1. Complete the intake flow. User shall be able to either upload an existing photo image file or take a photo from the camera.

    Call "Find Match" method (as of now provide dummy implementation for Find Match for testing purpose). If match exists the inmate details will be returned by "Find Match" method. Allow the user to enter other details like Admit date and Intake ID. Intake ID should be a varchar(25) field and it’s a mandatory field for the form submission. I did not see Intake Id in your form yesterday; hence I am adding this note.

2. Create a FaceRecognitionEvents table with following columns: a. eventid BIGInt column ( autoIncrement) b. location (varchar(25)) c. eventdatetime (DateTime) d. personrecognized (varchar(50) e. linktomatchReport ( varchar (100)) d. additional\_Info(varhcar(max)) e. IsProcessed ( small int or byte)

3.  Create a table by name Programs with following columns: a. programid ( int, Identity, autoincrement ) b. program name ( varchar (25) c. program\_default\_location varchar(25) d. defaultstarttime ( time in 24 hr format hh:mm:ss) e. defaultendtime ( time in 24 hr format hh:mm:ss)

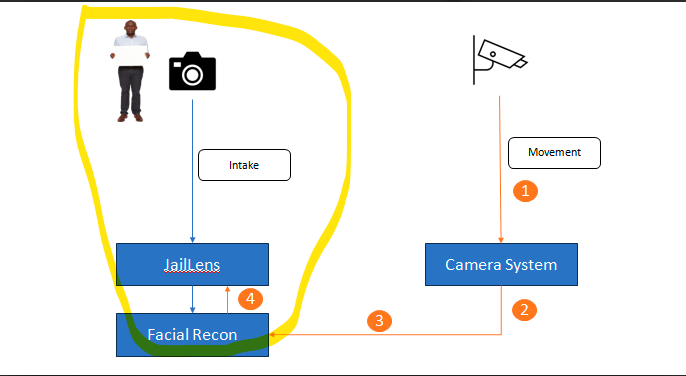
4. Locations: a. locationid ( identity) b. locationname

5. ProgramLocation : a. plid ( identity) b. ProgramId ( FK ) c. LocationId

6. InmateSchedule: a. isid ( identity) b. IntakeId (varchar(25)) c. programId (FK) d. starttime ( time in 24 hr format hh:mm:ss)  e. endtime ( time in 24 hr format hh:mm:ss)

7. JailLensAlerts: a. jlaid ( identity) b. intakeid c. programname d. inmatename e. alertdescription f. alertcategory g. createddate h. comments h. isprocessed

Add sample data to these tables for testing purpose.



Write a windows service that does the below tasks sequentially in a continuous loop (with slight delay as required):

1. Should call a webservice (API) to get the data of face recognition events in last “n” seconds. If API returns some rows then those details will be entered in to FaceRecognition events table. Windows service shall mark event the record as "IsProcessed=0" to indicate that the record is not processed.

[After collecting the data, the time of data collection can be recorded in a variable so as to compute the “n” seconds to get the data of face recognition events next time.]

[ A mock API can be written for the testing purpose returning some Face Recognition events]

1. Should look at all Unprocessed faceRecognition events; Checks if the location is appropriate considering the program schedule of the inmate. If the location is appropriate for the inmate then "IsProcessed" is should be marked as 1. An entry should be made in Program attendance table for the inmate for that particular program / corresponding to a specific time slot. If the location is not appropriate for the inmates as per the schedule then “IsProcessed” should be marked as 2 ( Indicating the alert scenario) and an entry should be written to Alerts table.
2. Should identify all the absentees for various programs corresponding to the current time slot. All absentee list for the program shall be written to the Alerts data.

**Screen for Alerts Display and Processing:**

If there are any Unprocessed alerts in the "Alerts" table then appropriate indication should appear on the screen; so that it will catch attention of the user. On Click of the Alerts Indicator the user should be taken to Alerts screen. The user shall be able to browse through the alerts and should be able to enter some comments against each alert. After entering the comments the user clicks on "Alert Processed" and the status of the Alert should be changed to Processed status.

|  |
| --- |
|  |
|  |