## Face Detection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** |  | | | |
| **Use Case Name:** | Detect Face | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | User/ App tester | | |
| **Description:** | |  | | |
| **Trigger:** | | When the application is started | | |
| **Preconditions:** | | 1. The user has integrated the UI/UX library with the application 2. The application has started 3. The camera is in working state | | |
| **Post conditions:** | | 1. The face of the user would be extracted through frames generated by the camera. | | |
| **Normal Flow:** | | 1. The user starts the application 2. The camera starts in the backend 3. The camera starts generating frames 4. The frames are sent to open cv port to detect face 5. The face is detected and a rectangle box is placed around it | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | |  | | |
| **Exceptions:** | | 1. The camera fails to start 2. The library doesn’t load | | |
| **Includes:** | | 1. Android cell phone with front camera | | |
| **Frequency of Use:** | |  | | |
| **Special Requirements:** | |  | | |
| **Assumptions:** | |  | | |
| **Notes and Issues:** | |  | | |

## Eyes Detection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** |  | | | |
| **Use Case Name:** | Detect Eyes | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | User/ App tester | | |
| **Description:** | |  | | |
| **Trigger:** | | When the library has detected the face | | |
| **Preconditions:** | | 1. The user has integrated the UI/UX library with the application 2. The application has started 3. The camera is in working state 4. The face is detected by the library | | |
| **Post conditions:** | | 1. The eyes of the user would be detected. | | |
| **Normal Flow:** | | 1. The user starts the application 2. The camera starts in the backend 3. The camera starts generating frames 4. The frames are sent to open cv port to detect face 5. The face is detected and a rectangle box is placed around it 6. The face is taken as input by the library and then in return it detects eyes from it. | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | |  | | |
| **Exceptions:** | | 1. The camera malfunctions 2. The library stop working | | |
| **Includes:** | | 1. Android cell phone with front camera | | |
| **Frequency of Use:** | |  | | |
| **Special Requirements:** | |  | | |
| **Assumptions:** | |  | | |
| **Notes and Issues:** | |  | | |

## Pupil Detection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** |  | | | |
| **Use Case Name:** | Detect pupil | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | User/ App tester | | |
| **Description:** | |  | | |
| **Trigger:** | | When the library has detected the eyes | | |
| **Preconditions:** | | 1. The user has integrated the UI/UX library with the application 2. The application has started 3. The camera is in working state 4. The face is detected by the library 5. The eyes are detected by the library | | |
| **Post conditions:** | | 1. The pupil of the user would be detected. | | |
| **Normal Flow:** | | 1. The user starts the application 2. The camera starts in the backend 3. The camera starts generating frames 4. The frames are sent to open cv port to detect face 5. The face is detected and a rectangle box is placed around it 6. The face is taken as input to open CV port which in return gives us location of eyes. 7. The eyes are then taken as input to open CV port which in return gives us location of pupil | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | |  | | |
| **Exceptions:** | | 1. The camera malfunctions 2. The library stop working | | |
| **Includes:** | | 1. Android cell phone with front camera | | |
| **Frequency of Use:** | |  | | |
| **Special Requirements:** | |  | | |
| **Assumptions:** | |  | | |
| **Notes and Issues:** | |  | | |

## Corner Detection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** |  | | | |
| **Use Case Name:** | Detect pupil | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | User/ App tester | | |
| **Description:** | |  | | |
| **Trigger:** | | When the library has detected the eyes | | |
| **Preconditions:** | | 1. The user has integrated the UI/UX library with the application 2. The application has started 3. The camera is in working state 4. The face is detected by the library 5. The eyes are detected by the library 6. The pupil is detected by the library | | |
| **Post conditions:** | | 1. The pupil of the user would be detected. | | |
| **Normal Flow:** | | 1. The user starts the application 2. The camera starts in the backend 3. The camera starts generating frames 4. The frames are sent to open cv port to detect face 5. The face is detected and a rectangle box is placed around it 6. The face is taken as input to open CV port which in return gives us location of eyes. 7. The eyes are then taken as input to open CV port which in return gives us location of pupil 8. The pupil of the person is given as input to the open cv port and in return it’s give the corners. | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | |  | | |
| **Exceptions:** | | 1. The camera malfunctions 2. The library stop working | | |
| **Includes:** | | 1. Android cell phone with front camera | | |
| **Frequency of Use:** | |  | | |
| **Special Requirements:** | |  | | |
| **Assumptions:** | |  | | |
| **Notes and Issues:** | |  | | |

## Heat Map Generation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** |  | | | |
| **Use Case Name:** | Detect pupil | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | User/ App tester | | |
| **Description:** | |  | | |
| **Trigger:** | | When the library has detected the eyes | | |
| **Preconditions:** | | 1. The user has integrated the UI/UX library with the application 2. The application has started 3. The camera is in working state 4. The face is detected by the library 5. The eyes are detected by the library 6. The pupil is detected by the library 7. The corners are detected by the library | | |
| **Post conditions:** | | 1. The pupil of the user would be detected. | | |
| **Normal Flow:** | | 1. The user starts the application 2. The camera starts in the backend 3. The camera starts generating frames 4. The frames are sent to open cv port to detect face 5. The face is detected and a rectangle box is placed around it 6. The face is taken as input to open CV port which in return gives us location of eyes. 7. The eyes are then taken as input to open CV port which in return gives us location of pupil 8. The pupil of the person is given as input to the open CV port and in return it’s give the corners. 9. The corners and computed data later help generate heat maps | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | |  | | |
| **Exceptions:** | | 1. The camera malfunctions 2. The library stop working | | |
| **Includes:** | | 1. Android cell phone with front camera | | |
| **Frequency of Use:** | |  | | |
| **Special Requirements:** | |  | | |
| **Assumptions:** | |  | | |
| **Notes and Issues:** | |  | | |