**Team 32 Software Requirements Specification Document**

**Members:**

1) Yashas S. B. - 2022113001

2) Saideekshith Vaddineni - 2022101110

3) Morampudi Srisai Krishna - 2022101115

4) Saketh Reddy Vemula - 2022114014

5) Ishita Bansal - 2022114004

# Brief problem statement

The project involves developing an application for automatically correcting the key features of a human face in photos. This involves selecting the best eyes and smile from all the clusters of the faces and stitch them into the group photo. The application is expected to take a group image and faces clusters corresponding to those images and correct the closed eyes and make the face smile. At the end we are expected to correct all the necessary faces present in the group image.

# System requirements

1. **Data Preprocessing:** Implement data preprocessing techniques to prepare the input images for analysis.
2. **Face Detection and Key Points Detection:** Use various state-of-the-art methods for face recognition and Face landmarks detection.
3. **Expression Analysis:** Develop algorithms to identify best eyes and smiles given a cluster of images.
4. **Image Stitching:** Develop a seamless position aware algorithm for stitching the patches.
5. **Blending:** Develop an algorithm which can take care of colours, edges for seamless clone onto the base image of patches.
6. **GANs or ML Integration (Bonus):** Explore GANs based approach and try to come up with a prototype method.
7. **Technologies and Libraries:** Requires technologies which can solve the problem efficiently. Suggested are state-of-the-art pretrained neural network models such as RetinaFace Face recognition and MediaPipe Landmark detection.

