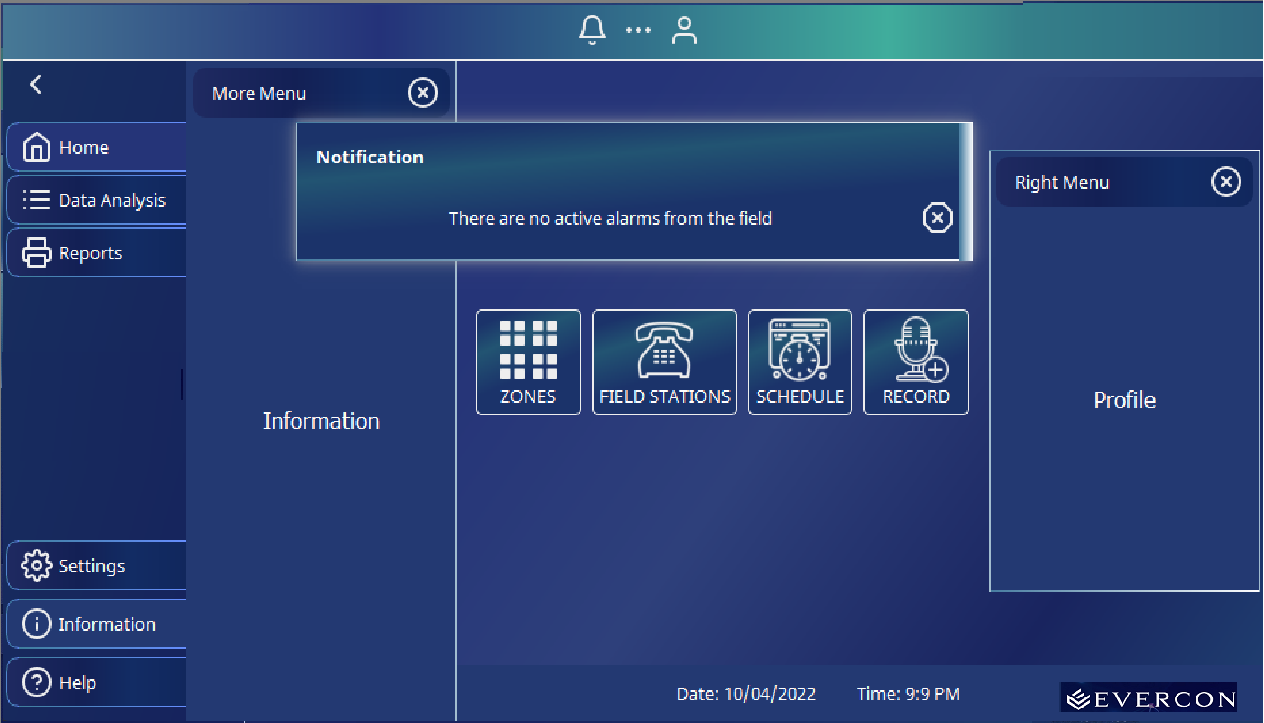
**Section 1 changes are applicable to all screens.**

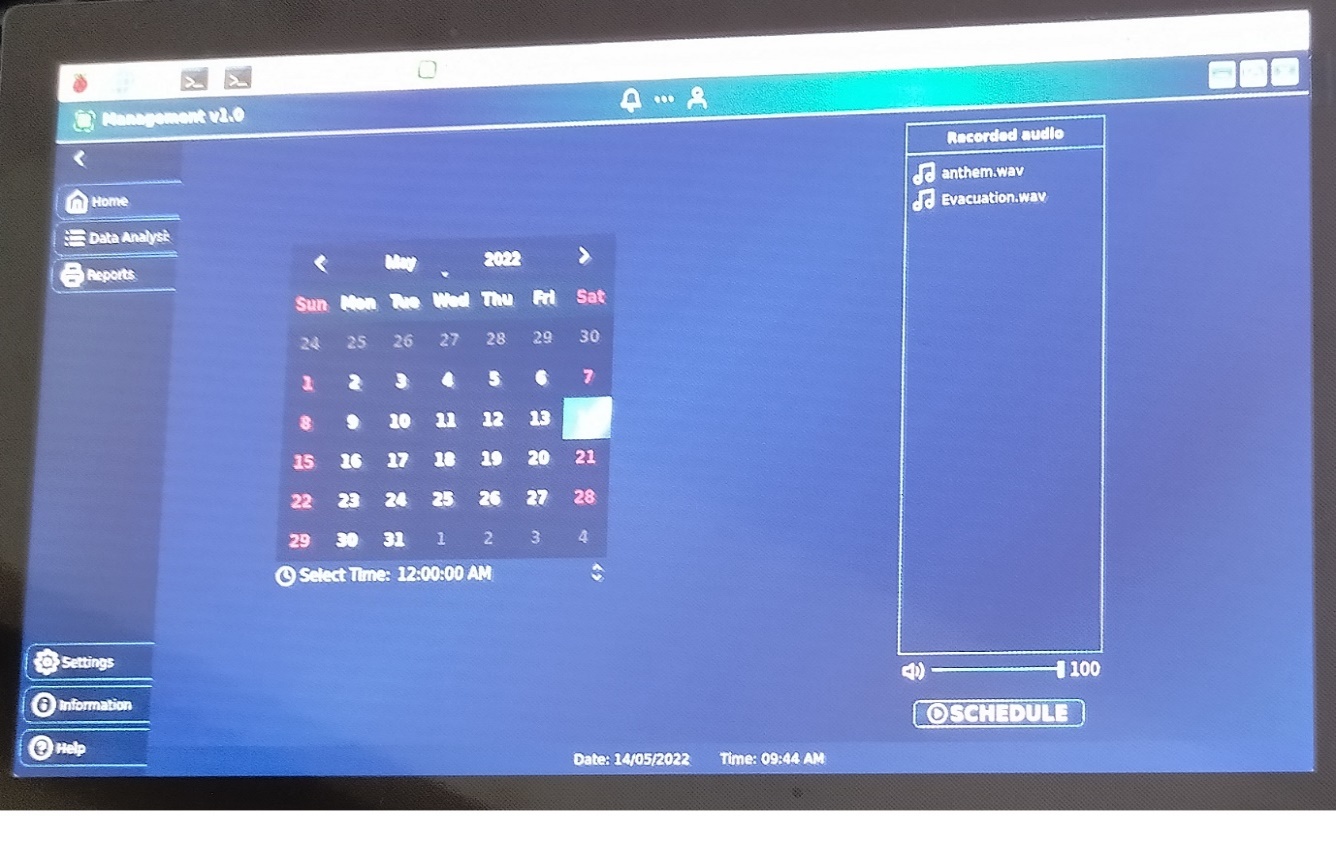
**Section 1 Changes (Applicable for all screens)**



