used as data to train our machine learning model.

The rating window uses animations and is done via reference to dialog function and Smiley Rating dependency [7].

The function used is ***helperRateModel()*** in ***CameraPage.java***

Recommendation System

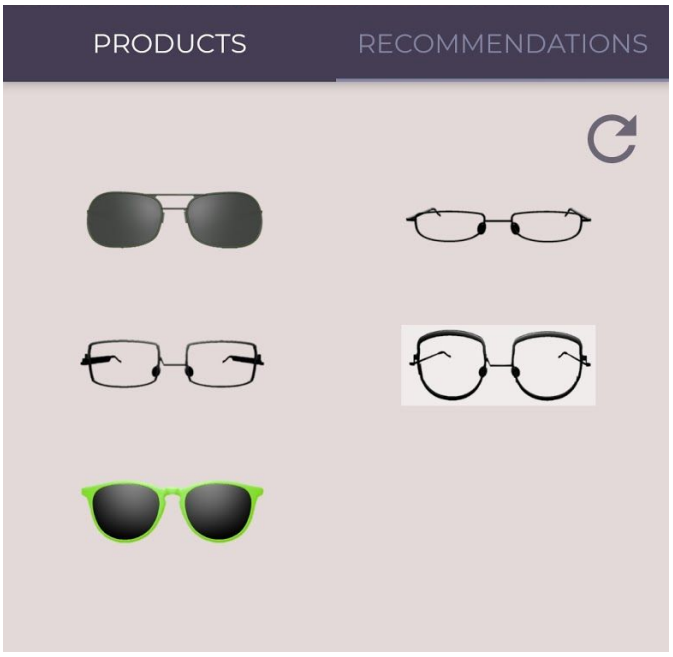


Figure 10: Recommendation Page

The recommendation system makes use of machine learning to suggest sunglasses for users. It is a content-based recommendation system [8] that is hosted on AWS server.

After testing different machine learning models, we have concluded the multivariate polynomial regression to be the most accurate, as it has the least mean squared error. The model uses specific attributes of the glasses such as the shape and colour similarities to input into the model and generate ratings as the

output/response. The model aims to predict the ratings of glasses that have not yet been rated by the user. The percentage of suitability of a model to a certain face shape is obtained by referencing several fashion experts' advice— essentially, Domain Experts' knowledge is used in determining the percentage. Rather than just performing a one-hot encoding on the colours of the sunglasses, we decided to calculate the colour similarities [9] of each pair with the RGB colour model. By doing so, the regression model will be able to recommend sunglasses with similar colours to those liked by the user. As compared to recommending sunglasses with the same colour if one-hot encoding was applied.

	sim_R	sim_G	sim_B	Round
0	1.00000	0.15024	0.25478	0.5
1	0.25478	0.11815	1.00000	0.2
2	0.47326	0.33319	0.42291	0.3
3	1.00000	0.15024	0.25478	0.3
4	0.62753	0.23543	0.62753	0.3
5	0.47326	0.33319	0.42291	0.3

Figure 11 : Table of Inputs to ML

With reference to Figure 11, the table shows the inputs/features to the model. The index represents the sunglass' id number (e.g. 0 → Glasses 1). sim_R, sim_G, and sim_B represent the similarity of the glasses' colour to the individual RGB colours respectively. The values represent the percentage of suitability of glasses to face shape. In this specific example, round refers to the type of face shape a particular user has.

This regression model is trained through supervised learning. After predicting the ratings of glasses, the predicted ratings will be sorted and returned to Android Studio.

The function in Android Studio is called ***getRec()*** which sends requests along with user's data to server hosted on AWS in ***RecommendationPage.java***

Refresh

In reference to Figure 10, when the refresh icon is clicked (top right hand corner), the application will gather the latest data—face shape and ratings of the user—from Firebase and send it to the AWS server to receive new recommendation data.

The function in Android Studio is called ***refreshReco()*** in ***CameraPage.java***

Take Picture

Figure 12: Camera page

This function enables the user to take a picture with the glasses model on via BitMap function and the ARSceneView as the picture view.

The function is called ***takePicture()*** in ***CameraPage.java***.

Preview Page

Figure 13: Preview Photo

Upon taking a photo, the user will be led to the preview page which displays three icons: (i) Shopcart, (ii) Download, and (iii) Share.

