Norman model

1. **Objective**: User wants to take the attendance of students
2. **Intention**: Use automatic attendance system app to take attendance
3. **Formulate** Steps:
    a. Opens the app
    b. Press the "Take Attendance" button
    c. Select the course and attendance source
    d. After all the students have been marked as present, press 'q' to get the attendance details
4. **Execution**: User presses the Take Attendance button. He then selects the course name and attendance source. If camera selected then each student passes in front of the camera to be marked as present. The student name is prompted on the screen to show the successful marking of attendance providing him with the feedback. After all the students have been marked, the user presses the 'q' button to close. On closing, the attendance is shown. If manual attendance is taken then the teacher marks the students attendendance manually.
5. User presses the "Show attendance" button which will display the attendance record of the class
6. User interprets the attendance sheet to see the present and absent students
7. User evaluates that his task of taking attendance has been successfully performed or not

Ergonomics → Study of the physical characteristics of interaction
Ergonomics→ In our project, user interacts directly with the system.
The buttons are arranged in a similar manner

Common interaction style used → menus, WIMP, point and click, form fills, dialogue boxes(For attendance show, for About), buttons, popup menus

Error & repair strategy applied, as the camera might malfunction during working so the user is prompted that the camera might be not working properly

Design tradeoffs: Due to high quality facial recognition constraint, the user feels a little lag in the display.

Design Principles

1. Learnability:

Predictability: The user can easily perceive from the buttons that what button will perform which task. For example, “Take Attendance” button will take the attendance of the students.

Synthesizability: Students will be prompted on attendance mark Show attendance button will display the attendance sheet

Familiarity:User is familiar to the icons and can easily perceive what to do
Consistency:Similar icons perform similar task such as close button will always close the screen

Consistency – likeness in input/output behaviour. Pointer based controls are provided

2. Flexibility:

Substitutivity: We can check the attendance on our app as well as from the generated csv file.

We have provided option to take the attendance using laptop camera, external camera or manually.

We can select the buttons by clicks or by shortcut keys

Task migratability – passing responsibility for task execution between user and system. User starts the attendance procedure and the system starts taking the attendance of students, then to save the attendance, the user takes control and stops the recognition and the system saves the attendance in the attendance sheet

Dialog initiative - User preemptive → User is the one to initiate an action on the system 3. Robustness

Observability → The user can easily perceive the state of the system by the interface

Recoverability – ability of user to take corrective action once an error has been recognized → User can easily recover from error if the camera stops working

Responsiveness → The system is responsive and actions are performed quickly

Task conformance - System allows a user to perform tasks he needs and in an expected way, for example, he wants to take attendance, he can, He wants to see the attendance, he can.

Task completeness: System covers all the tasks that an attendance system shall have

Task adequacy deals with the user's ability to understand the tasks. It is important that a system allows the user to perform any desired task in an application as specified in the system specification prior to the delivery of the system. About section contains details about how to use the attendance app effectively

Shneiderman

1) **Strive for consistency:** Consistency in buttons
2) **Enable frequent users to use shortcuts**: Shortcut keys added
3) **Offer informative feedback for every user action**: Attendance shown
4) **Design dialogs**: After successful attendance, attendance is displayed in dialog box
5) **Offer error prevention and simple error handling**: Clear error message is displayed and error is prevented by comboboxes
6) **Permit easy reversal of actions**: User can cancel the action if he wants and will return to the main screen
7)**Support internal locus of control**: User is in control of the system
8)**Reduce short-term memory:** Interface is simple and easy to use and breadcrumbs make the user recall where he is and where he came from

Norman

1)**Use both knowledge in the world and knowledge in the head**The simple interface and a how to use guide in about achieve goal1
2)**Simplify the structure of tasks:** Simple tasks are provided with the use of buttons and keyboard shortcuts
3)**Make things visible**: The clear visible buttons and their feedback achieves goal3
4)**Get the mappings right:** The buttons perform the same task as their labels suggest
5)**Exploit the power of constraints, both natural and artificial**: A rectangular window imposes a constraint on user angle, they must be directly facing the camera only then their attendance will be marked

6)**Design for error:** All the possible error creating scenarios have been handled, like user provides input through combobox ensuring only valid inputs
7)**When all else fails, standardize**: We have followed the standards that any basic attendance app may provide like taking and viewing attendance

Cognitive walkthrough

1)The effect of the action is what user wants?
Yes it is
2)Will the user able to see that the action is available
Yes he will be
3)Would he know that this is the particular action he needs to perform
Yes as the interface is simple
4)Feedback after action is provided?
Yes it is given

