**Detailed Requirements**

**Functional Requirements:**

1. **Upload or Take Photo of Eyes:**

* Allow users to upload an image or take a photo through the app.
* Validate image quality (e.g., clear, centered, proper lighting).
* Provide feedback and retry options for poor-quality photos.

1. **Getting Recommendation for Silicone Shield:**

* Analyze uploaded photo to determine eye shape and lash growth angle.
* Recommend a silicone shield with a brief explanation.
* Allow users to override or customize the recommendation.

1. **Acquire Knowledge on Lash Lift Field:**

* Provide educational content about lash lift techniques and tools.
* Include tutorials, FAQs, and step-by-step guides for users.
* Enable users to explore different silicone shields and their uses.

**Pre-Conditions:**

1. **User Access:**

* The user has access to the app and has registered or logged in.

1. **Image Submission:**

* The user has a functional camera or a pre-captured photo of the eyes ready for upload.

1. **Device Requirements:**

* The user’s device supports the app (e.g., compatible OS, sufficient storage, working internet connection).

1. **Lighting and Image Quality:**

* The photo must meet minimum quality standards, including clear visibility of the eyes and lashes.

1. **Database Availability:**

* The system has access to a database of silicone shield types and their suitability based on eye shapes and lash growth angles.

**Post-Conditions:**

1. **Recommendation Provided**:

* The system successfully analyzes the photo and provides a recommended silicone shield type for the lash lift.

1. **Error Feedback:**

* If the system encounters issues (e.g., poor image quality or analysis failure), the user receives actionable feedback for resolution.

1. **User Data Stored**:

* If necessary, the system securely stores user preferences, selected shield types, or uploaded images for future reference or analytics.

**Use Cases:**

**Use Case Name**: Getting recommendation for the best fitting silicone shield

**Participating Actors:** Beautician, Beautician's client, App, Database

**Flow of Events:**

1. The user authenticates
2. The system displays a list of user operations
3. The user selects the ‘take photo' option
4. The user takes photo of the client's eyes
5. The system takes the user’s photo and analyzing the image
6. The system presents the best fitting silicone shield for the client

**Alternative flow A:**

**A.1** At step 5 the system notifies the user that the photo is blurred, or the lighting is not sufficient.

**A.2** The user takes the photo again.

**Alternative flow B:**

**B.1** At step 5 the system informs the user about the difficulty in determining the eye shape.

**B.2** The user is given instructions to submit a new photo with specific guidance on capturing a better angle.

**B.3** The user takes the photo again.

**Alternative flow C:**

**C.1** At step 6 the system identifies two equally suitable options based on the analysis.

**C.2** The system presents multiple shield types with a brief explanation for each choice.

**C.3** User selects the one they prefer.

**Non-Functional Requirements:**

1. **Performance**

* The system should process images and provide recommendations within 5-10 seconds.

1. **Usability**

* The user interface must be intuitive, with clear instructions for uploading photos and viewing recommendations.

1. **Security**

* User data, including uploaded photos, must be encrypted and stored securely.

1. **Compatibility**

* The app should be compatible with iOS and Android devices.

1. **Availability**

* Ensure 99.9% uptime, with scheduled maintenance windows communicated in advance.

1. **Maintainability**

* The codebase should be modular and documented for ease of updates and bug fixes.
* Logs should track errors and user activity for troubleshooting.

1. **Image Quality**

* The system should support image resolutions between 720p and 4K for accurate analysis.

1. **Reliability**

* The app should handle errors gracefully, providing clear feedback and recovery options without crashes.