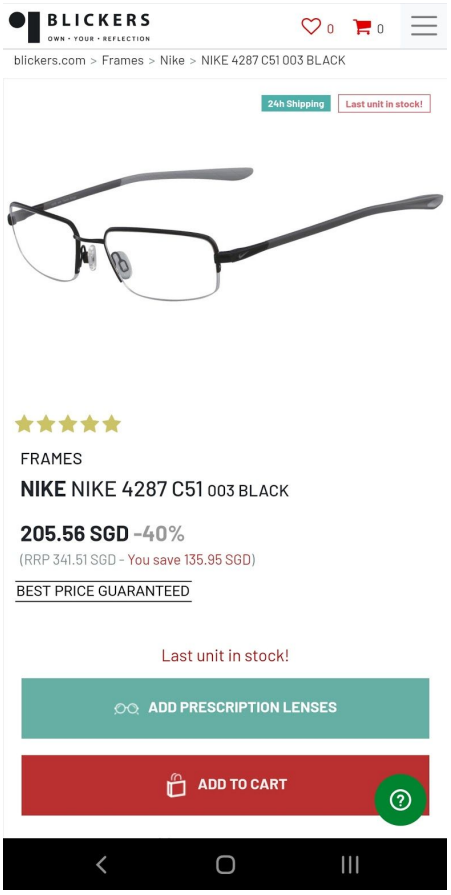


Figure 14: External shopping website

i. Shopcart

After clicking on the shopcart icon, the user will be directed to the website containing the glasses of interest. Each product is linked to their respective webpage URL, making it convenient for users to buy their products without exiting the app. This provides a seamless purchasing experience. The logic behind this is implemented via a string array (links[]) which contains the links to the respective websites, with the glasses number representing the array index.

The function is called ***shopButtonClicked()*** in ***PreviewPhoto.java***

Figure 16 : Download page

ii. Download

Users can download the picture they took to their phone by saving the BitMap to disk. When the picture is saved, a toast message will appear, indicating a success.

This function is called ***download()*** in ***PreviewPhoto.java***

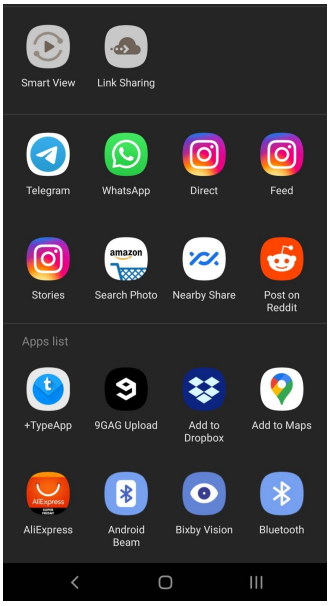


Figure 15: Share page

iii. Share

The share function allows the picture taken to be sent to other applications. It is done via Android Sharesheet [10]. The image is appended to the intent and the user can select which application to send the image to.

The function is called ***shareButtonClicked()*** in ***PreviewPhoto.java***

4.4 Discussion

Advantages

- The application takes into account the user's physical traits, past ratings and recommends suitable eyewear for the user, this allows the recommendation to be tailored made for every user.
- Built with AR technology, it presents an accurate perspective of the fitting experience. Users can try out various types of spectacles with just a single tap.
- The shopcart function directs users to the respective webpage upon deciding on a pair of glasses, thus improving user's convenience.

- Businesses are able to take advantage of our application to market their products. Additionally, providing their expertise and product information, so customers can be well-informed.
- It is hassle-free, convenient to use anytime, anywhere, and any place

Disadvantages

- The application is limited to only eyewear.
- We were restricted to the AWS free tier console, thus we were not able to host a better face detection machine learning algorithm, which requires more computing resources.
- Given more time and practice, our 3D models could be more accurate and realistic.

5. Conclusion and Recommendation

FashionAR is a mobile application that strives to improve and innovate the customer shopping experience online. By adopting technologies such as AR and machine learning, we hope to solve the woes and concerns of online shopping to achieve a trouble-free and, yet realistic shopping experience for consumers. Not only can users virtually experiment and try on the various pairs of glasses available, our machine learning algorithm also serves as a digital stylist by recommending the best pairs to suit the user's physical attributes and preferences. The inextricable blend between technologies and fashion in our application creates an enjoyable, convenient and fuss-free affair for the user.

