Bilkent University

Department of Computer Engineering

CS 491: Senior Design Project

Fakenstein

Analysis Report

Group Members:
Yusuf Ardahan Doğru
Atakan Dönmez
Öykü Irmak Hatipoğlu
Elif Kurtay
Cansu Moran

Supervisor: Dr. Selim Aksoy
Innovation Expert: Adnan Erdursun
Jury Members: Dr. Shervin Arashloo and Dr. Hamdi Dibeklioğlu

Analysis Report November 15 , 2021

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.

Contents

1. Introduction

The concerns about the violation of privacy have become more and more prominent in human lives with the ever growing internet. Every day, people are photographed without their consent and appear in the photographs that are uploaded online. On average, an American is caught on camera 75 times a day without being aware of it [1]. This poses a security threat for people on a daily basis. Examples of people trying to protect other people's privacy on social media is increasing everyday, especially with celebrities covering their children's faces with emojis to protect them from media publicity. Moreover, in social events when a picture is going to be taken, there are always a few people who do not want to appear in the photograph. Therefore, before taking a photograph, permission must be sought from all participants. With more widespread use of the internet from day to day, personal data protection laws are becoming stricter. When taking pictures, people have to be more sensitive about the privacy of the data holders. For example, from Fig. 1, it can be seen that Google Street View blurs the faces of people appearing in photographs in order to protect their privacy [2]. Apart from blurring the images, another method to protect the privacy rights of the individuals that appear in the photograph is removing them using external tools such as Adobe Photoshop which people should buy a subscription to use [3], and Magic Eraser which is actually not available for anyone who does not own a Google Pixel 6 and thus have a limited user base [4].

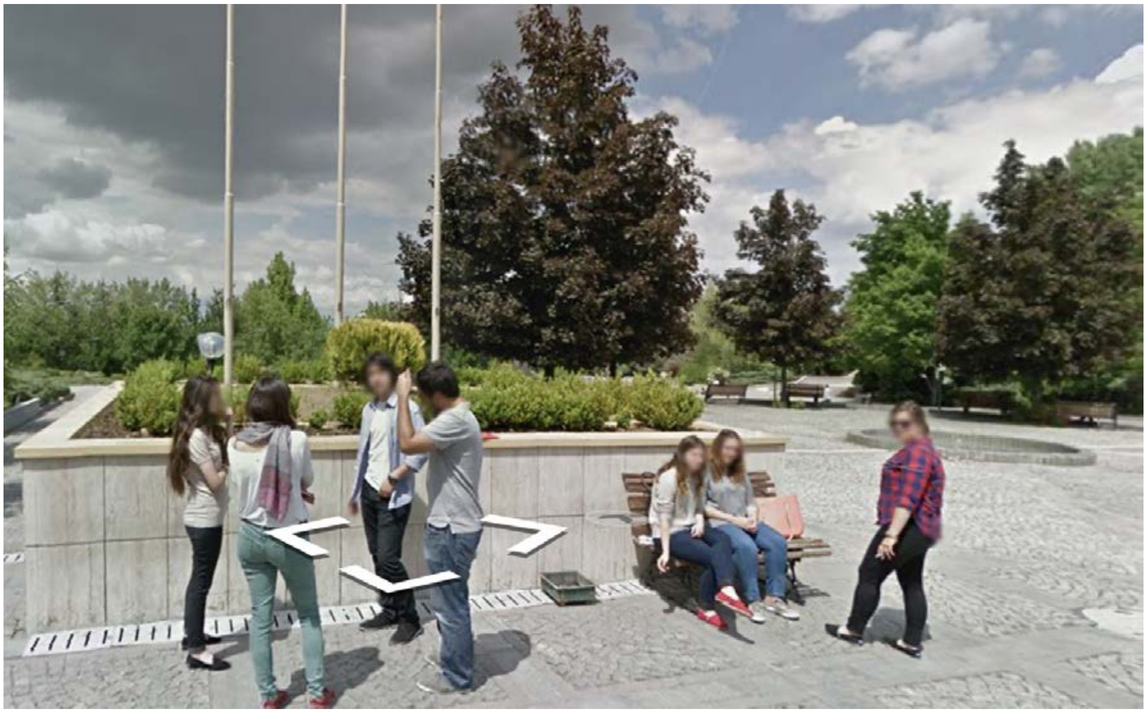


Fig. 1: People with blurred faces in the Google Street View of Bilkent.

Additionally, removing people and filling the remaining space with background is a task that requires advanced knowledge in Adobe Photoshop and would take a significant amount of time and effort. A more user-friendly alternative to remove unwanted people from the photograph is an image manipulation tool named Inpaint designed for the purpose of image restoration [5]. However, this tool requires the user to carefully paint each object to be removed. If the painting is not precise enough or if the photograph has a complicated background, then the Inpaint tool outputs unrealistic photographs. The Inpaint tool does not solely exist to remove humans but it is a general purpose removing algorithm. A complicated background is considered to be a background that does not have consistent lines or has different patterns in close areas.

Our proposed phone application, Fakenstein, aims to realistically replace the faces of unknown people with artificially generated faces in order to protect their anonymity. Unlike Inpaint or Adobe Photoshop, Fakenstein intends to keep user interaction as little as possible while outputting a realistic looking photograph. Today, the style-based GAN algorithms can produce extremely realistic artificially generated faces that do not exist. The examples of such faces can be examined from the website where each time the page is refreshed, the website displays a generated face [6]. For the task of replacing faces in a picture, the guidance of the infamous “deep fakes” can be used. Mostly originated from the GitHub repository “DeepFaceLab”, deep fakes are used to replace faces in videos to generate realistic fake videos. For example, it is possible to replace Iron Man’s face with Tom Cruise’s and produce a video that looks like Tom Cruise is the actual actor playing Iron Man. Using a similar method we aim to obtain seamlessly replaced faces. Additionally, we are trying to build an application that will be available for everyone unlike Magic Eraser which is only available for Google Pixel 6 owners and Adobe Photoshop which can only be used through an expensive subscription.

This report will start with the description of the current systems. Then, we will illustrate our proposed system, Fakenstein. Our system will be described thoroughly by specifying the functional, non-functional, and pseudo requirements. We will further explain our system by providing system models such as scenarios, use case models, object and class models, dynamic models, a navigational path, and screen mockups. We will further analyse our proposed system in terms of engineering design, alternatives and risks, project plan, teamwork, ethical and professional issues, and the learning strategies we use.

2. Current System

In order to protect the privacy of people appearing in pictures without consent, there are certain options available. One of the most common tools for photo manipulation is Adobe Photoshop [3]. With Adobe Photoshop, users can remove people from pictures or manipulate their faces to make them indistinguishable to protect their privacies. While Adobe Photoshop offers a wide selection of tools to achieve such goals, it is an expensive software which requires professional skills, effort and time to perform such tasks. Another alternative is the Inpaint software, which is aimed to make removal of objects from pictures easier [5]. While Inpaint is a free and a more easy-to-use alternative to Adobe Photoshop, the image quality in Inpaint doesn’t reach the quality of a professionally edited photo in Adobe Photoshop. Since Inpaint is an automated software, it doesn’t perform well in backgrounds that are complex. A newly introduced alternative is the Magic Eraser tool that is integrated in the latest Google Pixel phone, Google Pixel 6, released on 25th of October 2021 [4], [7]. Magic Eraser tool in Google Pixel 6 allows users to remove people from pictures automatically without the need of any professional skills. While this tool does not have much review available, since it is fairly new, the first impressions are that the Magic Eraser is quite successful in removing people from pictures while not disrupting the background [4]. However, no matter the quality of the tool, it is only available to a very limited audience, since only the owners of Google Pixel 6 can achieve it. Moreover, in instances where removing people from the picture is not the goal, both Magic Eraser and Inpaint are not applicable tools. For example, imagine an event holder that wants to share a picture taken of the attendees during the conference, but can't share it because there are attendees that don't give consent to their pictures being shared. In that case, removing the attendees from the picture would not be sensible, since the aim of the picture is to demonstrate the participation in the conference.

Considering these tools and their limitations, our proposed application offers a novel approach for protecting the privacy of people in the pictures. Our system doesn't require professional skills since it is automated, aims to create higher quality images than other automated approaches by replacing the faces of people with fake generated faces instead of trying to remove them from the picture, offers a lite version on mobile phones to reach a wider range of people and an extended version for desktop for more professional uses such as newspapers, blogs, etc. and offers a solution for protecting people's privacy even in cases where removal is not the goal.

Since our goal is to replace the faces of people in pictures with fake faces that don't exist, it is also required to mention current fake face generation tools. Replacing people's faces with other faces is already being done with the emerging Deepfake technologies. While visual artists are performing such replacements as a profession, websites such as ThisPersonDoesNotExist.com [6] or Generated.Photos [8] display computer generated fake faces for entertainment purposes. While there is advanced research for generating fake faces, there is no system available that can perform fake face replacement in a given picture in an automated way. In that sense, our proposed system aims to create models inspired by research in fake face generation and advance this technology while combining it with automated picture editing.

3. Proposed System

3.1 Overview

Fakenstein aims to produce realistic photos by replacing people's faces in the background with artificially generated faces in order to preserve people's privacy. Fakenstein will be developed as a cross-platform application and will be available as an Android and desktop application. Our application can be used by uploading a photograph from the gallery. Our algorithm will perform a face detection using Deep Learning models and find all the identifiable faces in the image and tag faces on the background of the photograph. Our main goal is to make the application as automated as possible. However, if the face detection algorithm can’t detect some faces in the picture, the user will be able to manually blur the faces. After the faces are selected, artificially generated faces will replace real human faces using the deepfake method to perform realistic relocation. In order to preserve the nature of the original photograph, the genders, skin colors, and the ages of people will be taken into consideration. This will be possible by performing classification on the detected faces in terms of their sex (man, woman), skin color (fair or dark skin), and age group (children, adult, elderly). After that, the corresponding generated faces will be used according to the classification results. Again, if the face generated does not seem to be suitable, the user will have the capability to change the artificial face with another one by specifying its qualities. Additionally, in order to increase the realisticity, the orientation of the face will be determined by using a head pose estimation algorithm and utilizing an appropriately generated face.

Our main target audience is anyone who uses social media and publishes photographs online. From event photography to news and blog posts, instead of trying to blur, cover, or remove people from photographs, the publishers can easily use Fakenstein to obtain realistic looking photographs that do not violate the privacy of individuals.

3.2 Functional Requirements

3.2.1. User Functionality

The user should be able to:

- download the application via Google Play Store to their mobile phone or via the website of the application to their desktop.
- agree or disagree to the terms and conditions.
- open the gallery on his/her device to upload pictures.
- choose faces to be replaced either from the application’s tagged faces or by tagging them manually.
- manually blur undetected faces.
- undo any blurs made.
- edit the features of the generated faces if the program does not create suitable faces.
- select any of the suggested faces.
- open the Fake Face Library.
- save new faces to the Fake Face Library.
- remove faces from Fake Face Library.
- save the new picture.
- restart the whole process.
- view the tutorial.

3.2.2. System Functionality

The system should be able to:

- open the gallery on the user's device.
- open pictures from the gallery.
- detect all the faces in the picture.
- label the detected faces as the main subject or background.
- enable the user to add or remove faces to be replaced with generated faces.
- enable the user to blur any undetected faces.
- classify the detected face’s sex, skin color, and age group.
- determine the orientation of the faces.
- generate fake faces that are compatible with the detected features of the original faces.
- generate suggested fake face options according to given face specifications.
- open Fake Faces Library.
- save a selected suggested face into Fake Faces Library.
- replace original faces with artificial ones as seamlessly as possible.
- save the final image to the user’s device.

3.3 Nonfunctional Requirements

3.3.1. Usability

- The application should have a simple and self-explanatory interface.
- The language used in the application should be clear and easy to understand.
- The application should enable the user to upload and get their image.

3.3.2. User Friendliness

- The application should not be hard for the user to understand and use.
- The application should include a tutorial section to help users adjust to the application.
- The application should be optimized in order not to bore users with long loading and processing time.
- The user should not be required to know about Deep Learning algorithms that will be used in the application.

3.3.3. Maintainability

- Any open source or third party software used will be required to have long term support for future maintenance.
- The application should have periodic version updates to use the latest versions of the external modules and third party applications.

3.3.4. Extensibility

- The application should be extensible so that in the future it can be used as a plug-in or be integrated by other applications.
- The application should be open to any upgrades considering there could be new functional/non-functional requirements or user interface changes.

3.3.5. Reliability

- The application will be tested for various cases in terms of features.
- The application should not crash for reasons other than operating system based version differences. The reason for the crash must be logged and the user should be informed about it.
- The application should work for differing sizes, brightness, image types, and image formats such as BMP, JPEG, PNG, WebP, and HEIF.

3.3.6. Accessibility

- The Android version of the application should be made available on the Google Play Store to reach a vast amount of users.
- The desktop version of the application should be available for download on the website of the application.

3.3.7. Portability

- The desktop application will be able to run independently from the operating system via Java Virtual Machine.
- The Android application will be able to run independently from the brand or model of the mobile phone. The only constraint should be the version of Android installed.

3.3.8. Efficiency

- The application should be able to run smoothly in older and newer generations of Android smartphones with Android Marshmallow 6.0 or higher installed.
- The picture upload should not take more than approximately 10 seconds (subject to change).
- The application should provide the user with the output image under approximately 20 seconds (subject to change).

3.3.9. Scalability

- The application is made for daily use, so available platforms and users may change as there are new developments.
- The service we are providing will not be affected by user/customer number as the application will work locally once downloaded.

3.4 Pseudo Requirements

- Fakenstein will be an Android application for personal use and a desktop application for professional use.
- Android Studio IDE will be used to create an Android application, using Java.
- The desktop application will also be written in Java so that it should run at every platform having JVM.
- Desktop application will be developed in Intellij IDE.
- Python will be used to implement Deep Learning algorithms.
- Pytorch framework will be used for Deep Learning tasks.
- The website will be created using HTML, CSS, SCSS, and JavaScript.
- Object Oriented Programming (OOP) paradigm will be adopted throughout the project.
- The online database Firebase and its own servers will be used for the project for saving fake face instances.
- For the desktop app, a local file storage system will be used next to the online database for saving the fake faces in Face Library.
- The website of this project will be hosted by GitHub servers.
- Git and GitHub will be used for version control and GitHub Issues will be used for issue tracking.
- An extensive Terms and Conditions will be included in the application.

3.5 System Models

3.5.1 Scenarios

Scenario 1

Use case: Accept terms and conditions
Primary actor: Mobile and desktop user
Entry condition: The user opens the application for the first time.
Exit condition: The user accepts the terms and conditions.
Main flow of events:
User:

- scrolls to view the terms and conditions.
- accepts the terms and conditions.

Scenario 2

Use case: Upload an image from the gallery

Primary actor: Mobile and desktop user
Entry condition: The user clicks the *Start* button on the Main Menu.
Exit condition: User clicks *Select Image* or *Cancel* button.
Main flow of events:
User:

- clicks the *Start* button in the Main Menu.
- chooses a picture from the device gallery.
- selects *Select Image* button to proceed processing the image or *Cancel* button to return to the Main Menu.

Scenario 3

Use case: Select faces to be replaced in the image
Primary actor: Mobile and desktop user
Entry condition: User has uploaded an image and is in the Face Selection page.
Exit condition: User clicks *Replace Faces* button, *Cancel* button or Return icon.
Main flow of events:
User:

- views all the detected faces with their bounding boxes.
- views the faces selected for replacing by the app itself.
- zooms in/out of the picture to view the image in more detail.
- adds/removes faces to be replaced by clicking on the desired face.
- either approves the face selections and proceeds to replace the faces by clicking *Replace Faces* button, discards the changes and goes back to main menu by clicking *Cancel* button or goes back to the picture upload page by clicking the Return icon.

Scenario 4

Use case: Blur undetected faces
Primary actor: Mobile and desktop user
Entry condition:User has uploaded an image and is in the Face Selection page.
Exit condition: The selected area is blurred.
Main flow of events:
User:

- clicks the Blur tool.
- blurs an undetected face by circling the face.

Scenario 5

Use case: Swap faces with fake generated ones
Primary actor: Mobile and desktop user
Entry condition: User has clicked the *Replace Faces* button in Face Selection page or *Replace* button in the Fake Faces Editing page.

Exit condition: User clicks *Done* button, *Cancel* button or Return icon.
Main flow of events:

- Fake faces are generated according to the detected specifications of the selected faces to be replaced or according to the manually given specifications by the user.
- The real faces are replaced with the fake faces in the image automatically if the user has entered the page for the first time from the Face Selection page or by clicking the *Replace* button if there are changes made to the face specifications.
- The user either approves the changes and finalizes the image by clicking the *Done* button, discards all the changes and goes back to the Main Menu by clicking the *Cancel* button or goes back to the Face Selection page by clicking the Return icon.

Scenario 6

Use case: View and edit the features of the replaced fake faces
Primary actor: Mobile and desktop user
Entry condition: User has selected a face to edit in the Fake Faces Editing page.
Exit condition: User clicks *Apply button*, *Remove* button or x icon.
Main flow of events:
The user:

- clicks on a face in the Fake Faces Editing page.
- adds/removes the face from the selected faces to be replaced.
- views/edits the face specifications which include the age, gender, race of the person and their head orientation in the image.
- saves the new specifications by clicking the *Apply* button, discards the specification changes by clicking the x icon or removes the fake face completely by clicking *Remove* button.

Scenario 7

Use case**:** Select a new fake face from suggested faces
Primary actor: Desktop user
Entry condition: User has selected a face to edit in the Fake Faces Editing page.
Exit condition: User clicks *Apply* button, *Remove* button or x icon.
Main flow of events:
The user:

- views fake face options that are generated according to the detected face.
- specifications or given face specifications.
- selects one of the suggested faces.
- saves the face selection by clicking the *Apply* button, discards all changes by clicking the *x* icon, or removes the fake face completely by clicking *Remove* button. If there is no suggested face selected and the *Apply* button is clicked, while swapping faces, a new random face will be generated according to the specifications.