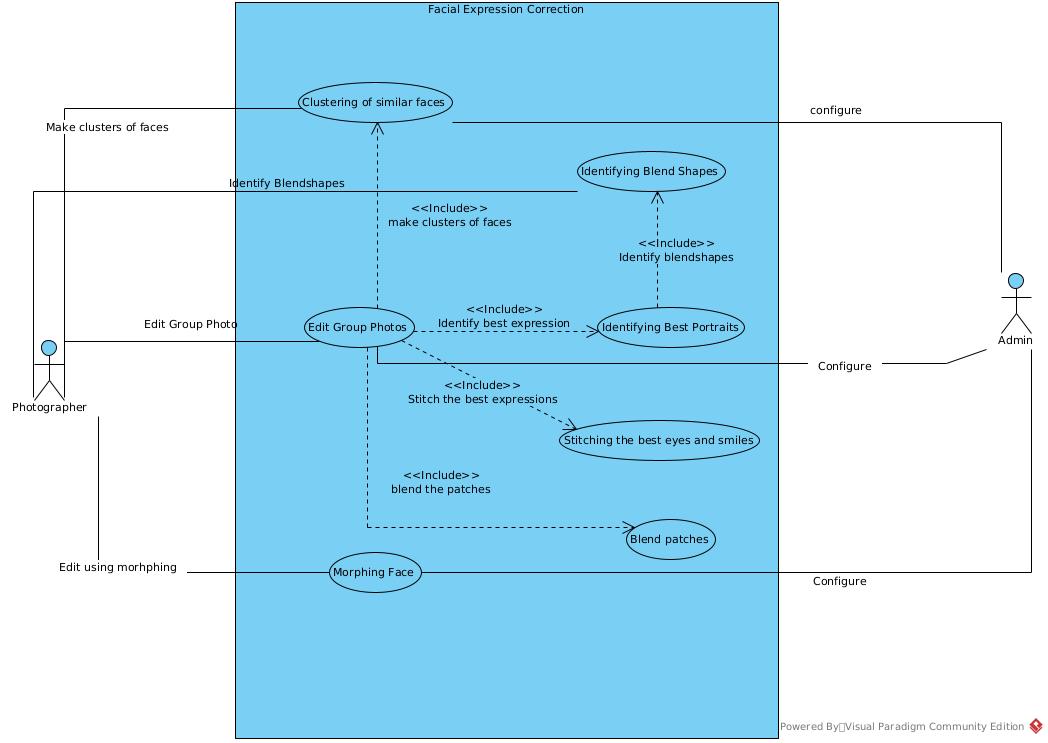
# Users profile

Although it is intended for general usage as well, professional wedding photographers will be the primary users of this application. In order to guarantee they catch every priceless moment, wedding photographers frequently take a ton of pictures. The purpose of this app is to allow photographers to edit the facial expressions in order to not loose on these precious moments. Specifically the application is expected to open the closed eyes and make the faces smile. The application is expected to be user-friendly. Through this application, photographers can edit the eyes and smile efficiently using their own datasets.

# Feature Requirements (described using use cases)

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **User Case Name** | **Description** | **Release** |
|  | Clustering of similar faces | Given some photos with containing a person, the software makes cluster of these photos containing the same orientation. | R1 |
|  | Identifying the best portrait | Many photographers take a burst of photos to capture the moment. Our software can pick the best image from the burst based on various metrics. | R1 |
|  | Identification of blend shapes | Our software can identify the blend shapes of a face in a photo. | R1 |
|  | Morphing the face | Our software can morph the selected face to incorporate the best smile of that person onto the face. | R1 |
|  | Stitching the best eyes and smile | Given a targets (which contains correct eyes and smiles) and base images, the application corrects facial expression in | R2 |
|  | Edit group photos | Our software not only edits portraits, it also edits group photos and generate the best possible one by using various predefined heuristics and stitching the best key features. | R2 |
| 7. | Blend the patches | Application blends the edges and adjust the colors of the patch to match the base image. Seamless clone is required for realistic results. | R2 |
| 8. | Direct Expression Correction with GANs(Generative Adversarial Networks) | Our application can use GANs to directly correct smile, eyes and face structure of all faces in a photo, without intermediate steps of stitching and best features identification. | R2 |

**Use case diagram**



**Use case description**

|  |  |
| --- | --- |
| **Use Case Number:** | **01** |
| **Use Case Name:** | Clustering of similar faces­­­­­ |
| **Overview:** | Given some photos with containing a person, the application makes cluster of these photos containing the same orientation. |
| **Actors:** | User: Initiates the process by providing all the photos of a single person |
| **Pre condition:** | Photos with the same person are given as input. |
| **Flow:** | 1. User provides all the photos of the same person he/she has. 2. The application will make clusters out of all photos in which each cluster has photos with the faces have similar orientation. 3. Then the user can view these clusters. |
|  | Alternate Flows: None |
| **Post Condition:** | The application has successfully clustered photos into faces with similar orientation. |

|  |  |
| --- | --- |
| **Use Case Number:** | **02** |
| **Use Case Name:** | Identifying the best portrait |
| **Overview:** | Many photographers take a burst of photos to capture the moment. Our application can pick the best image from the burst based on various metrics. |
| **Actors:** | User: Initiates the process by providing the burst of photos or interacting with the software interface. |
| **Pre condition:** | The entire burst of photos is accessible. |
| **Flow:** | 1. User provides a burst of photos to the software. 2. The application analyzes each photo in the burst based on predefined metrics such as focus quality, facial expressions, and composition. 3. The application selects the photo with the highest overall quality as the best portrait and this selected portrait is presented to the user. Software doesn’t use ML. |
|  | Alternate Flows: None |
| **Post Condition:** | The software has successfully identified and presented the best portrait from the provided burst of photos. |

|  |  |
| --- | --- |
| **Use Case Number:** | **03** |
| **Use Case Name:** | Identification of blendshapes |
| **Overview:** | Our software can identify the blendshapes of a face in a photo. |
| **Actors:** | User: Initiates the process by providing all the photos he/she has. |
| **Pre condition:** | The entire collection of photos is accessible. |
| **Flow:** | 1. User provides all the photos he/she has. 2. The software analyzes the photos and gets the blendshapes of the face in the photo. 3. Based on the blendshapes, the software can identify the quality of smile and eyes features in each photo. |
|  | Alternate Flows: None |
| **Post Condition:** | The application has successfully identified blendshapes in each photo.. |