As our group was limited by time and skills, we believe that we could build a more comprehensive mobile application that includes beyond glasses. We could expand our catalogue with other fashion accessories such as hats, earrings, coloured contact lenses and make-up. Aside

from accessories, it could be expanded to clothes, recommending appropriate articles of clothing to any body shape.

Another field that could be improved on is the recommendations machine learning algorithm. It could be changed to a hybrid-based recommendation system which would enhance the application's capability to suggest suitable models for the user.

Lastly, the face detection system can be improved to provide a more accurate and detailed facial recognition result. Outside of the standard face shapes, it can also take into account users' jawlines, cheekbones and the size of their forehead. This could prove a more precise recommendation of models for the users.

References

[1] S. Overby, “How to explain augmented reality in plain English,” The Enterprisers Project, 07-Oct-2019. [Online]. Available: https://enterprisersproject.com/article/2019/10/ar-augmented-reality-explained-plain-english. [Accessed: 05-Dec-2020].

[2] B. Marr, “9 Powerful Real-World Applications Of Augmented Reality (AR) Today,” *Forbes*, 30-Jul-2018. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2018/07/30/9-powerful-real-world-applications-of-augmented-reality-ar-today/?sh=1c420752fe95. [Accessed: 03-Dec-2020].

[3] “Sceneform (1.15.0) ,” *Google Developers*. [Online]. Available: https://developers.google.com/sceneform/develop. [Accessed: 05-Dec-2020].

[4] “Getting started with Material Components for Android,” *Material Design*. [Online]. Available: https://material.io/develop/android/docs/getting-started. [Accessed: 06-Dec-2020].

[5] “Sheets: bottom,” *Material Design*. [Online]. Available: https://material.io/components/sheets-bottom#usage. [Accessed: 06-Dec-2020].

[6] “Tabs,” *Material Design*. [Online]. Available: https://material.io/develop/android/components/tabs#scrollable-tabs. [Accessed: 06-Dec-2020].

[7] *YouTube*, 19-Feb-2018. [Online]. Available: https://www.youtube.com/watch?v=sKfXX673VjE&ab_channel=TechnoLoud. [Accessed: 06-Dec-2020].

[8] *YouTube*, 09-Feb-2017. [Online]. Available: https://www.youtube.com/watch?v=9siFuMMHNIA. [Accessed: 06-Dec-2020].

[9] T. Riemersma, “Colour metric,” *CompuPhase*. [Online]. Available: https://www.compuphase.com/cmetric.htm. [Accessed: 06-Dec-2020].

[10] “Sending simple data to other apps,” *Android Developers*. [Online]. Available: https://developer.android.com/training/sharing/send. [Accessed: 06-Dec-2020].

Bibliography

"Augmented Faces | Sceneform (1.15.0) | Google Developers," *Google*. [Online]. Available: https://developers.google.com/sceneform/develop/augmented-faces. [Accessed: 07-Dec-2020].

"Import and preview 3D assets | Sceneform (1.15.0) | Google Developers," *Google*. [Online]. Available: https://developers.google.com/sceneform/develop/import-assets. [Accessed: 07-Dec-2020].

"Installation & Setup on Android | Firebase Realtime Database," *Google*. [Online]. Available: https://firebase.google.com/docs/database/android/start. [Accessed: 07-Dec-2020].

Introduction to Sceneform. [Online]. Available: https://kiosk-dot-codelabs-site.appspot.com/codelabs/sceneform-intro/index.html?index=..%2F..devfest18. [Accessed: 07-Dec-2020].

rajendra7406-zz, "rajendra7406-zz/FaceShape," *GitHub*. [Online]. Available: https://github.com/rajendra7406-zz/FaceShape. [Accessed: 07-Dec-2020].

YouTube, 11-Jan-2018. [Online]. Available: https://www.youtube.com/watch?v=x_juFIQeTgI&t=19s. [Accessed: 07-Dec-2020].