Scenario 8

Use case**:** Save a suggested fake face to the face library
Primary actor: Desktop user
Entry condition: User has selected a suggested fake face.
Exit condition: User clicks *Save Face* or *Don’t Save Face* button.
Main flow of events:
The user:

- selects one of the suggested faces.
- enters a name for the fake face to be saved.
- saves the selected face to the library by clicking the *Save Face* button or doesn’t save it by clicking *Don’t Save Face* button.

Scenario 9

Use case**:** Select a fake face from the face library
Primary actor: Desktop user
Entry condition: User has selected a face to edit in the Fake Faces Editing page and opened the face library.
Exit condition: User clicks *Select Face* button, *Remove Face* button or x icon.
Main flow of events:
The user:

- views saved fake faces in the face library.
- selects one of the faces in the library.
- either saves the face selection by clicking the *Select Face* button, removes the face from the library by clicking the *Remove Face* button or discards any selection and closes the library by clicking the x icon.

Scenario 10

Use case: Finalize and save image to the gallery
Primary actor: Mobile and desktop user
Entry condition: User has clicked the *Done* button in Fake Faces Editing page.
Exit condition: The user either clicks *Return to Main Menu* button or clicks the Return icon.
Main flow of events:
The user:

- views the finalized image
- can save the image to their local gallery by clicking the *Save Image* button.
- returns to Main Menu by clicking *Return to Main Menu* button or goes back to editing by clicking the Return icon.

Scenario 11

Use case: View the tutorial
Primary actor: Mobile and desktop user
Entry condition: User has clicked the *Tutorial* button in the Main Menu.
Exit condition: User clicks the *Exit* button.
Main flow of events:
The user:

- views the tutorial.
- goes back to the Main Menu by clicking the *Exit* button.

3.5.2 Use Case Model

The use case model of our application is shown in Figure 2. The model displays different types of actors and use cases that each actor can perform. By using inheritance between actors, we aim to distinguish basic features available to any user and additional features available specifically to “Desktop User”.

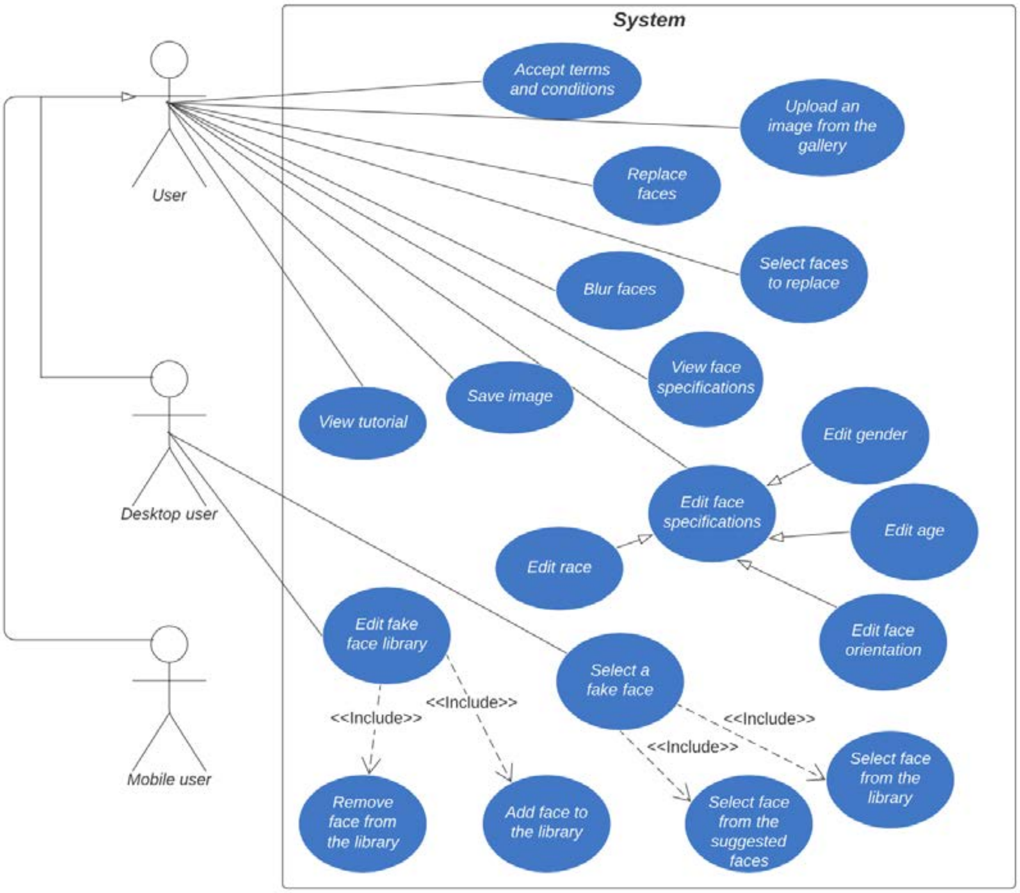


Fig. 2: The use case model for the application.

3.5.3 Object and Class Model

This section includes an object class diagram to demonstrate our initial design for software architecture. The purpose of the diagram is to model the relationships and qualities of envisaged objects and classes of the application. Since the project is currently at the analysis phase, the details of this model might change in the future during implementation. Below are detailed descriptions of the classes we have included in our object and class model.

The **Image** class is responsible for the main form of input that the user provides to the application, namely a photo that the user wants to edit. It has the width, height and matrix attributes. Width and height are straightforward; they store the size values of the image as int values. The matrix has the size width x height x 3 (for Red-Blue-Green(RGB) channels) where each matrix element represents one color channel of a pixel of the image. The Image class has two helper classes: **ImageExporter** and **Magnifier**. The ImageExporter class is responsible for exporting the output image, with the modified faces, to the user's gallery. The Magnifier class is responsible for zooming in and out of the photo to better identify and modify faces in it.

The **FaceIdentifier** class is responsible for detecting the faces in the input photo. It has an Image, a face detection model, and a background detection model for attributes. By using these two models, the FaceIdentifier class detects all face instances in the photo and labels the background faces.

The **Boundary** class provides an interface to represent boundaries of the faces in the Image. It allows sites in the Image to be modified and edited. It has two types, **BoxBoundary** and **CircleBoundary**, both of which contain the area attribute which is stored in a ImageTypes data type. The BoxBoundary class contains additional attributes width and height to represent a bounding box, and the CircleBoundary class contains the attribute radius to represent a boundary circle. While BoxBoundary is used during detection and editing of the faces, CircleBoundary is used while using the blur tool, which is provided by the **BlurModel** class which takes the Boundary and blurs inside of it.

The **Face** class stores the values of faces in a photo. It actually is a cropped version of the photo provided to the application, so it has the width and height attributes. The faces in a photo are identified by the FaceIdentifier class, and the attributes and details of the faces are stored in the Face class itself. After the FaceIdentifier class identifies the faces, certain attributes are saved for the face. The attribute name is a string which is a specific name for the face that is given by the application. The backGround attribute is a boolean value that specifies if the identified face is a background face or not, which is provided by the FaceIdentifier class. The area attribute is the area that the face takes up in the photo, and the values for it are provided by the Boundary class. The skinColor attribute is a string which the skin color of the identified face is stored. The ageGroup attribute stores the age group that the face belongs to. The gender attribute stores the gender value in an int format. An additional attribute is the Pose, which determines the pose of the face and includes details like which way the face faces, the tilting of the head and so on. The Pose attribute is identified and taken by the **Pose** class.

The **FaceModifier** class is responsible for modifying the faces in the input photo. It has an Image, a face classification model, a face generation model, a face combination model and the connection to the database of the faces. These models allow the class to efficiently modify the faces in the Image that is given. The class also contains an array of Face class named faces, which are the faces in the input photo. The availableFaces array holds boolean values for the face array. Each face in the faces array has a corresponding boolean value in the availableFaces array that shows whether the face is available for modifying or not. The class also has an attribute called restrictedMode which is a boolean that shows whether the application is operating on the desktop or mobile. RestrictedMode is True when the application is operating on mobile. Therefore, the options of viewing suggested faces and access to Face Library are restricted. The class can get the face information from the FaceIdentifier and Face classes, classify the faces accordingly, generate, suggest and combine artificially generated faces to replace the original face, and update the faces in the Image.

The **DatabaseExtractor** class provides an interface to the database that stores the faces in photos. One of its helper classes, the **OnlineDatabaseHelper** class, holds an array of generated faces and a boolean value called connectToDatabase that indicates whether the database was connected to or not. The other helper class of the DatabaseExtractor is the **LocalDatabaseHelper** class, which contains the array of the saved fake faces in the Face Library for a user. It provides the adding and removing functions to the user's local face database, and searching and showing the faces specific to the user. The Faces are therefore stored in the database both locally and globally. The DatabaseExtractor class as an interface takes on the role of an intermediary between the database and the application.

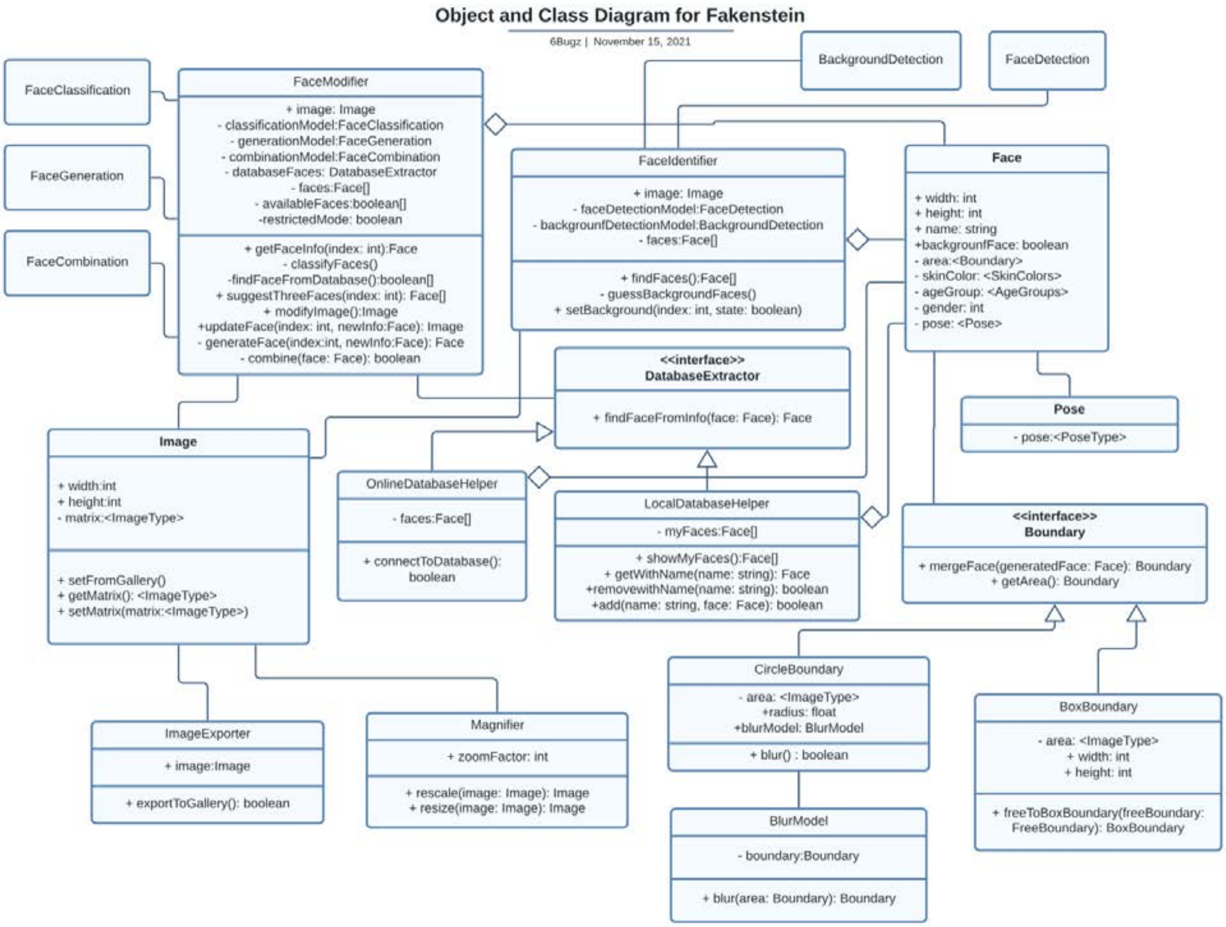

Object and Class Diagram for Fakenstein
6Bugz | November 15, 2021
FaceClassification
FaceGeneration
FaceCombination
FaceModifier
+ image: Image
- classificationModel:FaceClassification
- generationModel:FaceGeneration
- combinationModel:FaceCombination
- databaseFaces: DatabaseExtractor
- faces:Face[]
- availableFaces:boolean[]
-restrictedMode: boolean
+ getFaceInfo(index: int):Face
- classifyFaces()
-findFaceFromDatabase():boolean[]
+ suggestThreeFaces(index: int): Face[]
+ modifyImage():Image
+updateFace(index: int, newInfo:Face): Image
- generateFace(index:int, newInfo:Face): Face
- combine(face: Face): boolean
BackgroundDetection
FaceDetection
FaceIdentifier
+ image: Image
- faceDetectionModel:FaceDetection
- backgrounfDetectionModel:BackgroundDetection
- faces:Face[]
+ findFaces():Face[]
- guessBackgroundFaces()
+ setBackground(index: int, state: boolean)
Face
+ width: int
+ height: int
+ name: string
+backgrounfFace: boolean
- area:<Boundary>
- skinColor: <SkinColors>
- ageGroup: <AgeGroups>
- gender: int
- pose: <Pose>
<<interface>>
DatabaseExtractor
+ findFaceFromInfo(face: Face): Face
Pose
- pose:<PoseType>
Image
+ width:int
+ height:int
- matrix:<ImageType>
+ setFromGallery()
+ getMatrix(): <ImageType>
+ setMatrix(matrix:<ImageType>)
OnlineDatabaseHelper
- faces:Face[]
+ connectToDatabase():
boolean
LocalDatabaseHelper
- myFaces:Face[]
+ showMyFaces():Face[]
+ getWithName(name: string): Face
+removewithName(name: string): boolean
+add(name: string, face: Face): boolean
<<interface>>
Boundary
+ mergeFace(generatedFace: Face): Boundary
+ getArea(): Boundary
CircleBoundary
- area: <ImageType>
+radius: float
+blurModel: BlurModel
+ blur() : boolean
BoxBoundary
- area: <ImageType>
+ width: int
+ height: int
+ freeToBoxBoundary(freeBoundary:
FreeBoundary): BoxBoundary
BlurModel
- boundary:Boundary
+ blur(area: Boundary): Boundary
ImageExporter
+ image:Image
+ exportToGallery(): boolean
Magnifier
+ zoomFactor: int
+ rescale(image: Image): Image
+ resize(image: Image): Image

Fig. 3: The Object and the Class Diagram for the application.

In Figure 3, we provide the object class diagram which is the overall scheme that combines the classes described above, using the appropriate relationships. We introduce the notation <DataType> to refer to a platform/library specific data type or unclear data types that have the potential to be represented in more than one way. In addition, getter and setter functions of variables will automatically be included in all classes during implementation. However, they were not written in the diagram to provide a simpler overview of the application.

Note that there will be other classes that are necessary to carry out the graphical user interface. Since they do not shape the main architecture of the application, they were not included in the object class diagram in the Analysis stage. The given model specifically focuses on the vital methods and attributes, and keystone classes to provide a better understanding. A complete overview will be provided with the High-Level Design Report that captures the technical details as well.

3.5.4 Dynamic Models

3.5.4.1 Sequence Diagrams

Main Menu:

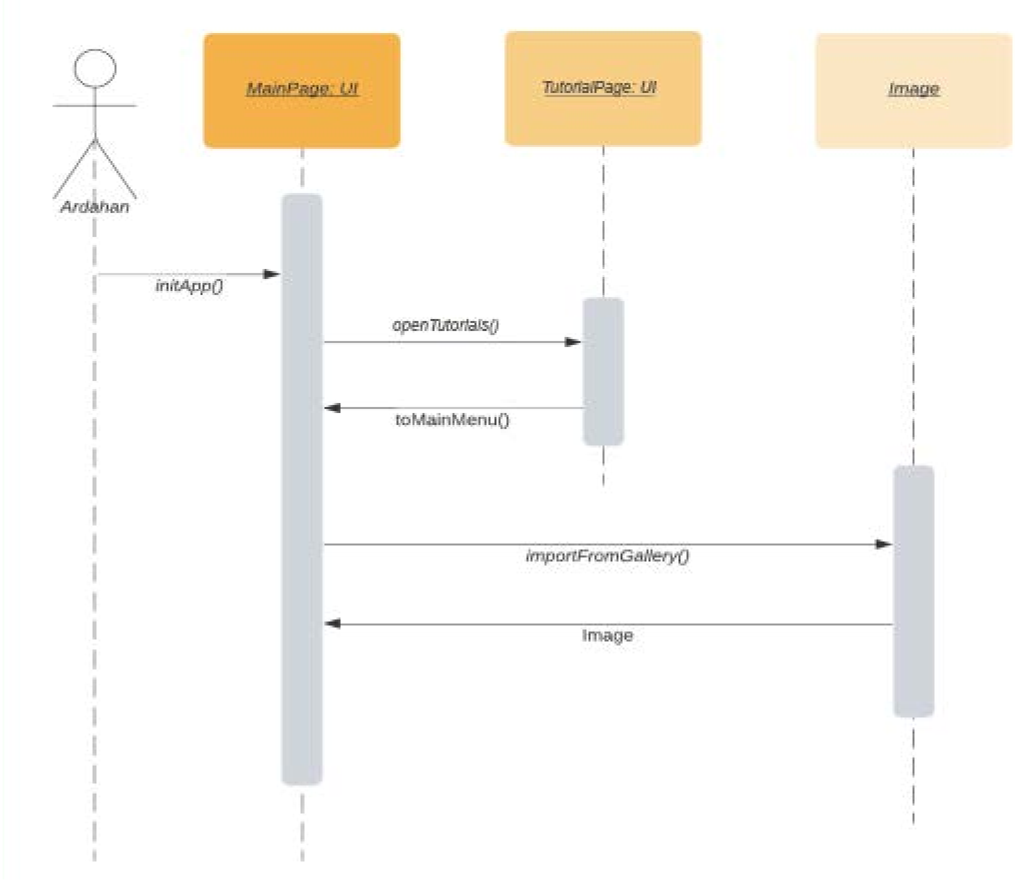


Fig. 4: The sequence diagram for the main menu actions

The figure demonstrates the simple Main Menu actions that the user can make, and the corresponding functions evoked while doing so. For the specific figure, after opening the app, the user enters the Tutorial page, which triggers the openTutorials() function. Then, the user goes back to the Main Menu, and proceeds with uploading an image, which invokes importFromGallery(). The user then returns to the main menu after accessing the settings page, or importing an image.