|  |  |
| --- | --- |
| **Use Case Number:** | **04** |
| **Use Case Name:** | Morphing the face |
| **Overview:** | Our software can morph the selected face to incorporate the best smile of that person onto the face. |
| **Actors:** | User: Initiates the process by selecting the photo he/she wants to morph |
| **Pre condition:** | The entire collection of photos is accessible. |
| **Flow:** | 1. After the software has identified the best smile of a person, it morphs the selected face to incorporate this best smile. 2. Give this new modified photo to the user. 3. Will use a separate software for face morphing. |
|  | Alternate Flows: None |
| **Post Condition:** | The morphing of face incorporating best smile should be done. |

|  |  |
| --- | --- |
| **Use Case Number:** | **05** |
| **Use Case Name:** | Stitching the best eyes and smile |
| **Overview:** | Our software can stitch the best eyes and smile of a person from all the photos onto the selected face of the person. |
| **Actors:** | User: Specifies which photo to fix. |
| **Pre condition:** | The software must have identified the best eyes and smile. |
| **Flow:** | 1. After the software has identified the best eyes and smile of a person, it should stitch them onto the photo that has been asked to fix. 2. Give this new modified photo to the user. 3. Will use a separate software to stitch eyes and smile onto image. |
|  | Alternate Flows:  1. Software identifies best possible **set** of features and user chooses from this set.  2. The software modifies the base image and returns stitched photo. |
| **Post Condition:** | The stitching of best eyes and smile should be done. |

|  |  |
| --- | --- |
| **Use Case Number:** | **06** |
| **Use Case Name:** | Edit group photos |
| **Overview:** | Our software not only edits portraits, it also edits group photos and generate the best possible one by using various predefined heuristics and stitching the best key features. |
| **Actors:** | User: Provides photos to the software. |
| **Pre condition:** | The photos must have multiple faces for the analysis |
| **Flow:** | 1. User must provide photos to the software. 2. The software will analyze group photos and find best key features based on some heuristics and stitch them accordingly. 3. These modified photos will be given to the user. |
|  | Alternate Flows: None |
| **Post Condition:** | Photos involving some key people will be clustered under their identity. Best features of these people will also be found with analysis. |

|  |  |
| --- | --- |
| **Use Case Number:** | **07** |
| **Use Case Name:** | Blend the patches |
| **Overview:** | Application blends the edges and adjust the colors of the patch to match the base image. Seamless clone is required for realistic results. |
| **Actors:** | Photographer: Specifies the area for blending |
| **Pre condition:** | The application must complete the patching/stitching in that area |
| **Flow:** | 1. After, patching is done, the application processes the area of the patch 2. Sharp edges are smoothened out. 3. Colors of the patches are adjusted to have a seamless clone with the colors/gradient of the base image. |
|  | Alternate Flows:  1. Return to Stitching/patching on the faces. |
| **Post Condition:** | Return the seamlessly blended faces. |

|  |  |
| --- | --- |
| **Use Case Number:** | **08** |
| **Use Case Name:** | Direct Expression Correction with GANs(Generative Adversarial Networks) |
| **Overview:** | Our application can use GANs to directly correct smile, eyes and face structure of all faces in a photo, without intermediate steps of stitching and best features identification. |
| **Actors:** | User: Specifies which portrait to fix. |
| **Pre condition:** | The application has got pretrained models for testing. |
| **Flow:** | 1. Take the portrait as input and apply the pretrained model to create an edited mesh 2. Give this new modified photo to the user. 3. Edited portrait is returned to the user. |
|  | Alternate Flows:  1. Software outputs the best possible edited face and shows the user for satisfaction.  2. The application shifts to Stitching and Blending methods. |
| **Post Condition:** | Edited portrait corresponding to the input image is returned to the user. |