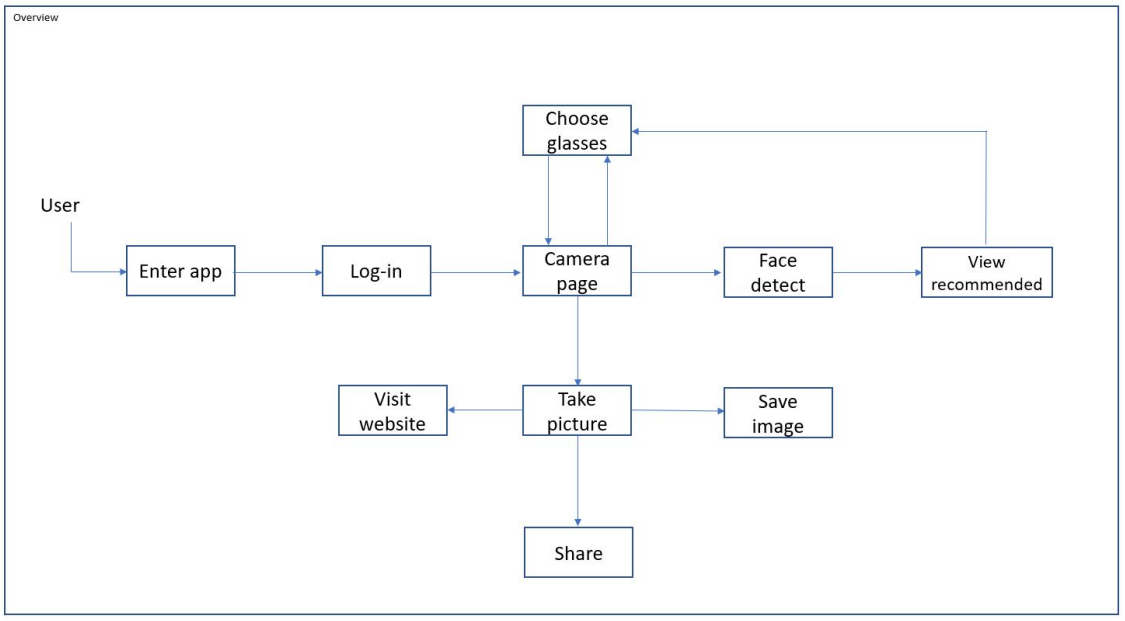
YouTube, 19-Feb-2018. [Online]. Available: https://www.youtube.com/watch?v=sKfXX673VjE&ab_channel=TechnoLoud. [Accessed: 07-Dec-2020].

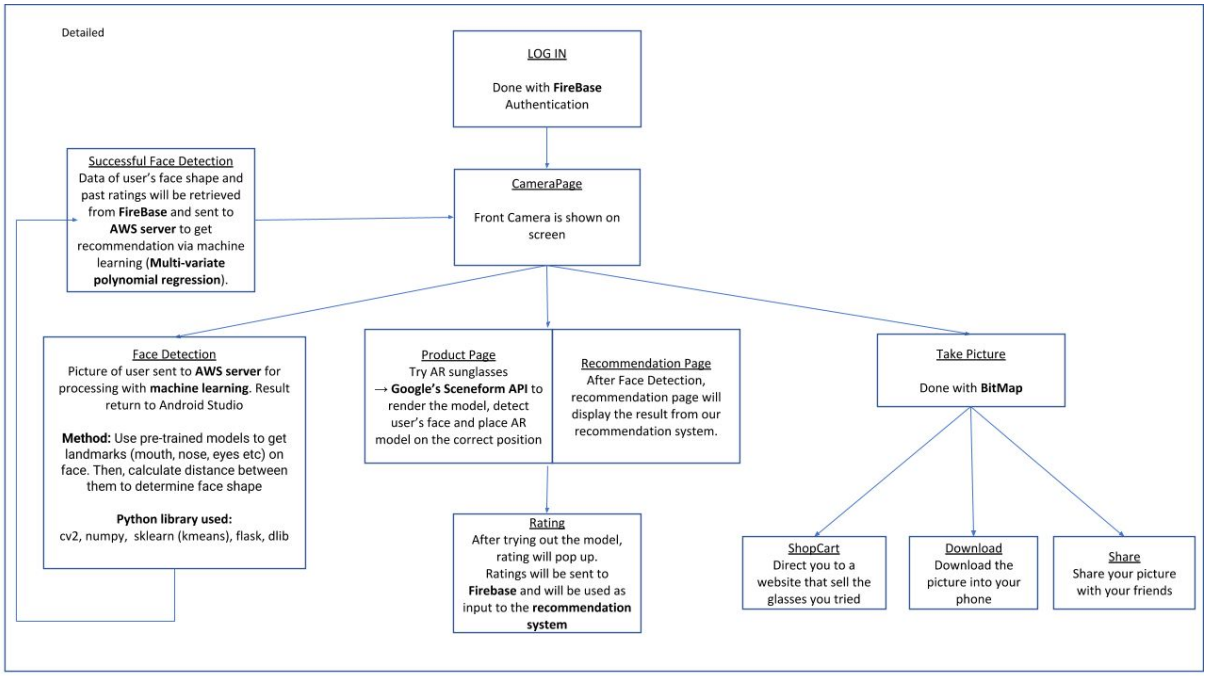
YouTube, 07-Sept-2019. [Online]. Available: https://www.youtube.com/watch?v=3Zu9Bd2Pwg4 [Accessed: 07-Dec-2020].