Modifying Information of a Generated Face:

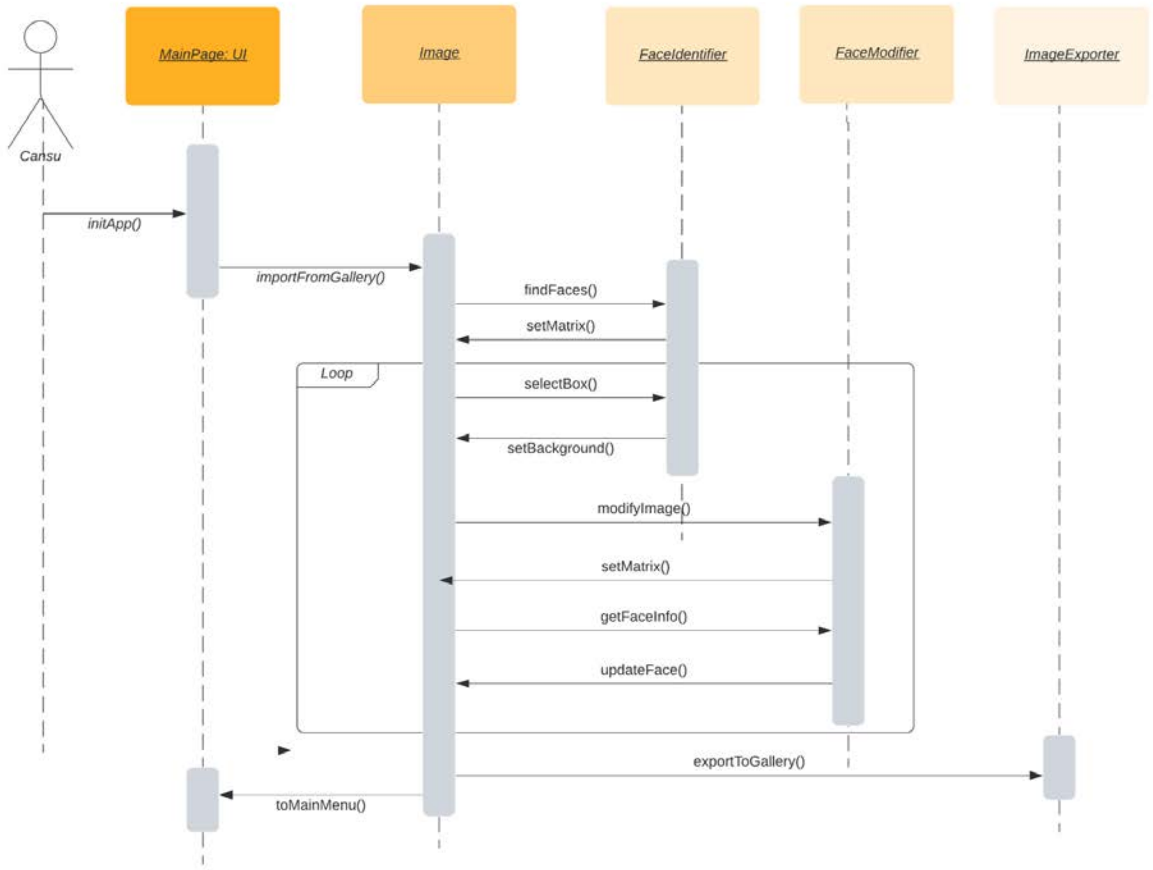


Fig. 5: The sequence diagram for the face information modification

The figure demonstrates the steps that are taken to modify the information of the generated faces. After the user opens the application and uploads a photo, the application starts to detect and classify the faces in the photo using Deep Learning algorithms and generates faces accordingly. After this process is over, the user is able to modify the detected face specifications. When choosing to do so, the user can change the selected faces subject to replacement, or modify the information of a generated face. The faces are updated according to the changes. After the user finalizes an image, they can export it to the gallery and return back to the main menu.

Face Blurring:

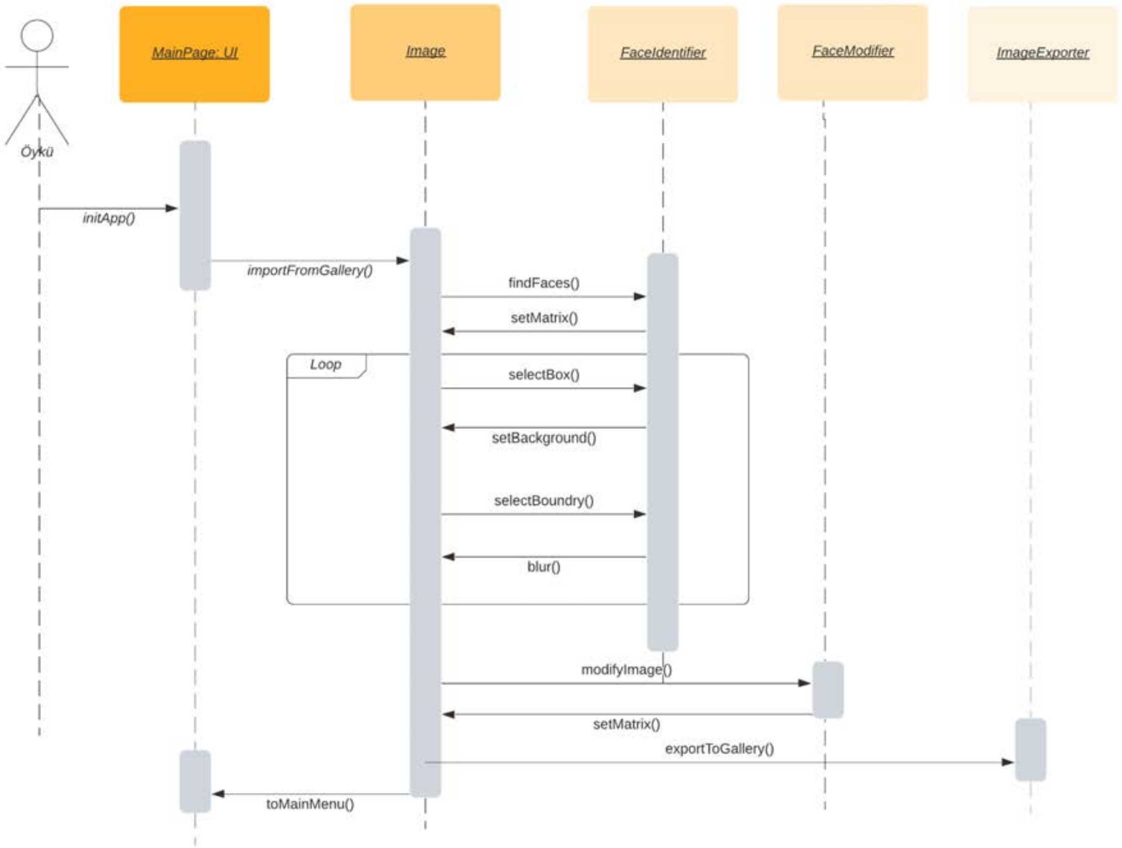


Fig. 6: The sequence diagram for blurring a face

The figure demonstrates the steps that are taken to blur faces in a photo that the user uploads to the application. After the user opens the application and uploads a photo, the application detects the faces in the photo. If there are undetected faces in the photo, the user can blur them by selecting a boundary and using the blur tool. The rest of the face modification process is the same as the previous sequence diagram displaying face modification sequence.

For Desktop App Select A Different Fake Face:

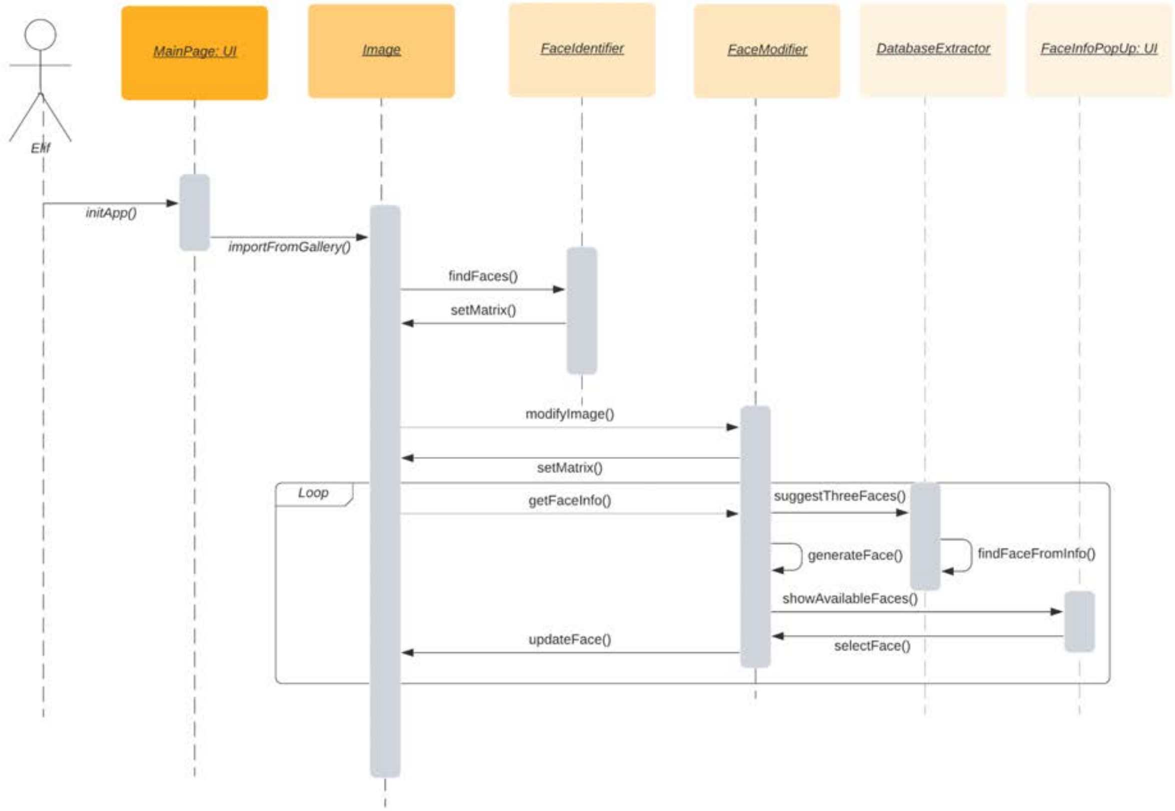


Fig. 7: The sequence diagram for selecting an alternative fake face to replace the original

The figure demonstrates the steps that are taken to select a generated face for the desktop application. While modifying the faces in the image, the desktop user is given the option to select a face from the suggested fake faces. After the user selects a suggested fake face for the people in the photo, the faces, and therefore the photo, is updated accordingly.

For Desktop User, Save and Remove a Fake Face:

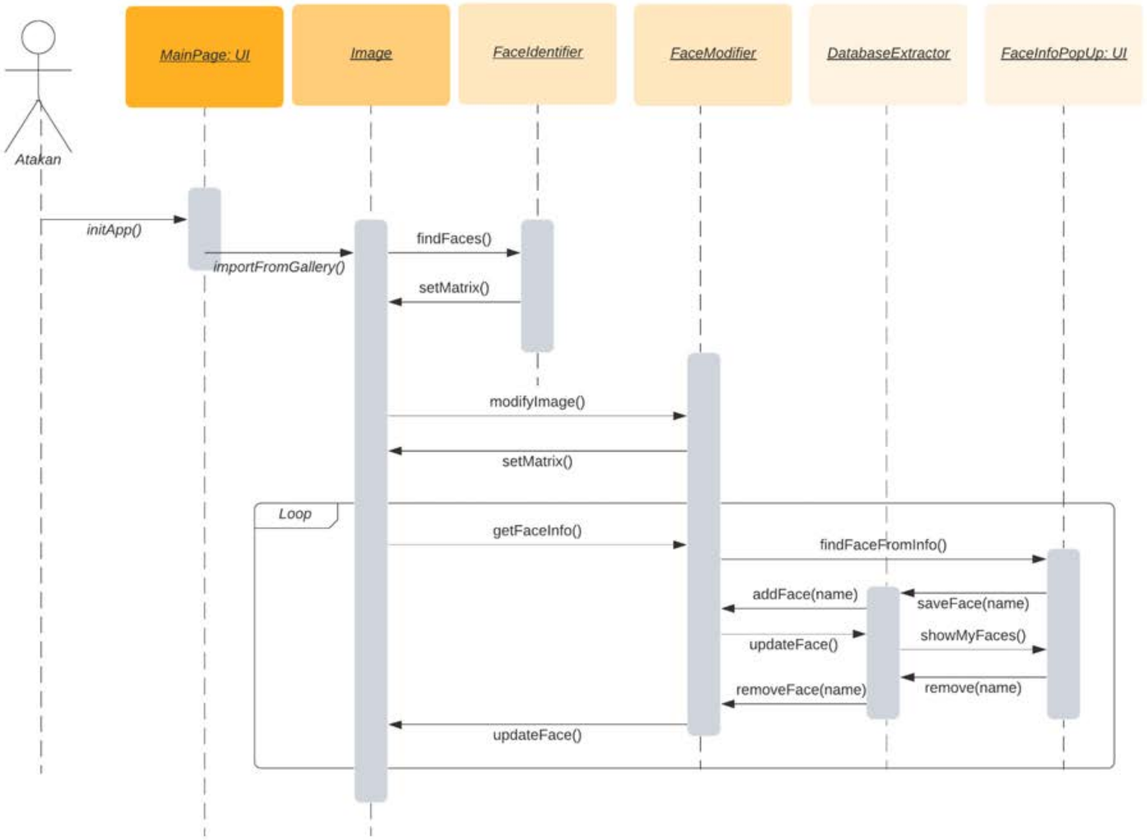


Fig. 8: The sequence diagram for the user to save or remove a fake face

The figure demonstrates the steps that are taken to select a face from the Face Library, save a face to the Face Library and remove a face from the Face Library in the desktop application. While modifying the faces in the image, the desktop user is given the option to select a face from the saved fake faces in the Face Library. When the Face Library is open, the user can select a face from the library, or remove a selected face from the library. The user can also save a suggested fake face to the Face Library with a name. If the user selects a face from the library, the photo is updated accordingly.

3.5.4.2 Activity Diagram

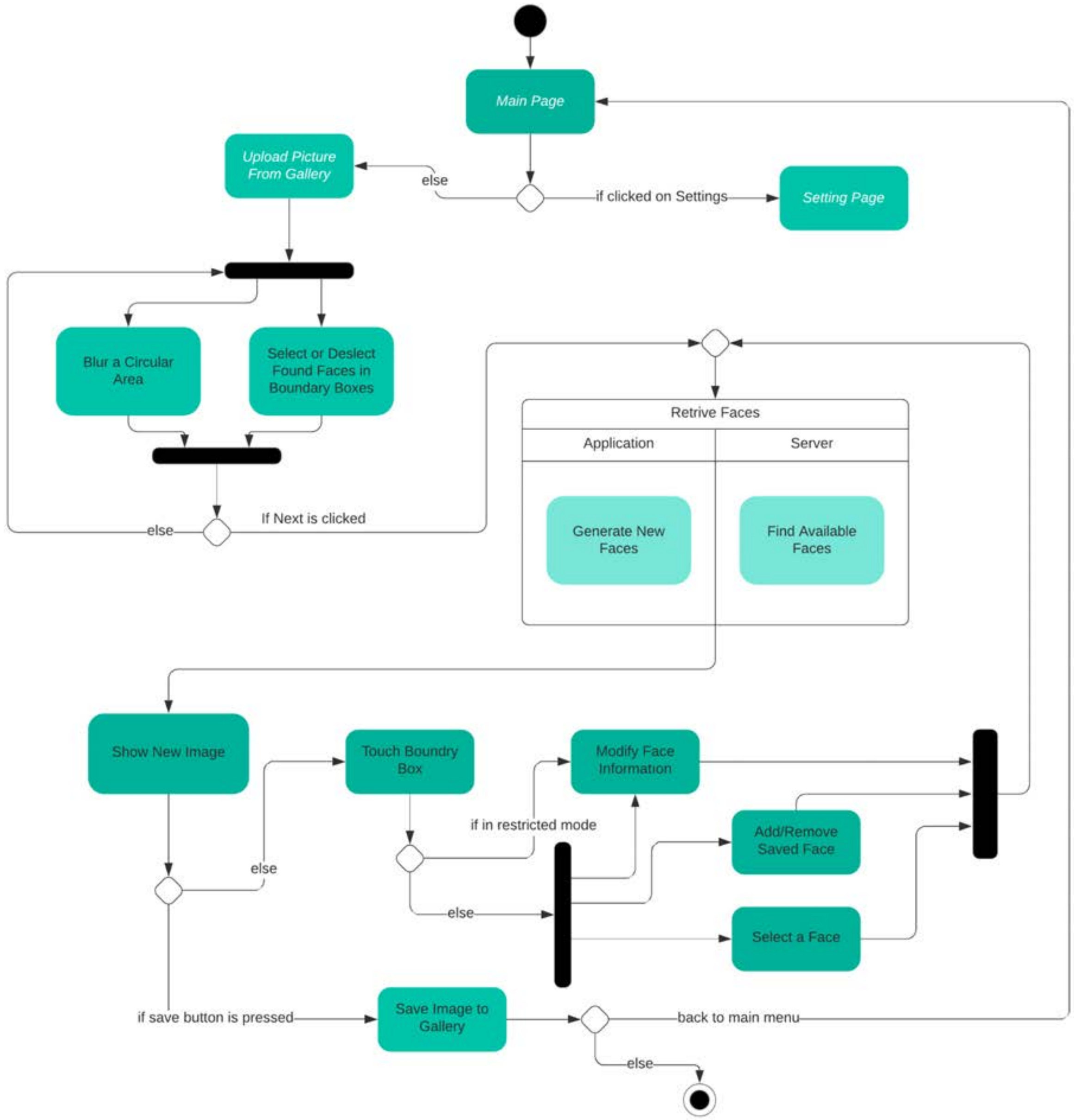


Fig. 9: The Activity Diagram for the application

The activity diagram in Figure 9 displays the main flow of the application. In the diagram, light green filling is used to indicate model states, dark green filling with white text is used to indicate user interface(UI) activities, and dark green filling with black text is used to indicate actions in the application. Restricted mode specifies whether the application is mobile or desktop version.