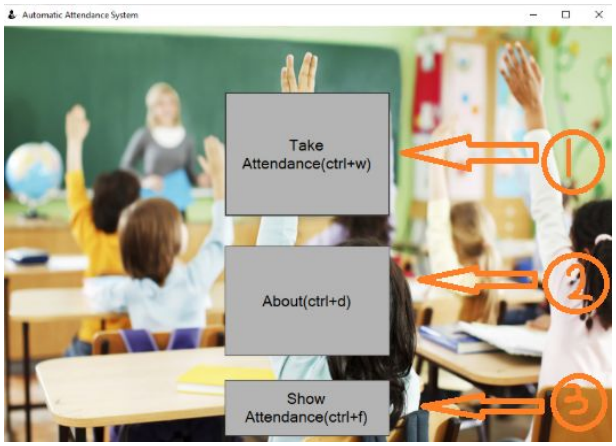
Nielsen's heuristic

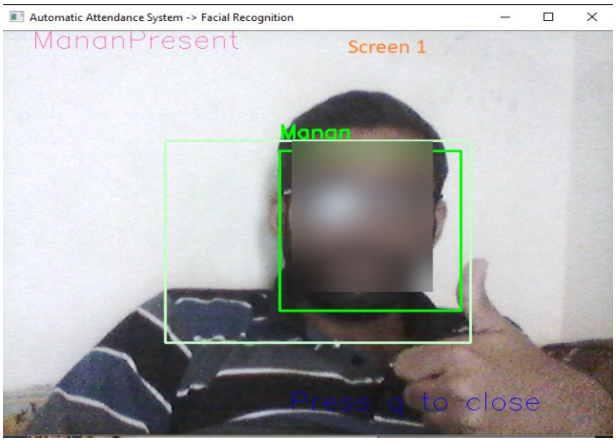
1)**Visibility of system status:**User clearly knows wat is going on by the window names
2)**Match between system and the real world:**The labels of the buttons are simple to understand and there is no confusion
3)**User control and freedom:** There are easy cancel operations if the user has selected something by mistake
4)**Consistency and standards:** The standards have been followed and the system has been simplified for use
5)**Error prevention:** Errors have been prevented by the help of combo boxes
6)**Recognition rather than recall:** We have focused on the pointer based interface for which we have clearly mentioned the labels on buttons so that the user may not get confused
7)**Flexibility and efficiency:** Shortcut keys have been added for frequent use
8)**Aesthetic and minimalist design:** Dialog boxes present the user with relevant information only
9)**Help users recognize, diagnose and recover from errors:** In the case of error, the user is prompted with how to recover from error message
10)**Help and documentation:** A little instruction set is provided in the about section

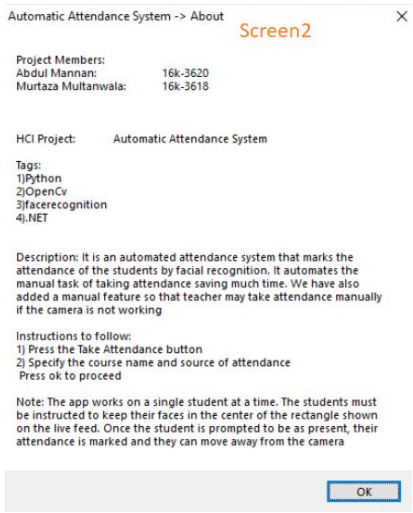
Facades & Fixations have been handled in our project

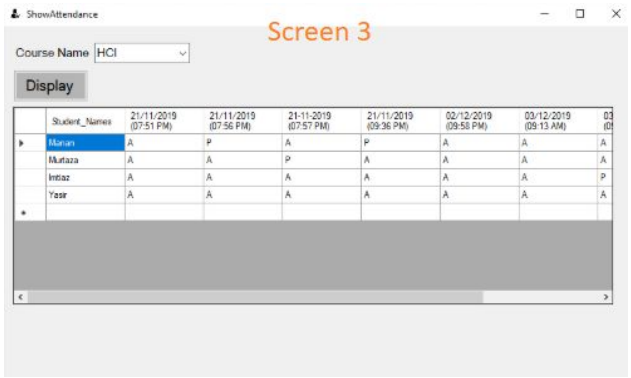
Design Rationale

The problem we want to tackle is the manual attendance procedure in the classrooms. Taking attendance by calling all the names takes a lot of effort and time and it is such a waste of time keeping in mind that we are living in an era of technology. So we wanted to automate this task for which we proposed an Automatic attendance system. It takes the attendance of the class automatically by facial recognition. It is a simple to use app and any novice user can start using it without any hurdles.









Each screen is focused to perform one particular functionality keeping in mind the inductive user interface. The user clearly knows where he is and what functionality he needs to perform.