YouTube, 13-Jan-2020. [Online]. Available: https://www.youtube.com/watch?v=eXK4VMI9XLI&t=608s. [Accessed: 07-Dec-2020].

Appendices

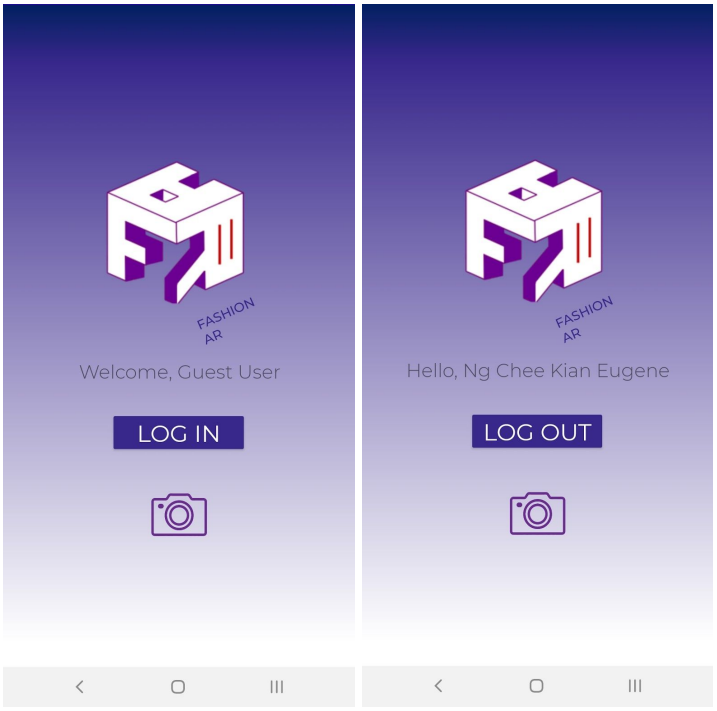
Design Diagrams



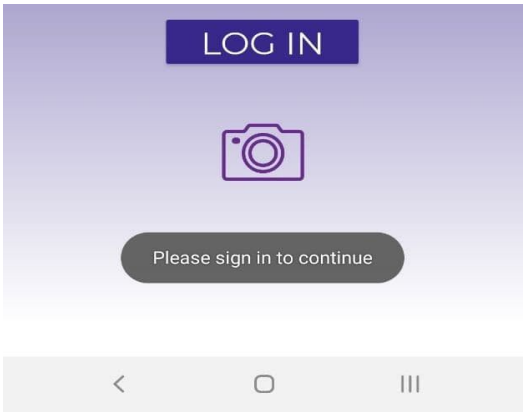


User Guide

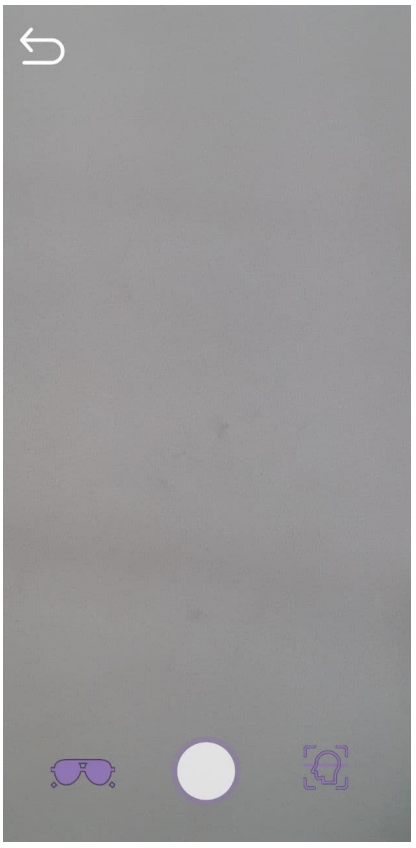
Log-in Page



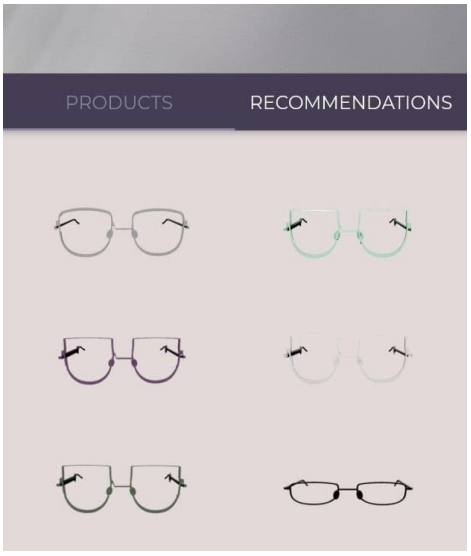
1. Tap the log-in button. You will be prompted to enter your Gmail details (Applicable to first time log-in only).
2. Enter your Gmail account and password.
3. To proceed to the main page/camera page, click on the camera icon. If you have yet to log in, you will be prompted to do so before accessing the camera function.

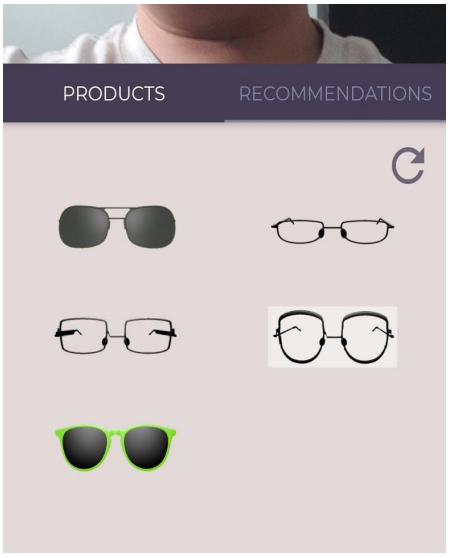


Main Camera Page



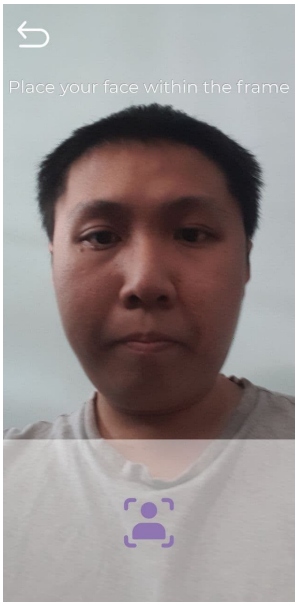
1. Tap the bottom right icon to access the face detection page.