3.5.5 User Interface - Navigational Paths and Screen Mock-ups

3.5.5.1 Navigational Path

Figure 10 displays the user navigational path of the application. For simplicity purposes, both desktop and mobile navigational paths are shown in the same figure. While both mobile and desktop navigational paths include the pages shown in white, the page indicated with blue is only included in the desktop navigational path. It should be noted that the path only shows the flow of between the pages of the application and does not include any tools (blur tool, different face specifications that can be edited, etc.), since they are encapsulated in the pages themselves. In order to provide an easy-to-follow navigational path, only one directional path is displayed below. In other words, while returning to the previous page is available via the Return icon in the pages, they are not included in the figure. Although the Terms and Conditions page is the first page displayed when the application is opened for the first time, it is not included in the navigational path since it is only displayed once and is not part of the normal flow.

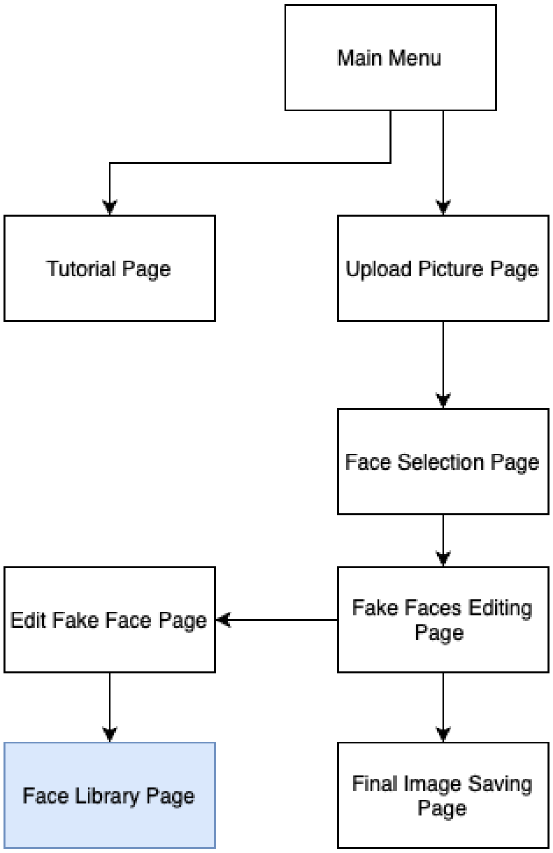


Fig. 10: The Navigational Path for the User in the application

3.5.5.2 Screen Mock-ups

The screen Mock-ups will be provided in order to better describe our proposed system. Our system will work on two different platforms: mobile (Android) and desktop (Windows).

3.5.5.2.1 Mobile Mock-ups

3.5.5.2.1.1 Main Menu Screen

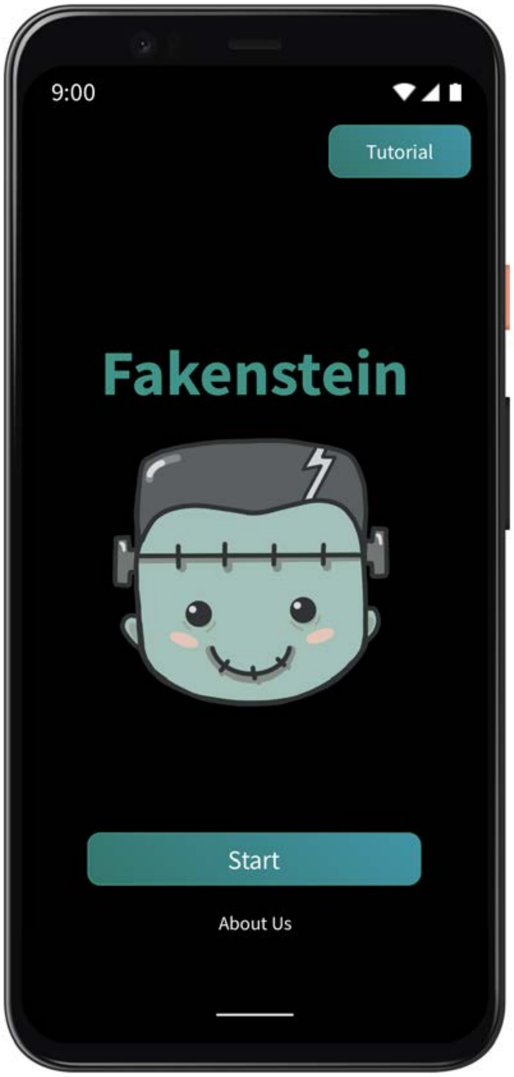


Fig. 11: The Main Menu screen design for the application

The Main Menu is the opening screen of the app, if it is not being opened for the first time (when opening the app for the first time, the Terms and Conditions is the opening screen). From the Main Menu, the user can click About Us to view information about the development team, Tutorial to view the tutorial on how to edit pictures using the app and Start to start editing a picture.

3.5.5.2.1.2 Picture Upload Screen

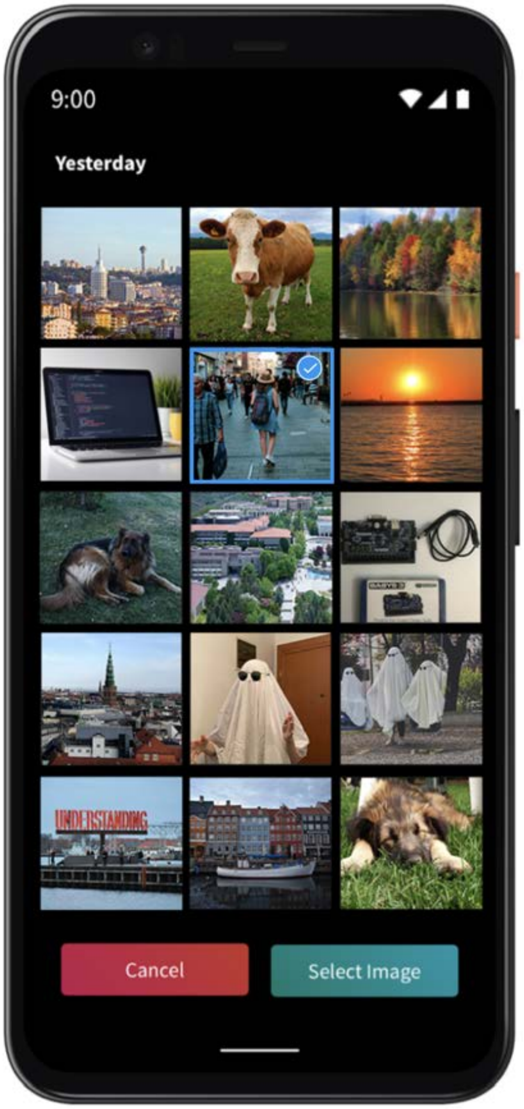


Fig. 12: The Picture Upload Screen Design for the application

Picture Upload screen is displayed following the Start button in Main Menu. In this screen, the user's local gallery is opened where the user can choose any image to edit. After selecting an image from the gallery, the user can proceed editing the image by clicking the Select Image button or cancel the process and return back to the Main Menu by clicking the Cancel button.

3.5.5.2.1.3 Detecting Faces Loading Screen

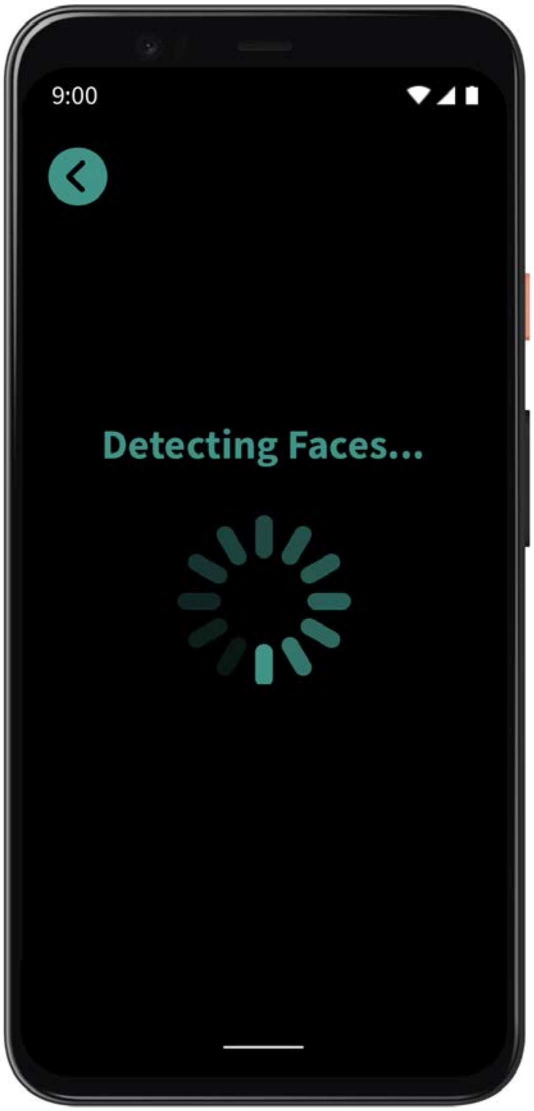


Fig. 13: The Detecting Faces Loading Screen Design for the application

After a picture is uploaded, the app will require a certain amount of time to detect all the faces in the picture and tag faces that seem to be in the background or faces that the app predicts to be subject to replacement. While the app is processing the image, a loading page will be displayed.

3.5.5.2.1.4 Face Selection Screen

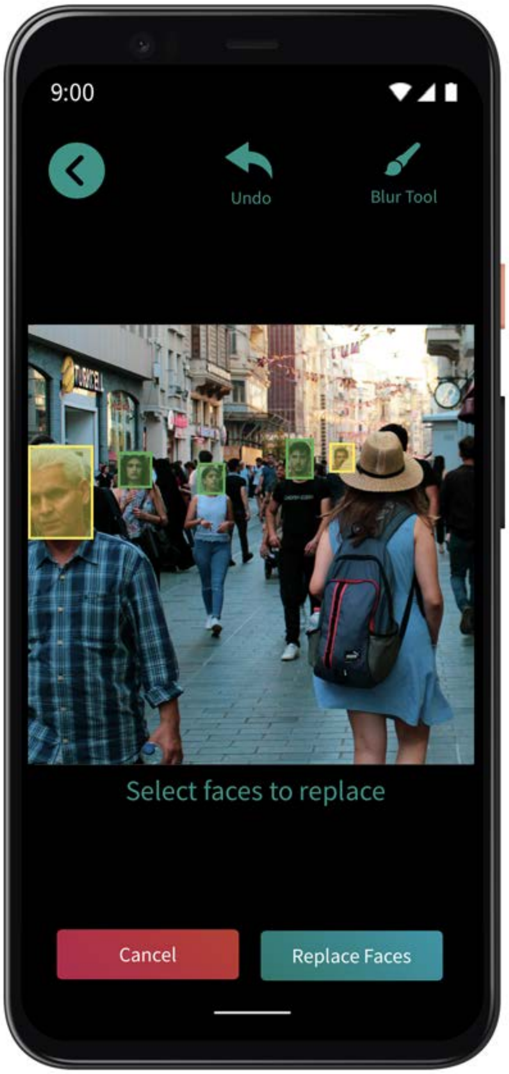


Fig. 14: The Face Selection Screen Design for the application

Face Selection screen displays all the faces detected by the app in addition to tagged faces that the app will take as faces to replace with their corresponding bounding boxes. The faces that are detected but not tagged for replacement (meaning they are the faces in the focus of the photo) are shown in green bounding boxes while faces that will be replaced (faces in the background) are shown in yellow bounding boxes. The user can add/remove faces to be replaced by simply clicking on the bounding box of the face. According to the action performed, the color of the bounding box will change. The page also includes the blur tool, which is given to the users in case there are faces in the picture that the app was unable to detect. In that case, by clicking the blur tool, the users can blur an undetected face. The user can undo each change by clicking the Undo button on top of the screen. In addition, the user can return to the Picture Upload page by clicking Return icon, cancel the process entirely and return back to the Main Menu by clicking cancel or proceed replacing the selected faces by clicking Replace Faces button.

3.5.5.2.1.5 Fake Faces Loading Screen

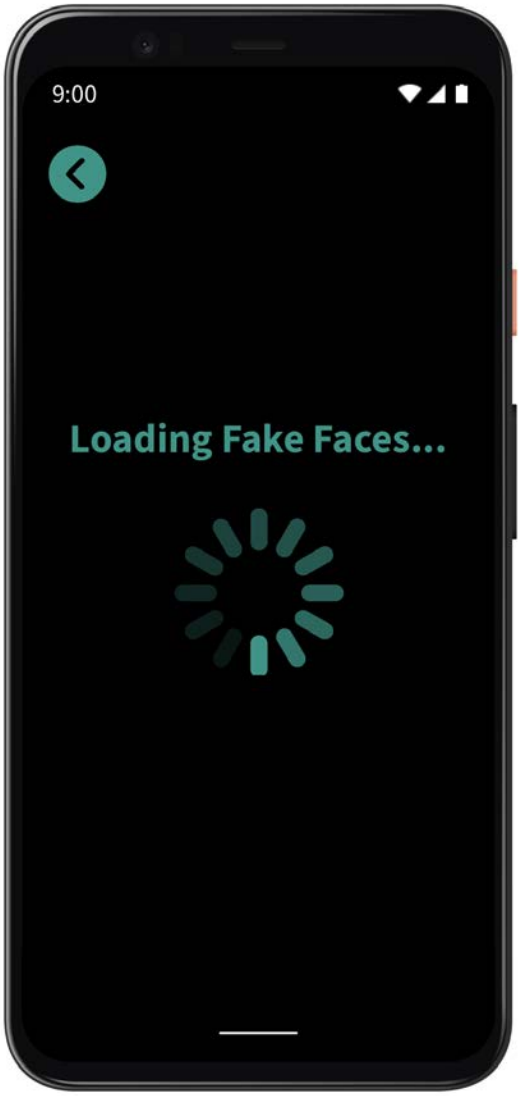


Fig. 15: The Fake Face Loading Screen Design for the application

After faces are selected for replacement, the app needs to classify each face to be replaced according to age, race, gender and face orientation. According to the classification, appropriate fake faces will be generated and replaced with the corresponding real face. Since this process takes time, the user will be displayed a loading screen.