2. Tap the bottom centre icon to take a photo.
3. Tap the bottom left icon to view the product catalogue.



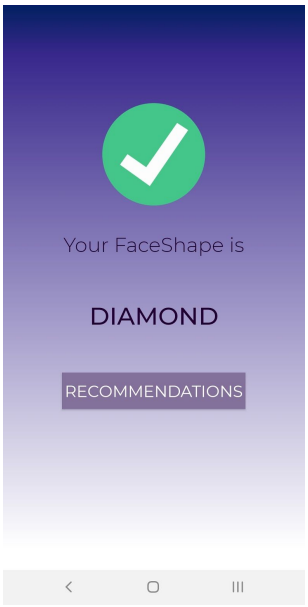


a. To view all available products, you can go to the products tab. Select the pair that you want and the model will appear on your face
b. If you did not use the face detection function, the recommendations tab will appear empty. You can view your recommended pairs of glasses after using the face detection function, or by refreshing your recommendations list if you have already obtained your facial shape results. The highlighted box represents the model that is selected.

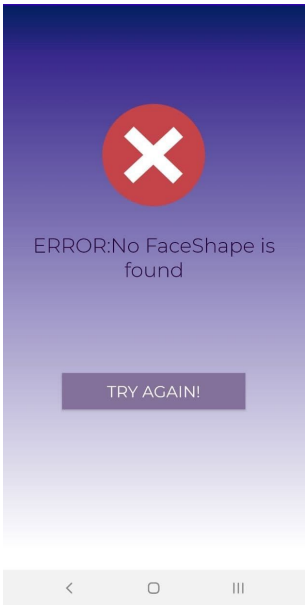
Face Detection Page



1. Press the icon in the bottom centre to scan your face.
2. After processing your face shape, your face shape results will be displayed.



3. Tap the recommendations button to go back to the camera page. The recommendations tab will be updated with suggested glasses.



4. If there is no face detected, an error message would appear, prompting you to try again.

Preview Page

1. Tap the bottom right ‘share’ icon to share the captured image.
2. Tap the bottom centre ‘download’ icon to save your image to your phone.
3. Tap the bottom left ‘shopping cart’ icon to be directed to the corresponding website to purchase your selected item.
4. Tap the top left ‘back’ button to head back to the camera page.

Source Code

Refer to our GitHub

Android Application: https://github.com/Vanessa-24/DIPGrp7

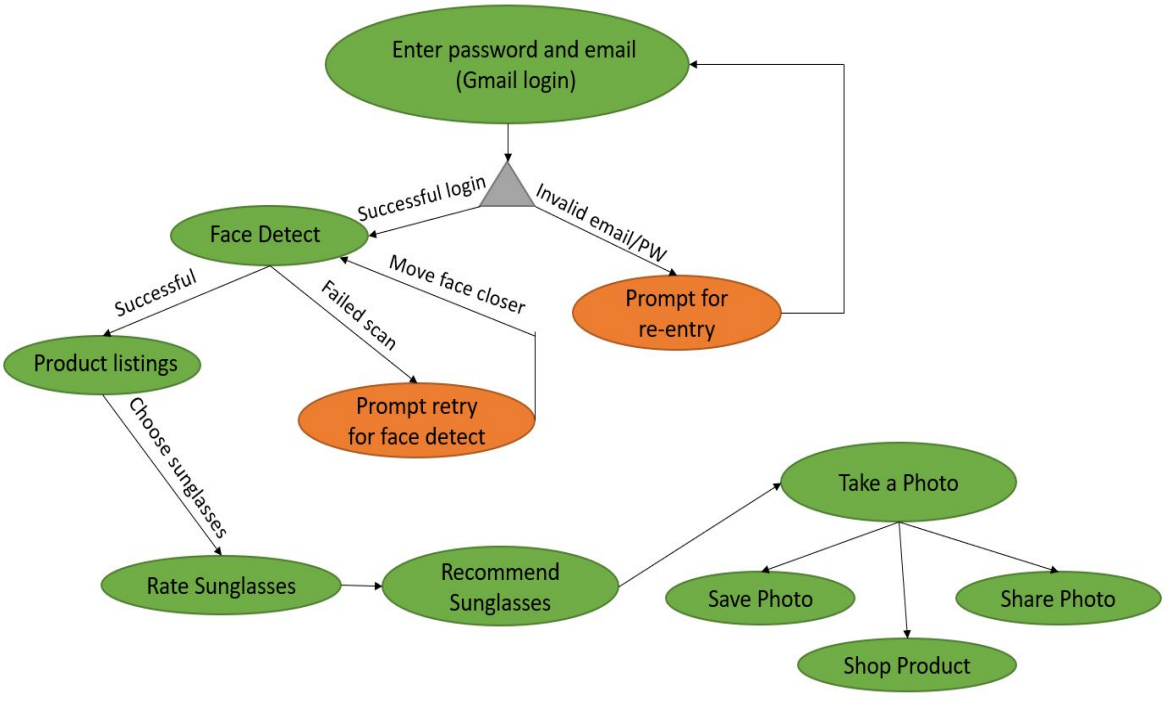
Server hosted on AWS: https://github.com/leminhviett/FaceShapeServer

Others

Use Cases

Use Case 1: First-time user

1. Log in via gmail
2. Click on the camera icon to proceed to camera page
3. Use face detection to obtain face shape result
4. Head to products tab and try all available sunglasses
5. User can rate glasses after trying it on
6. Switch to the recommendation tab to look at suggested models based on user's face shape and past ratings
7. Take a picture with the glasses on
8. Share the picture with your friends, download it, or click on the shopcart to purchase the product on the merchant site



Use Case 2: Repeated user whose log-in information and face shape results have been saved

1. Click on the camera icon to proceed to camera page
2. Try the glasses available in the products tab
3. Rate the products after trying
4. Head to recommendation tab. Press the refresh icon to see new recommended models
5. Take a picture with the model
6. Share the picture with your friends, download it, or click on the shopcart to purchase the product on the merchant site