3.5.5.2.1.6 Fake Faces Editing Screen

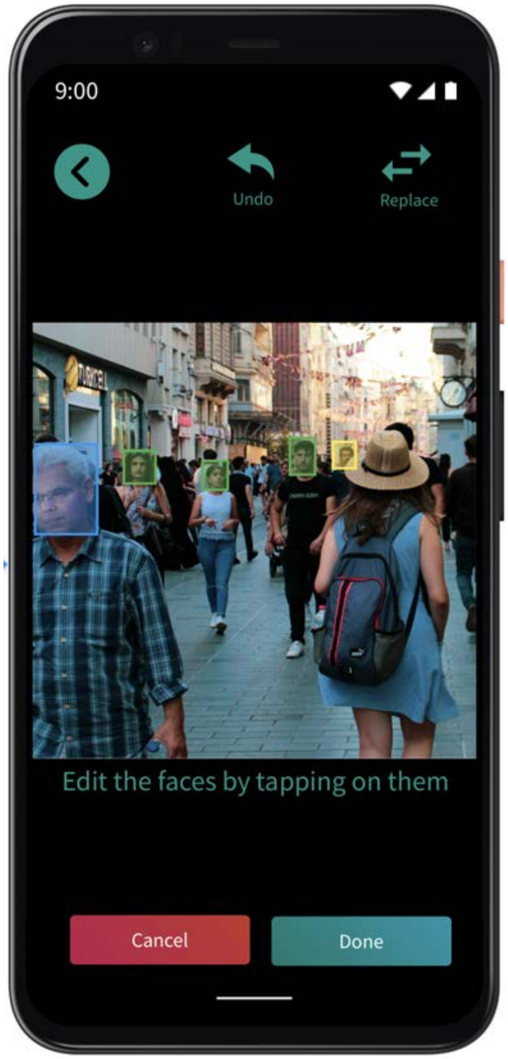
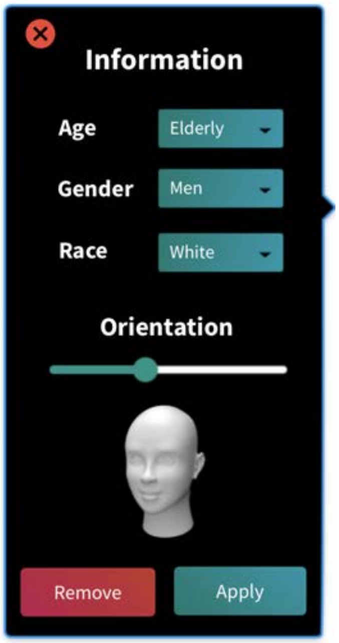


Fig. 16: The Fake Face Editing Screen Design for the application

After the faces in the picture have been replaced with fake faces, the user will be given the opportunity to edit any changes that were made to the picture. For example, if the detection of a certain face was incorrect, the user can click on the corresponding face to change the fake specifications. The user can also choose to add more faces to be replaced or remove faces to be replaced in this screen. To perform any operation on any face, the user needs to tap on to the corresponding face. When a face selection has been made, the bounding box of the face will turn blue and an editing pop-up screen will appear displaying the face specifications. After editing the face specifications, the faces won't be replaced with new fake faces according to given specifications immediately. The faces will be replaced when the Replace button on top of the screen is clicked. The user can undo any change by clicking the Undo button, return back to the Face Selection screen by clicking the Return icon, cancel the process and go back to the Main Menu by clicking the Cancel button or finalize the image by clicking the Done button.

3.5.5.2.1.7 Fake Face Editing Popup



Age: Elderly
Gender: Men
Race: White

Fig. 17: The Fake Face Editing Pop-up Design for the application

Fake Face Editing Popup appears when a face is selected on the Fake Face Editing screen. In the pop-up, detected face specifications are listed which include the age, gender, race and face orientation of the person. The user can make any changes to these specifications in this screen by changing their corresponding values. The user can close the pop-up and discard all changes in the pop-up screen by clicking the x button, apply the changes by clicking the Apply button or remove the fake face from the picture and return to the real face by clicking the Remove button.

3.5.5.2.1.8 Final Image Save Screen

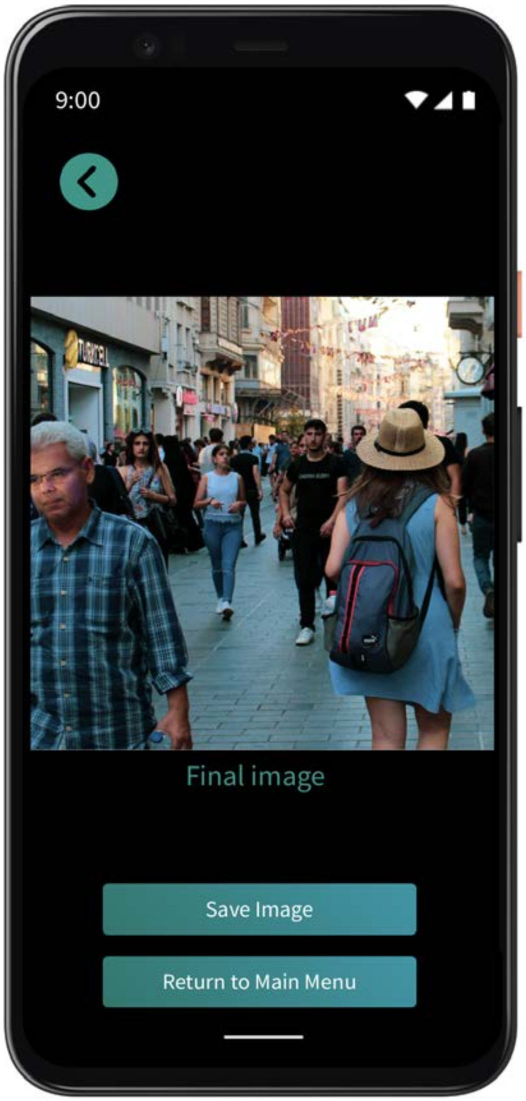
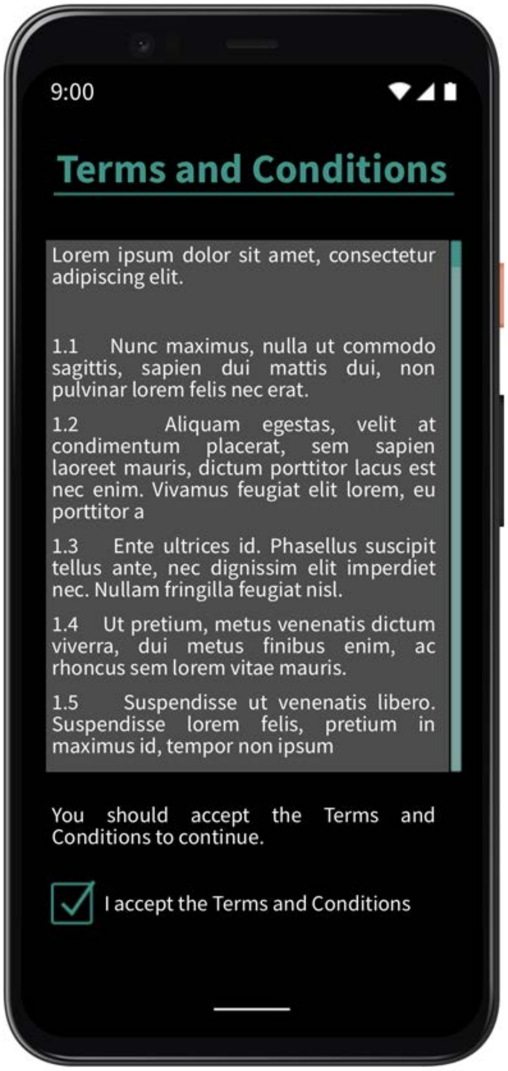


Fig. 18: The Final Image Save Screen Design for the application

After the user clicks the Done button in the Fake Face Editing screen, the image is finalized and ready to be saved. The user can save the image to their device by clicking Save Image button or return to the Main Menu without saving by clicking Return to Main Menu button or go back to editing the image by clicking the Return icon. It should be noted that no user images are stored by the application itself. Therefore, if the user chooses to return to the main menu without saving, the image will be lost.

3.5.5.2.1.9 Terms and Conditions Screen



Terms and Conditions

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

1.1 Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat.

1.2 Aliquam egestas, velit at condimentum placerat, sem sapien laoreet mauris, dictum porttitor lacus est nec enim. Vivamus feugiat elit lorem, eu porttitor a

1.3 Ente ultrices id. Phasellus suscipit tellus ante, nec dignissim elit imperdiet nec. Nullam fringilla feugiat nisl.

1.4 Ut pretium, metus venenatis dictum viverra, dui metus finibus enim, ac rhoncus sem lorem vitae mauris.

1.5 Suspendisse ut venenatis libero. Suspendisse lorem felis, pretium in maximus id, tempor non ipsum

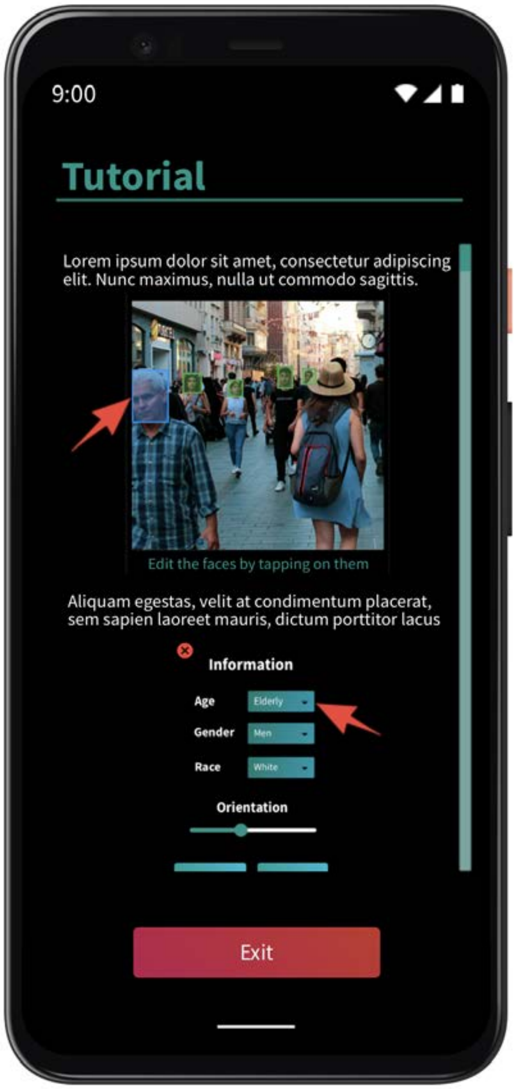
You should accept the Terms and Conditions to continue.

☑ I accept the Terms and Conditions

Fig. 19: TheTerms and Conditions Screen Design for the application

Terms and Conditions screen is the first screen that is displayed when the application is opened for the first time. The user cannot proceed using the application before accepting Terms and Conditions. Once the Terms and Conditions are accepted in a device, this screen won’t be displayed again.

3.5.5.2.1.10 Tutorial Screen



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis.

Edit the faces by tapping on them

Aliquam egestas, velit at condimentum placerat, sem sapien laoreet mauris, dictum porttitor lacus

Information

Age: Elderly
Gender: Men
Race: White

Fig. 20: The Tutorial Screen Design for the application

Tutorial screen displays a tutorial about the application and how to use it. The user can scroll up and down on the Tutorial screen to view the tutorial. The user can exit from the Tutorial screen and return back to the Main Menu by clicking the Exit button.

3.5.5.2.2 Desktop Mock-ups

3.5.5.2.2.1 Main Menu Screen

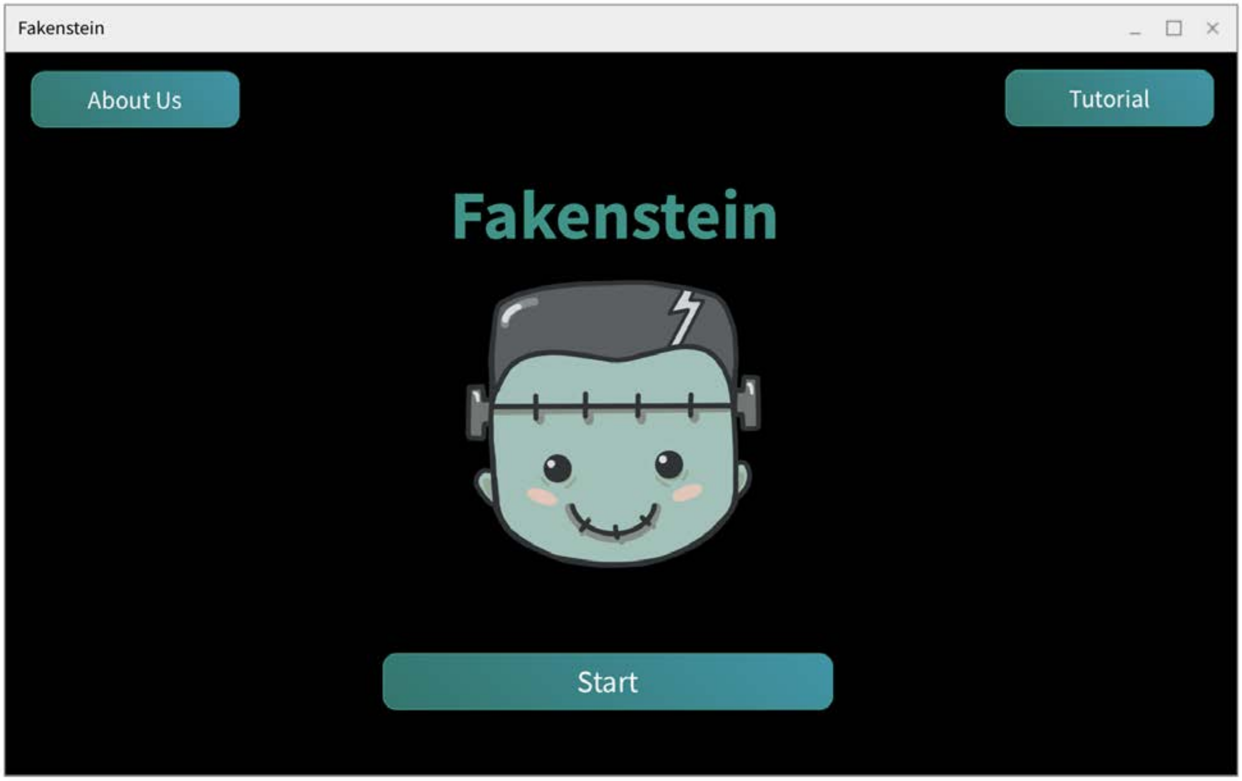


Fig. 21: The Main Menu Screen Design for the desktop application

The Main Menu is the opening screen of the app, if it is not being opened for the first time (when opening the app for the first time, the Terms and Conditions is the opening screen). From the Main Menu, the user can click About Us to view information about the development team, Tutorial to view the tutorial on how to edit pictures using the app and Start to start editing a picture.

3.5.5.2.2.2 Picture Upload Screen

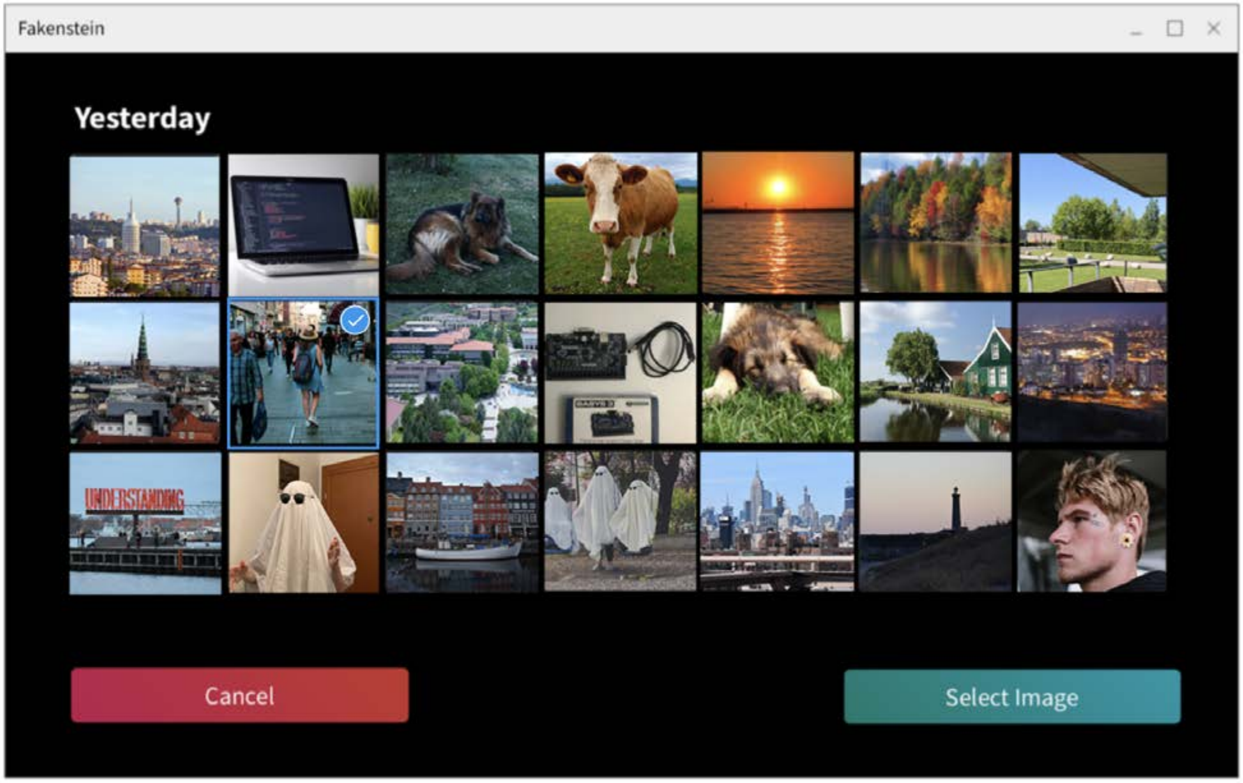


Fig. 22: The Picture Upload Screen Design for the desktop application

Picture Upload screen is displayed following the Start button in Main Menu. In this screen, the user’s local gallery is opened where the user can choose any image to edit. After selecting an image from the gallery, the user can proceed editing the image by clicking the Select Image button or cancel the process and return back to the Main Menu by clicking the Cancel button.

3.5.5.2.2.3 Detecting Faces Loading Screen

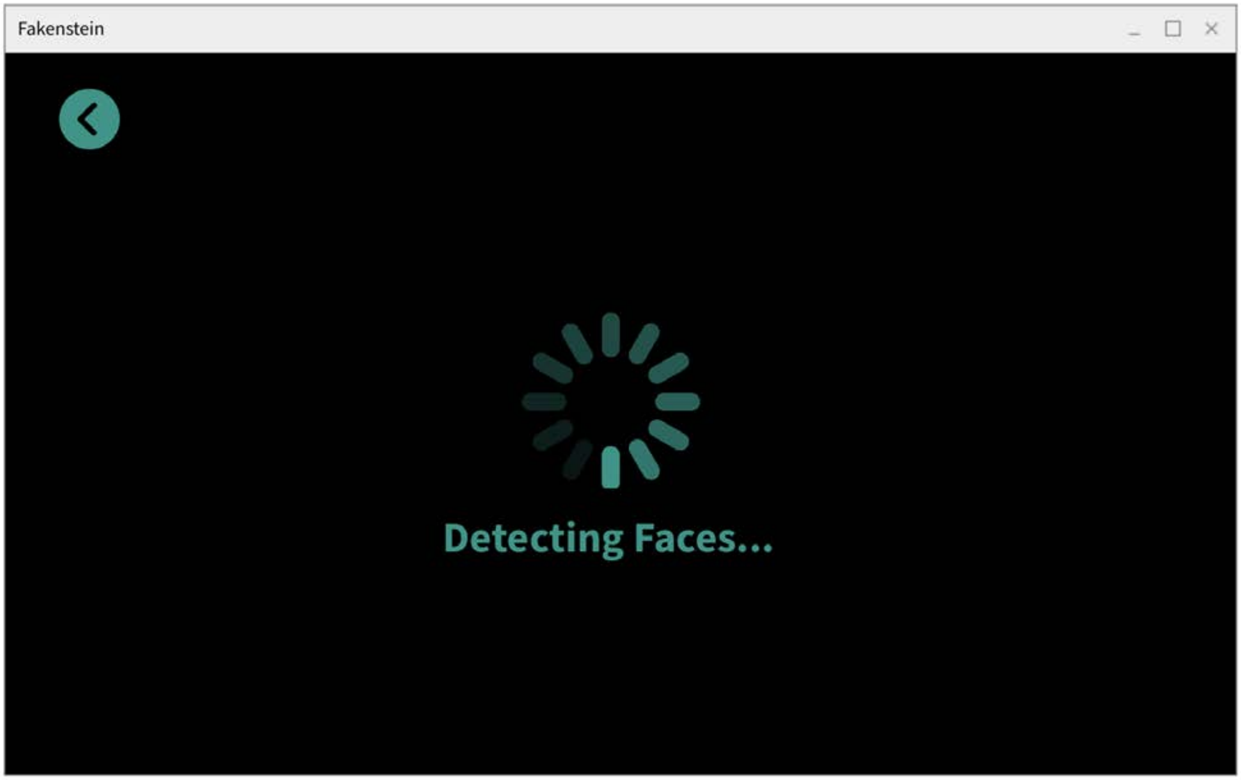


Fig. 23: The Detecting Faces Loading Screen Design for the desktop application

After a picture is uploaded, the app will require a certain amount of time to detect all the faces in the picture and tag faces that seem to be in the background or faces that the app predicts to be subject to replacement. While the app is processing the image, a loading page will be displayed.

3.5.5.2.2.4 Face Selection Screen

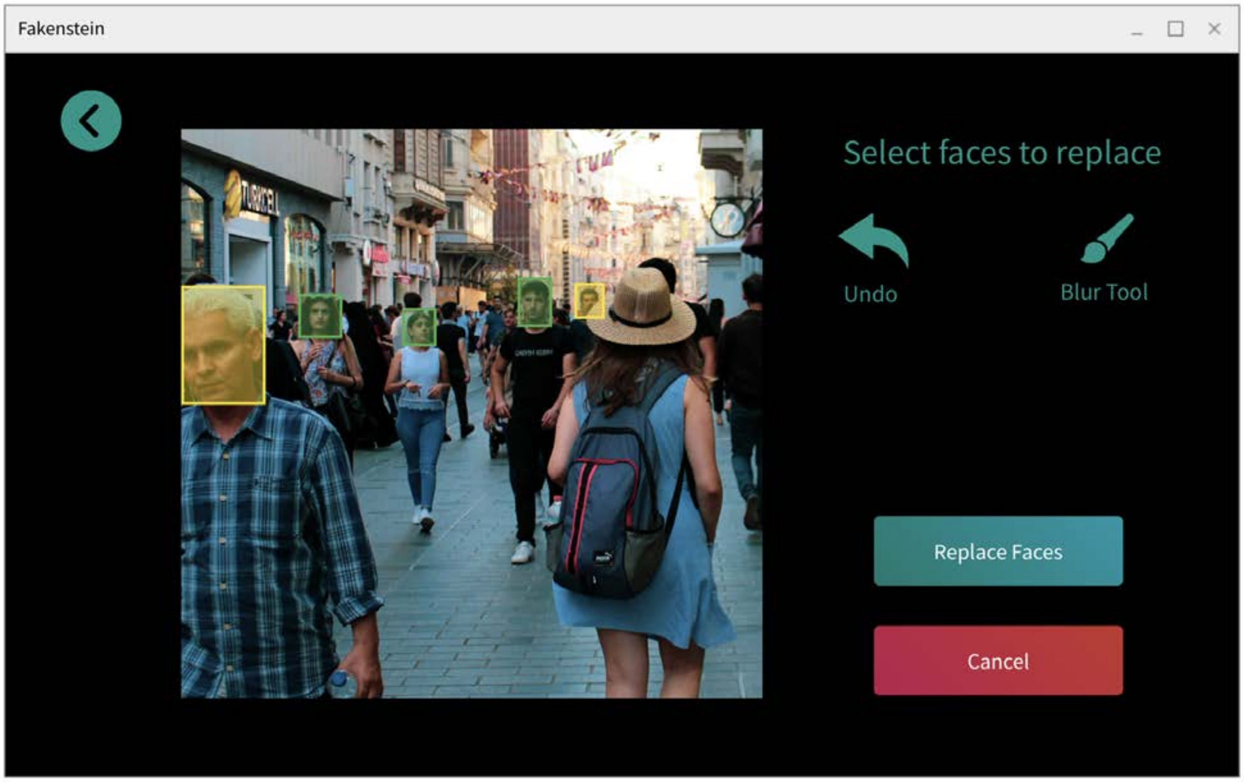


Fig. 24: The Face Selection Screen Design for the desktop application

Face Selection screen displays all the faces detected by the app in addition to tagged faces that the app will take as faces to replace with their corresponding bounding boxes. The faces that are detected but not tagged for replacement are shown in green bounding boxes while faces that will be replaced are shown in yellow bounding boxes. The user can add/remove faces to be replaced by simply clicking on the bounding box of the face. According to the action performed, the color of the bounding box will change. The page also includes the blur tool, which is given to the users in case there are faces in the picture that the app was unable to detect. In that case, by clicking the blur tool, the users can blur an undetected face. The user can undo each change by clicking the Undo button on top of the screen. In addition, the user can return to the Picture Upload page by clicking the Return icon, cancel the process entirely and return back to the Main Menu by clicking cancel or proceed replacing the selected faces by clicking Replace Faces button.

3.5.5.2.2.5 Fake Faces Loading Screen

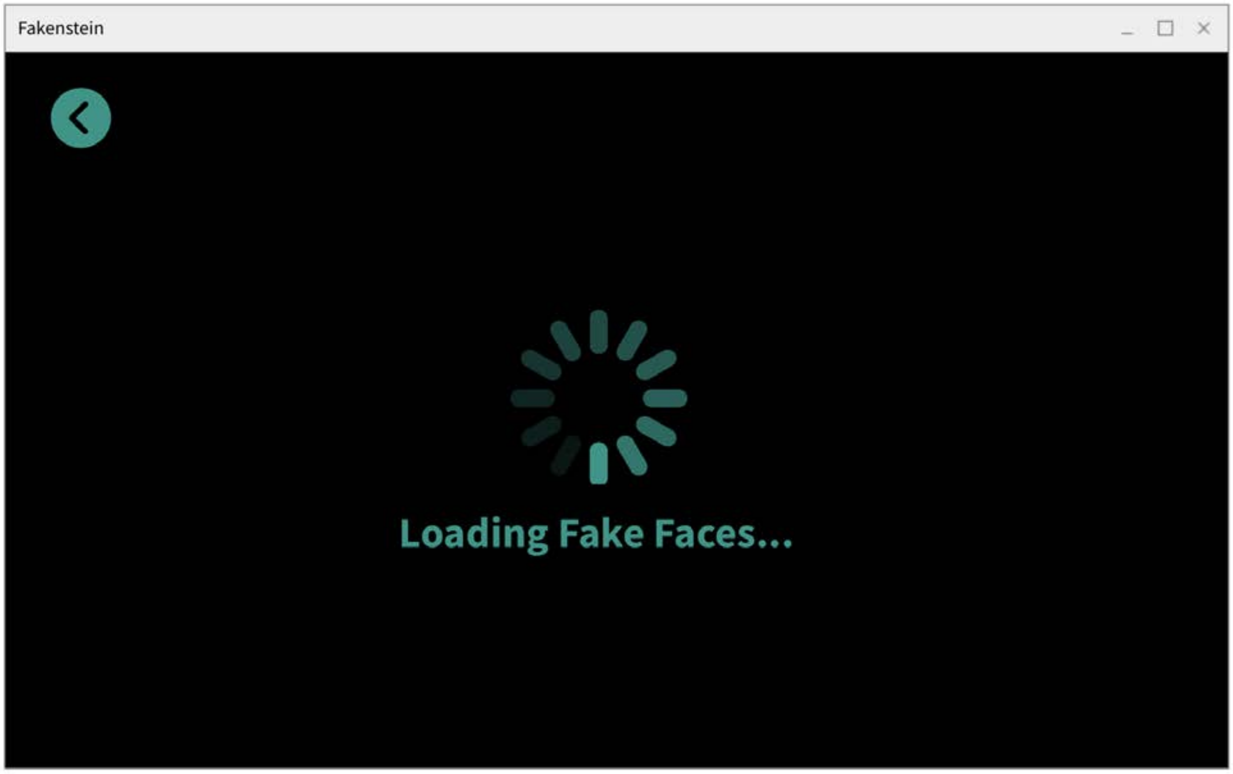


Fig. 25: The Fake Faces Loading Screen Design for the desktop application

After faces are selected for replacement, the app needs to classify each face to be replaced according to age, race, gender and face orientation. According to the classification, appropriate fake faces will be generated and replaced with the corresponding real face. Since this process takes time, the user will be displayed a loading screen.

3.5.5.2.2.6 Fake Faces Editing Screen

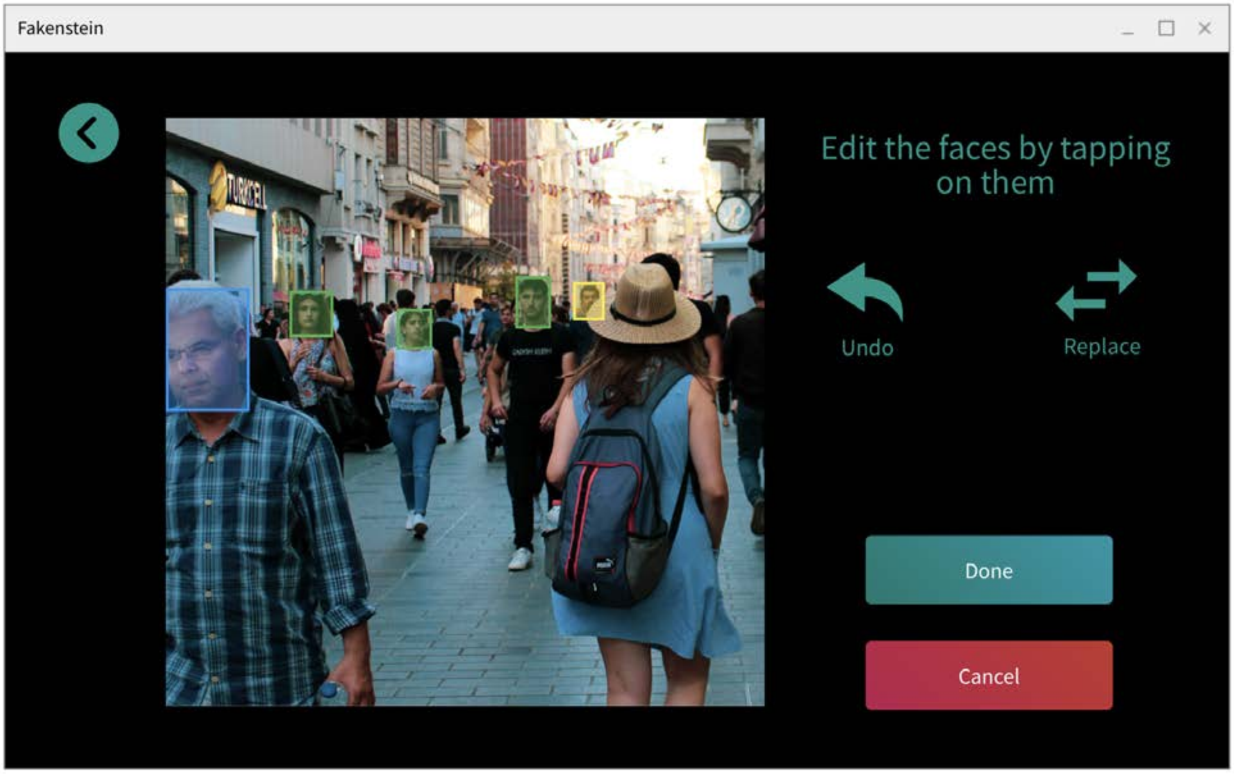
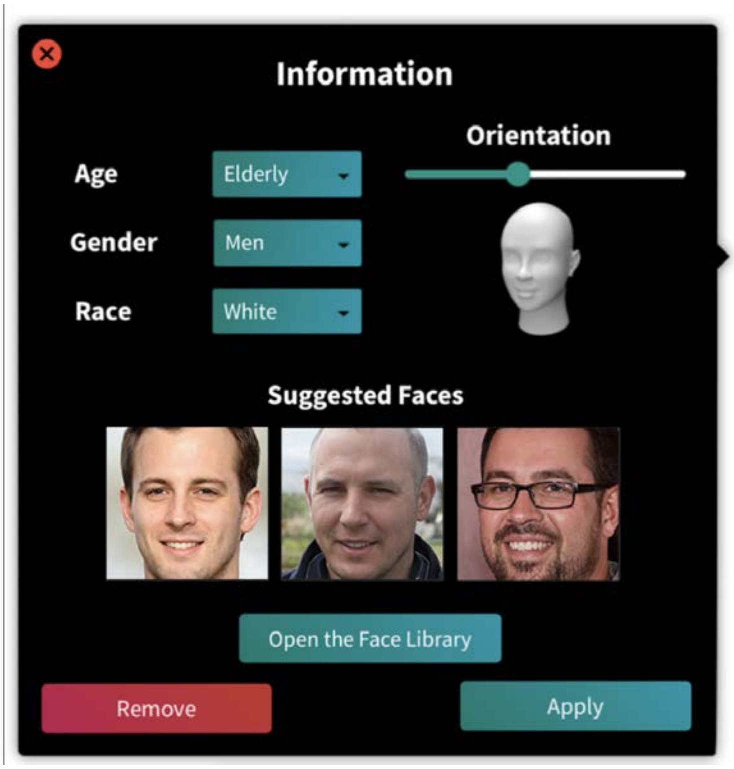


Fig. 26: The Fake Faces Editing Screen Design for the desktop application

After the faces in the picture have been replaced with fake faces, the user will be given the opportunity to edit any changes that were made to the picture. For example, if the detection of a certain face was incorrect, the user can click on the corresponding face to change the fake specifications. The user can also choose to add more faces to be replaced or remove faces to be replaced in this screen. To perform any operation on any face, the user needs to tap on to the corresponding face. When a face selection has been made, the bounding box of the face will turn blue and an editing pop-up screen will appear displaying the face specifications. After editing the face specifications, the faces won't be replaced with new fake faces according to given specifications immediately. The faces will be replaced when the Replace button on top of the screen is clicked. The user can undo any change by clicking the Undo button, return back to the Face Selection screen by clicking the Return icon, cancel the process and go back to the Main Menu by clicking the Cancel button or finalize the image by clicking the Done button.

3.5.5.2.7 Fake Face Editing Popup



Age Elderly

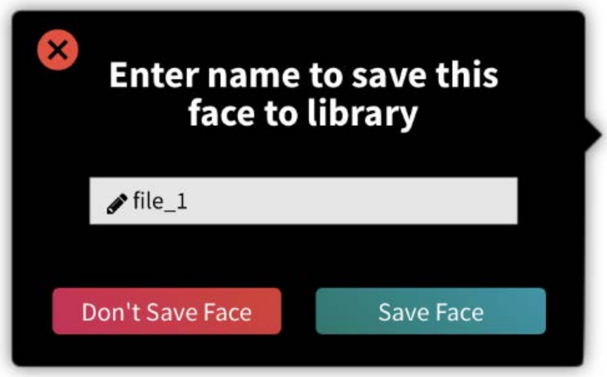
Gender Men

Race White

Fig. 27: The Fake Face Editing Pop-up Design for the desktop application

Fake Face Editing Popup appears when a face is selected on the Fake Face Editing screen. In the pop-up, detected face specifications are listed which include the age, gender, race and face orientation of the person. The user can make any changes to these specifications in this screen by changing their corresponding values. Unlike the mobile version, the desktop version also offers a selection of fake faces that are generated according to given specifications to allow users to choose a fake face instead of giving them a random face. If the user selects one of the suggested faces, they will be asked whether they would like to save that face to the face library or not. The user can also choose a face from their saved faces in the face library. The user can close the pop-up and discard all changes in the pop-up screen by clicking the x button, apply the changes by clicking the Apply button or remove the fake face from the picture and return to the real face by clicking the Remove button. If the user has selected a face from suggested faces or from the face library when the Apply button is clicked, the selected face will be the replacement for the real face in the picture. If no faces are selected in the pop-up, a new face will be generated for replacement according to the given specifications.

3.5.5.2.8 Fake Face Save Popup



Enter name to save this face to library

file_1

Fig. 28: The Fake Face Save Pop-up Design for the desktop application

When a user selects a face from suggested faces, they will be asked if they would like to save the face in face library for further use. The user needs to enter a name for the fake face to be able to save it to the face library. Users can proceed by saving the face by clicking the Save Face button or not save the face and use it only for that one time by clicking Don’t Save Face. If the user has selected the wrong face by accident he/she can exit via the x button.

3.5.5.2.9 Face Library Popup

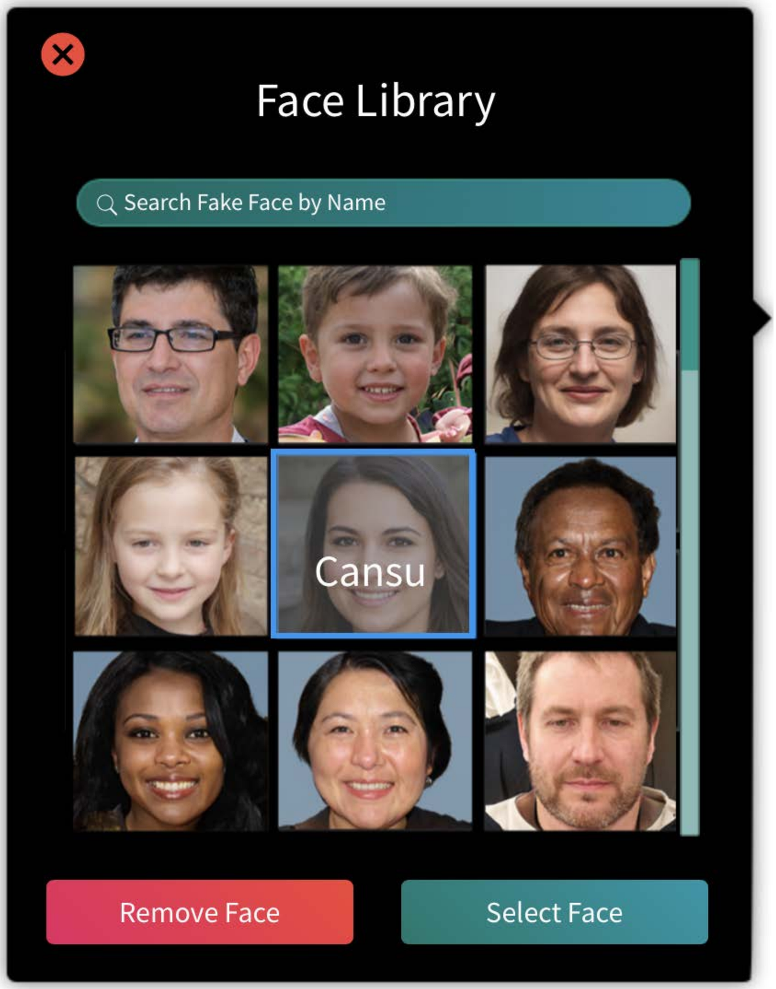


Fig. 29: The Face Library Pop-up Design for the desktop application

If the user clicks the Open Face Library button in the Fake Face Editing pop-up, the face library with all the saved fake faces will appear. In the Face Library pop-up, the user can search for a specific face by their saved name by typing the name into the search bar or simply scroll through the faces. When a user clicks on a face, the name of the face will appear on screen. The user can remove the selected face from the face library by clicking Remove Face button, select the face for replacement by clicking Select Face button or close the pop-up and not select any face from the face library by clicking the x button.

3.5.5.2.10 Final Image Save Screen

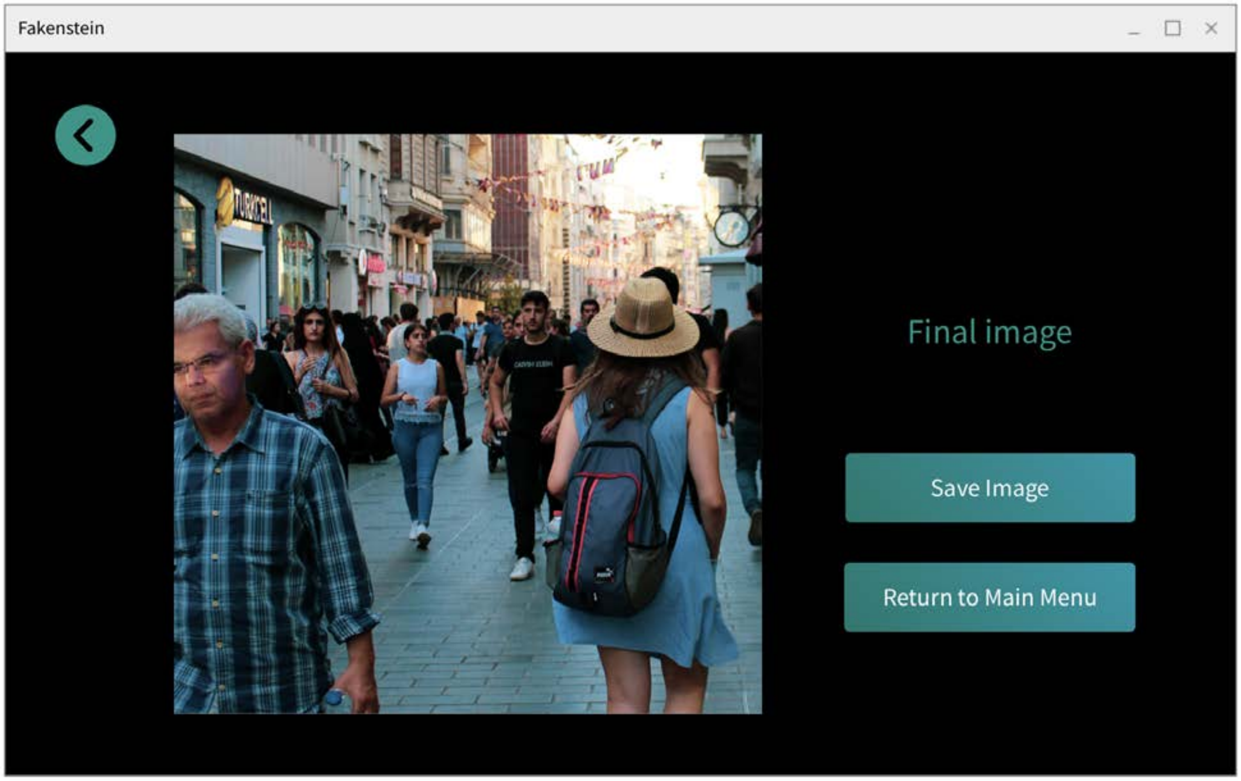
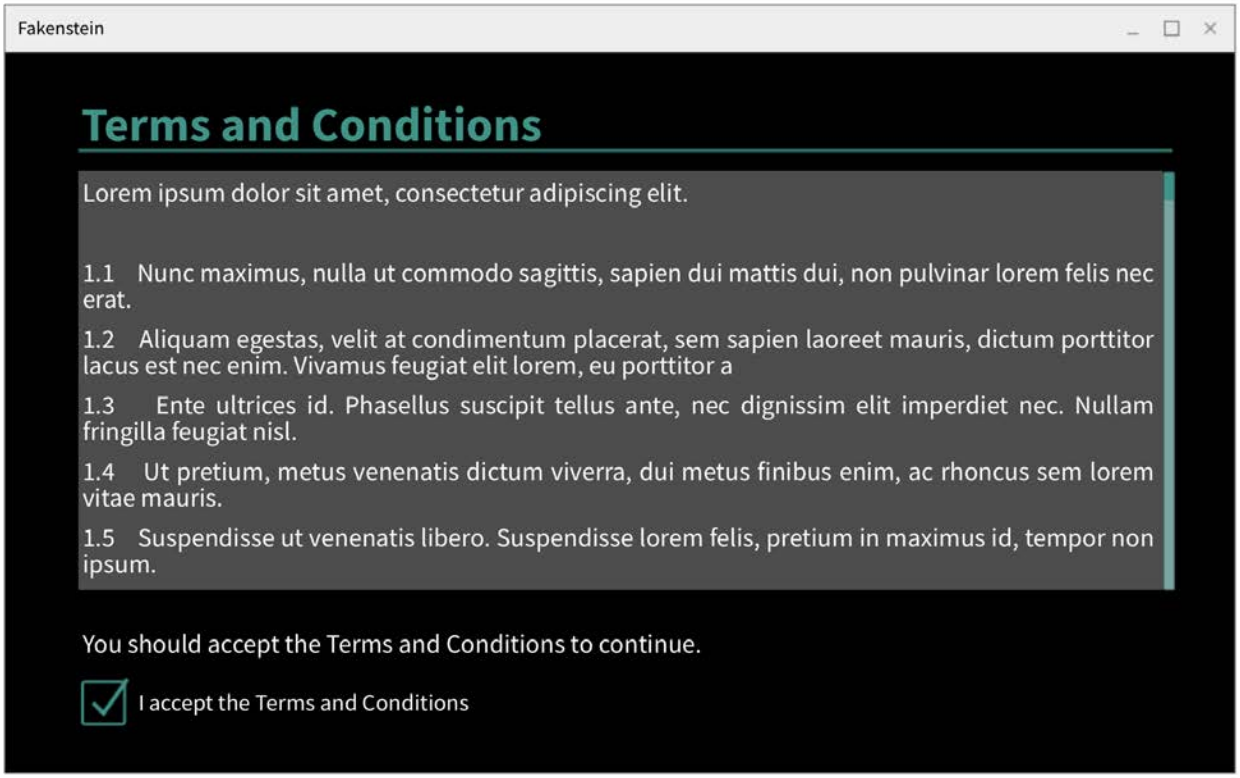


Fig. 30: The Final Image Save Screen Design for the desktop application

After the user clicks the Done button in the Fake Face Editing screen, the image is finalized and ready to be saved. The user can save the image to their device by clicking Save Image button or return to the Main Menu without saving by clicking Return to Main Menu button or go back to editing the image by clicking the Return icon. It should be noted that no user images are stored by the application itself. Therefore, if the user chooses to return to the main menu without saving, the image will be lost.

3.5.5.2.11 Terms and Conditions Screen



Terms and Conditions

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

1.1 Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat.

1.2 Aliquam egestas, velit at condimentum placerat, sem sapien laoreet mauris, dictum porttitor lacus est nec enim. Vivamus feugiat elit lorem, eu porttitor a

1.3 Ente ultrices id. Phasellus suscipit tellus ante, nec dignissim elit imperdiet nec. Nullam fringilla feugiat nisl.

1.4 Ut pretium, metus venenatis dictum viverra, dui metus finibus enim, ac rhoncus sem lorem vitae mauris.

1.5 Suspendisse ut venenatis libero. Suspendisse lorem felis, pretium in maximus id, tempor non ipsum.

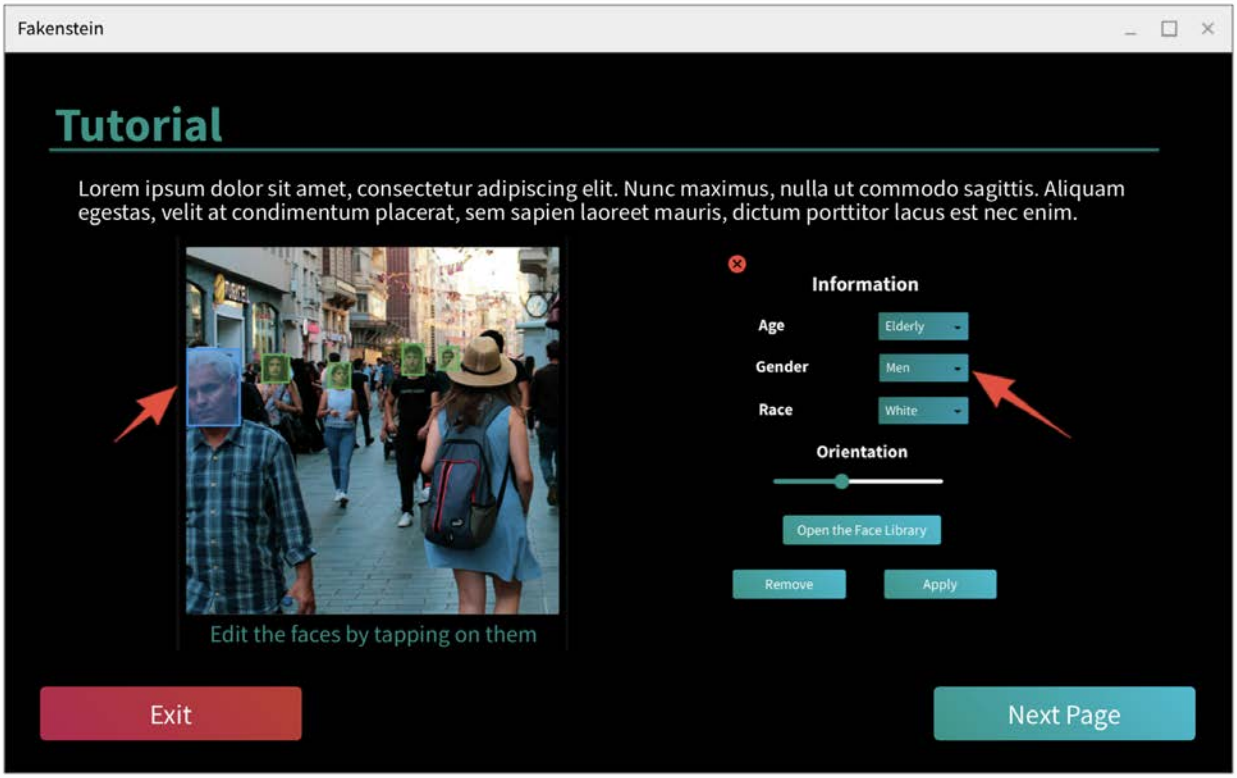
You should accept the Terms and Conditions to continue.

☑ I accept the Terms and Conditions

Fig. 31: The Terms and Conditions Screen Design for the desktop application

Terms and Conditions screen is the first screen that is displayed when the application is opened for the first time. The user cannot proceed using the application before accepting Terms and Conditions. Once the Terms and Conditions are accepted in a device, this screen won’t be displayed again.

3.5.5.2.12 Tutorial Screen



Tutorial

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis. Aliquam egestas, velit at condimentum placerat, sem sapien laoreet mauris, dictum porttitor lacus est nec enim.

Edit the faces by tapping on them

Information

Age: Elderly

Gender: Men

Race: White

Fig. 32: The Tutorial Screen Design for the desktop application

Tutorial screen displays a tutorial about the application and how to use it. The user can go to the next page of the tutorial by clicking the Next Page button. The user can exit from the Tutorial screen and return back to the Main Menu by clicking the Exit button.

4 Other Analysis Elements

4.1 Consideration of Various Factors in Engineering Design

4.1.1 Public Health Factors

Fakenstein’s contains no features that might affect or correlate to public health in any shape or form.

4.1.2 Public Safety Factors

Fakenstein will need access to the user’s media and also permissions to modify such media. Collection and safety of the data are major issues therefore Fakenstein will not store any of the user’s media to an external server and database. Thus, personal information safety will not be infringed and be protected by performing the operations on the user’s local device.

4.1.3 Global Factors

Fakenstein’s contains no features that might have a global impact.

4.1.4 Cultural Factors

There are many cultures that hold every bit of their privacy as sacred due to malpractices in their history. Fakenstein can help restore that privacy as appearing in the photographs that others post online is a possible way to determine where someone was at a particular time. There are also cultures where drawings or photography of individuals are frowned upon or even straight up forbidden. Taking a picture that includes other individuals in the background can trespass on such cultural values without the individuals’ knowledge [12, 13].

4.1.5 Social Factors

Protection of privacy is getting harder by the day in our society as social media grows bigger and bigger. Providing a way for people to remove the background people in their photographs allows people to reclaim their daily privacy. Also, there is a recent trend of people (especially celebrities) removing or censoring the face of their children on social media is growing more and more popular. This comes from the fact that introducing parts of people’s personal lives takes their ability to participate in the society as a regular individual and invites strangers into intimate moments of their lives. Fakenstein is an application that can help reinstate such social barriers that permit control over individuality.

4.1.6 Environmental Factors

Fakenstein’s contains no features that might affect or correlate to environmental factors other than the amount of electricity used by your phone or computer while the application is running.

4.1.7 Economic Factors

Posting photos of people without their permission infringes on their rights of privacy (namely Turkish Personal Data Protection Law No. 6698 [1] and European “Data Protection and online privacy” Law [2]) and therefore allows for possible lawsuits. Through the use of Fakenstein, such personal privacy infringements can be thwarted to avoid penalty fines. Furthermore, the application eliminates the need to use costly photoshopping programs, therefore saving money to application’s users on that front. The basic application for the mobile phone will be free so that everyone can use it. The professional system for the desktop will require a paid subscription for 1 dollar a month which is insignificant next to Adobe Photoshop which is 21 dollars a month [14].

Table 1: The table of design factors, their effect levels and their effects in summary.

Factor	Effect Level	Effect
Public Health Factors	0	Fakenstein’s contains no features that might affect or correlate to public health in any shape or form.
Public Safety Factors	10	Fakenstein will not store any user data that has been accessed or modified
Global Factors	0	Fakenstein’s contains no features that might have a global impact
Cultural Factors	4	Fakenstein allows removal of figures of individuals which can be utilized to fit certain cultural niches.
Social Factors	8	Fakenstein bolsters protection of privacy in the society
Environmental Factors	1	Fakenstein’s contains no features that might affect or correlate to environmental factors other than the amount of electricity used by your phone or computer while the application is running.
Economic Factors	5	The mobile application will be free and the desktop application will require a subscription that costs 1 dollars a month.

4.2 Risks and Alternatives

Fakenstein has three different risks that are related to the hardware environment. Firstly, the response time may be high if the hardware is inadequate to support the application efficiently. It is recommended that users install the application with the recommended software taken into account to work around this issue. Not having the recommended software could lead to lags, or extended wait times. Another potential problem is that there may be a lack of libraries, frameworks, algorithms or research that is essential for the complex functionalities that were determined. The supervisor will be consulted to determine the solution to this matter if the occasion arises. Lastly, there is some concern that the application will be used for immoral purposes, such as forging fake evidence or blaming an offense on someone else. This issue is aimed to be tackled with presenting a Terms and Conditions page in the beginning of the application to ensure that this application should not be used for any immoral purposes and manipulation of pictures but only be used to protect the privacy of other people.

Table 2: The table of expected risks, their likelihoods, their effects on the project, and the b plan summaries.

	Likelihood	**Effect on the Project**	**B Plan Summary**
Risk 1 - Hardware inadequacy	High	High response time	Constant software updates
Risk 2 - Lack of implementation libraries	Low	Increasing the implementation time	Supervisor will be consulted
Risk 3 - Unappropriate usage	High	Legal issues	Terms and Conditions

4.3 Project Plan

Table 3: The table of work packages, the title leaders and members involved in these work packages.

WP	**Work Package**	**Title Leader**	**Members Involved**
WP1	Web Page	Ardahan Doğru	Atakan Dönmez, Öykü Hatipoğlu
WP2	Reports	Elif Kurtay	All members
WP3	Face Identification	Öykü Hatipoğlu	Cansu Moran, Ardahan Doğru

WP4	Blurring	Atakan Dönmez	Cansu Moran, Elif Kurtay
WP5	Face Modification	Cansu Moran	Elif Kurtay, Atakan Dönmez
WP6	Desktop (Windows) GUI	Elif Kurtay	Öykü Hatipoğlu, Ardahan Doğru
WP7	Mobile (Android) GUI	Öykü Hatipoğlu	Elif Kurtay, Ardahan Doğru
WP8	Database Integration	Atakan Dönmez	Öykü Hatipoğlu, Cansu Moran
WP9	Testing	Cansu Moran	All Members

Gantt Chart for planned Work Package start and end dates:

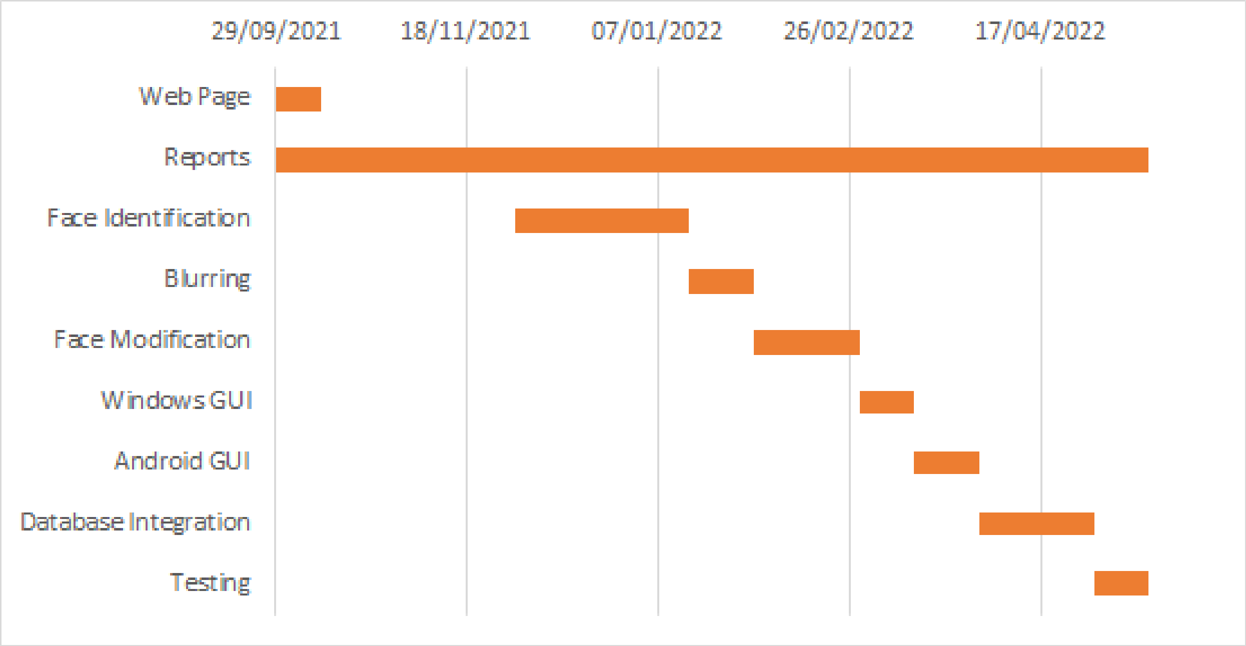


Fig. 33: The Gantt Chart.

Work Package 1: Web Page	
Start Date: 29/09/2021	**End Date:** 11/10/2021
Leader: Ardahan Doğru	**Members Involved:**
Objectives: Create a web page for the project that introduces the team and the project. The web page should also serve as a way to access all reports for the project.	
Tasks **Task 1.1:** Write a short description about the project and create an about page where it is displayed. **Task 1.2**: Create a members page with introductory information for the members of the group **Task 1.3:** Create a page where the reports will be uploaded by the group and always accessible to those that have the link for the web page	
Deliverables **Deliverable 1.1:** About Page **Deliverable 1.2:** Members Page: **Deliverable 1.3:** Reports Page:	

Work Package 2: Reports	
Start Date: 29/09/2021	**End Date:** 15/05/2022
Leader: Elif Kurtay	**Members Involved:** All Members
Objectives: Documenting the steps needed for the project and preparing reports accordingly. The reports also help clear up the details of the project and set up a course of actions to follow.	
Tasks **Task 2.1:** Decide the overall specifications of the project and write a report about it. **Task 2.2**: Analyse the systems that are to be implemented in the project and elements of the strategies to implement them then write a report about it. **Task 2.3:** Design the low-level structure of the project and write a report about it. **Task 2.4:** Design the high-level structure of the project and write a report about it. **Task 2.5:** Finalise the details of the project and write a report about it	
Deliverables **Deliverable 2.1:** Project Specifications Report **Deliverable 2.2:** Analysis Report **Deliverable 2.3:** High-Level Design Report **Deliverable 2.4:** Low-Level Design Report **Deliverable 2.5:** Final Report	

Work Package 3: Face Identification	
Start Date: 01/12/2021	**End Date:** 15/01/2022
Leader: Öykü Hatipoğlu	**Members Involved:** Cansu Moran, Ardahan Doğru
Objectives: Implement an algorithm that identifies faces, classes them as foreground and background and marks the area of the face with a bounding box.	
Tasks **Task 3.1:** Find suitable data to train a model that identifies faces. **Task 3.2:** Implement a model that identifies faces. **Task 3.3**: Implement a model that classes the identified faces based on whether they are on the background or foreground. **Task 3.4:** Mark the area of each face with a bounding box that shows which class it is a part of by the color of the bounding box.	
Deliverables **Deliverable 3.1:** Face identification algorithm **Deliverable 3.2:** Algorithm to show which faces are on the background	

Work Package 4: Blurring	
Start Date: 15/01/2022	**End Date:** 01/02/2022
Leader: Atakan Dönmez	**Members Involved:** Cansu Moran, Elif Kurtay
Objectives: Implementing the feature to allow the users to select parts of the image to blur	
Tasks **Task 4.1:** Implement selecting a circular boundary in the image **Task 4.2**: Implement blurring for a circular area	
Deliverables **Deliverable 4.1:** Blurring functionality	

Work Package 5: Face Modification	
Start Date: 01/02/2022	**End Date:** 01/03/2022
Leader: Cansu Moran	**Members Involved:** Elif Kurtay, Atakan Dönmez
Objectives: Implementing the face modification feature where the user replaces the identified face with that of an artificially generated one.	

Tasks **Task 5.1:** Implement a model that classifies the faces according to their properties such as position, age, gender pose, etc. **Task 5.2**: Implement a model that generates a face according to the given properties such as position, age, gender pose, etc. **Task 5.3:** Implement placing the generated faces on the place of the selected faces. **Task 5.4:** Implement blending for the new faces to not look out of order. **Task 5.5:** Combine the algorithms from the previous tasks so that the outputs feed into one another's inputs.
Deliverables **Deliverable 5.1:** Face Classification **Deliverable 5.2:** Face Generation **Deliverable 5.3:** Face Combination

Work Package 6: Windows GUI	
Start Date: 01/03/2022	**End Date:** 15/03/2022
Leader: Elif Kurtay	**Members Involved:** Öykü Hatipoğlu, Ardahan Doğru
Objectives: Design and implement a desktop app for the functionalities	
Tasks **Task 6.1:** Design the UI for a desktop app **Task 6.2**: Implement the GUI for the desktop app **Task 6.3:** Combine the frontend with the backend **Task 6.4:** Implement temporary placeholders for the features that are not implemented yet.	
Deliverables **Deliverable 6.1:** Desktop app	

Work Package 7: Android GUI	
Start Date: 15/03/2022	**End Date:** 01/04/2022
Leader: Öykü Hatipoğlu	**Members Involved:** Elif Kurtay, Ardahan Doğru
Objectives: Design and implement a mobile Android app for the functionalities	
Tasks **Task 6.1:** Design the UI for a mobile Android app **Task 6.2**: Implement the GUI for the mobile Android app **Task 6.3:** Combine the frontend with the backend **Task 6.4:** Implement temporary placeholders for the features that are not implemented yet.	
Deliverables **Deliverable 7.1:** Mobile Android app	

Work Package 8: Database Integration	
Start Date: 01/04/20212	**End Date:** 01/05/2022
Leader: Atakan Dönmez	**Members Involved:** Öykü Hatipoğlu, Cansu Moran
Objectives: Create an online database that stores prior generated faces to reduce the waiting times for the users.	
Tasks **Task 8.1:** Generate a large set of faces. **Task 8.2**: Create an online database. **Task 8.3:** Combine the database with the desktop and mobile applications. **Task 8.4:** Use generated faces in the classification process for optimization.	
Deliverables **Deliverable 8.1:** Database with prior generated faces **Deliverable 8.2:** Optimized classification process	

Work Package 9: Testing	
Start Date: 01/05/2022	**End Date:** 15/05/2022
Leader: Cansu Moran	**Members Involved:** All Members
Objectives: Implement tests that inspect model outputs, connections between different models, UI functionalities and application response to invalid/unexpected input.	
Tasks **Task 9.1:** Write tests to inspect model output accuracy **Task 9.2**: Write tests to inspect how models utilize other models' output as input. **Task 9.3:** Write test for UI functions **Task 9.4:** Write time/efficiency tests **Task 9.5:** Write tests to inspect invalid/unexpected input	
Deliverables **Deliverable 9.1:** Testing results **Deliverable 9.2:** Tools for testing future features.	

4.4. Ensuring Proper Teamwork

In order to ensure proper teamwork our team uses synchronous and asynchronous communication tools. Weekly Zoom calls and face to face meetings are held where job distribution and progress tracking is done. Decisions made during the meetings are noted. Then through WhatsApp and emails all members are reminded of said decisions (or informed if they were unable to attend the meeting) and also sent a calendar invitation and a link to the next meeting. After the meetings the members are split up and paired to accustom the difficulty of tasks at hand. Google Docs is used as a collaborative tool for simultaneous documentation and report preparation. GitHub will be utilised to implement parts of the project concurrently after which merges and conflict resolutions will be solved in the weekly meetings. The project requires multiple new technologies for the members to learn such as web-development, Deep Learning, Android Studio, etc. therefore ensuring each member is able to focus on specific subjects rather than having to tend to every task is crucial for a good end-product and the aforementioned methods allow such an environment.

4.5. Ethics and Professional Responsibilities

Our background research has revealed that some of the previous Deep Learning implementations that also deal with identifying faces and some of its properties have suffered from biases due to different amounts of data for each class [15]. Therefore, we are responsible to make sure that our dataset is not tweaked or skewed for particular characteristics to be able to train our models properly with equal amounts of data for classes (gender, race, age as well as pose properties). Moreover, we will be utilizing datasets, libraries and implementation techniques from outer sources so we must make sure that any help we get from outside sources are properly cited and that we do not violate any licensing for them. Furthermore, the program will be accessing the user's gallery and modifying images therefore we have to request appropriate permissions and make sure to only make changes in the user’s local device and not store their information on an outside storage. Another professional responsibility we have is to make sure the application covers the users’ needs even in cases where the Deep Learning models are insufficient. Thus, we will be adding a manual blurring option for the cases where the user wants to hide parts of the image that are not faces or faces that are too distorted in the image that it does not register as one in the identification model. A Terms and Conditions agreement will have to be accepted in order to use this application. In the agreement, the users will have to accept not to use this application for manipulating photos for unethical purposes.

4.6. Planning for New Knowledge and Learning Strategies

The implementation of our project requires us to learn many subjects and technologies that we have not previously used before, therefore it is crucial that we develop learning strategies. Some of the new technologies we need to learn how to operate are:

- Issue tracking and agile project management
- Application design
- Android application development
- Windows application development
- Database design
- Database management
- Image processing
- Deep Learning
- Optimization
- Testing

Issue and project tracking will be done through GitHub and Jira. We are familiar with GitHub from previous projects and with Jira from internships or research, however skills for both will need to be improved through hands-on experience. Application design and application development are concepts we have experience with but still we will take a udemy course [8] in order to learn the proper development techniques and principles on ReactJS. The database design and management parts will be mostly handled by members who have previously taken a related course but materials from the CS353 Database Systems will be referred to for problems. Image processing, optimization and testing are concepts the members of our group are familiar to some extent but still lacking therefore online learning will be of great use for these. Four members of our group are taking the CS464 Introduction to Deep Learning course as Deep Learning makes up most of our work in the project. Concepts such as Deep Learning Models, Classification, Generative Adversarial Networks will be learned from the course and also literature review will be done for the specific features that we want to implement for our project. Even if we devise plans and strategies to gain new knowledge, it is in our experience that unknown problems arise due to not having enough experience with the concepts we are working with. Thus, we aim to start the processes of learning and following implementations as early as we can in order to give leeway for any possible setbacks. We have also planned a divide and conquer approach in our project plan so that not every member has to learn every aspect of the project, but rather teams of two or three focusing on specific parts and moving onto the next unfinished parts to achieve a more efficient development schedule.

5. References

[1] R. Eveleth, “How Many Photographs of you are out there in the world?” ,” *The Atlantic*, 03-Nov-2015. [Online]. Available: https://www.theatlantic.com/technology/archive/2015/11/how-many-photographs-of-you-are-out-there-in-the-world/413389/. [Accessed: 15-Nov-2021].

[2] “Contributed street view imagery policy,” Google. [Online]. Available: https://www.google.com/intl/en_uk/streetview/policy/. [Accessed: 09-Oct-2021].

[3] “Photoshop free trial | official adobe photoshop.” [Online]. Available: https://www.adobe.com/products/photoshop/free-trial-download.html. [Accessed: 9-Oct-2021].

[4] E. Then, “Pixel 6 magic eraser removes uninvited people from photos,” *SlashGear*, 20-Oct-2021. [Online]. Available: https://www.slashgear.com/pixel-6-magic-eraser-removes-uninvited-people-from-photos-19695941/. [Accessed: 15-Nov-2021].

[5] “Remove people from photo: The easy way,” *Inpaint*. [Online]. Available: https://theinpaint.com/tutorials/pc/how-to-remove-unwanted-people-from-photo. [Accessed: 09-Oct-2021].

[6] ”This person does not exist,” *This Person Does Not Exist*. [Online]. Available: https://thispersondoesnotexist.com/. [Accessed: 09-Oct-2021].

[7] D. Lumb, “Google Pixel 6 review,” *TechRadar*, 05-Nov-2021. [Online]. Available: https://www.techradar.com/reviews/google-pixel-6. [Accessed: 14-Nov-2021].

[8] “Unique, worry-free model photos,” *Generated Photos*, 2021. [Online]. Available: https://generated.photos/#. [Accessed: 14-Nov-2021].

[9] “Turkish Personal Data Protection Law No. 6698 • Kişisel Verilerin korunması mevzuatı • KVKP,” *Kişisel Verilerin Korunması Platformu (KVKP)*. [Online]. Available: https://www.kisiselverilerinkorunmasi.org/kanunu-ingilizce-ceviri/. [Accessed: 10-Oct-2021].

[10] “Data Protection and online privacy,” *Your Europe*. [Online]. Available: https://europa.eu/youreurope/citizens/consumers/internet-telecoms/data-protection-online-privacy/index_en.htm. [Accessed: 11-Oct-2021].

[11] E. by F. Jerga and F. Jerga, “REACT JS & Firebase Complete Course (incl.. chat application),” *Udemy*. [Online]. Available: https://www.udemy.com/course/react-js-firebase-complete-course-incl-chat-application/. [Accessed: 15-Nov-2021].

[12] R. Farkas, “Who cares about privacy? surprising facts from around the Globe,” *Argonaut*, 2015. [Online]. Available: https://www.argonautonline.com/blog/attitudes-to-privacy-surprising-global-facts/. [Accessed: 15-Nov-2021].

[13] S. Karolius, “Perception of privacy – different cultures and different approaches,” *Behavioral Development Economics*, 10-Jul-2017. [Online]. Available: https://behavioraldevelopmentblog.wordpress.com/2017/07/10/perception-of-privacy-different-cultures-and-different-approaches/. [Accessed: 15-Nov-2021].

[14] “Adobe Creative Cloud Plans, pricing, and membership.” [Online]. Available: https://www.adobe.com/creativecloud/plans.html. [Accessed: 15-Nov-2021].

[15] A. Palmer, “Study finds popular face ID systems may have racial bias,” *Daily Mail Online*, 12-Feb-2018. [Online]. Available: https://www.dailymail.co.uk/sciencetech/article-5382979/Study-finds-popular-face-ID-systems-racial-bias.html. [Accessed: 15-Nov-2021].