1. Change the screen size from ‘sizable’ and fix it to 1280×720
2. Remove the Screen title as shown on the above screen and remove the Close, max, min, buttons.
3. Add a logo (Logo files attached). Let the logo be in the same background (not in the Blue background as shown). Add it to every screen
4. The ‘Sub Menu’ which has buttons like ‘Home’, ‘Data Analysis’, ‘Reports’ etc. should always be visible and should go hidden on the press of a button. Now it is hidden by default.
5. Rename the ‘Right Menu’ as ‘Profile’. Load this from ‘profile.txt’. Place this file somewhere and let me know the path.
6. Load the Notification Menu content from a list (or any other python data type) and let me know. When I integrate the code, I tell someone to fill that List.
7. Time (displayed at the bottom of every screen) should be real-time with Date/Month/ year and Hours:Minutes:Seconds
8. Help should be loaded from a file help.txt. I can modify the content later.
9. Size of all buttons, and texts (which are expected by the user to touch, should be close to the size of the tip of the pointing finger. For example, in the schedule screen, the font size of recorded audio files is very small and it is expected to be selected by the user.
10. Attached ‘Commad\_handler.py’. Please add a command to know the status of each screen.

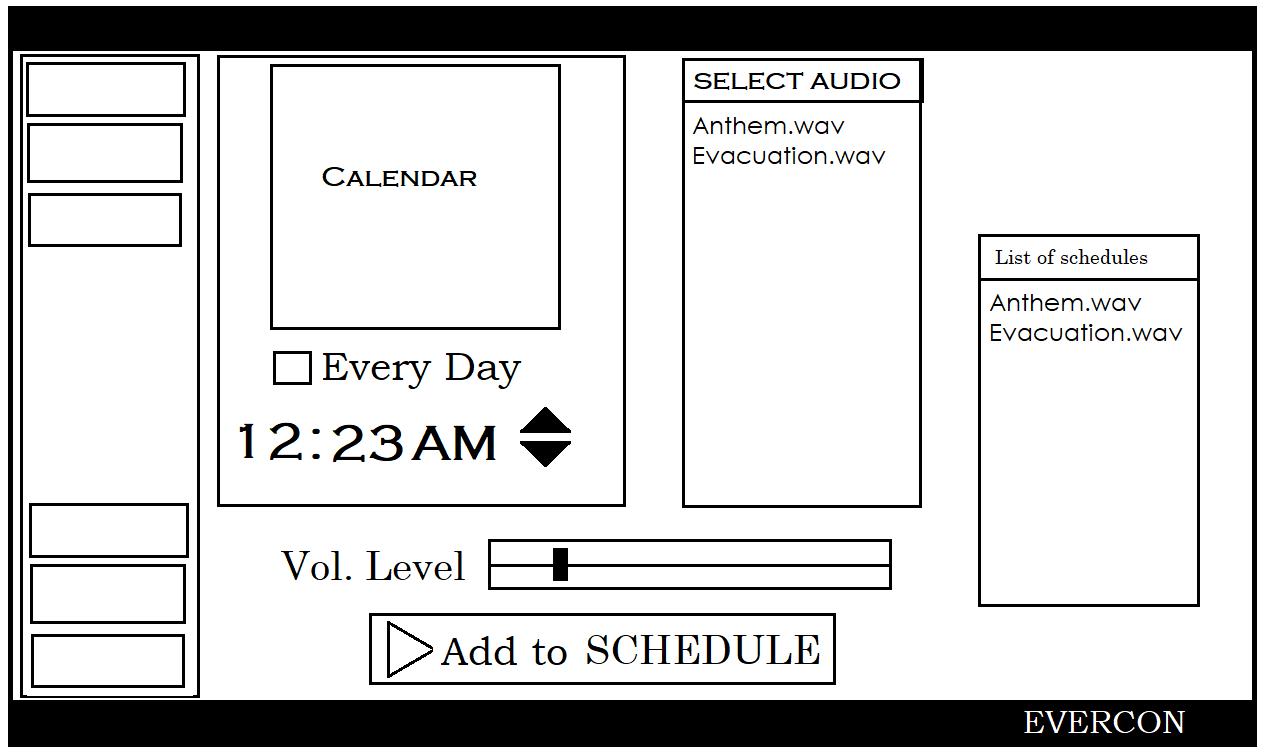
For example, if there is a new schedule added, provide a command to dump the content of the data structure.

**Section 2 Screen specific changes**



1. SCREEN SCHEDULE, SCREEN 5

Change it as below.



Summary of changes:

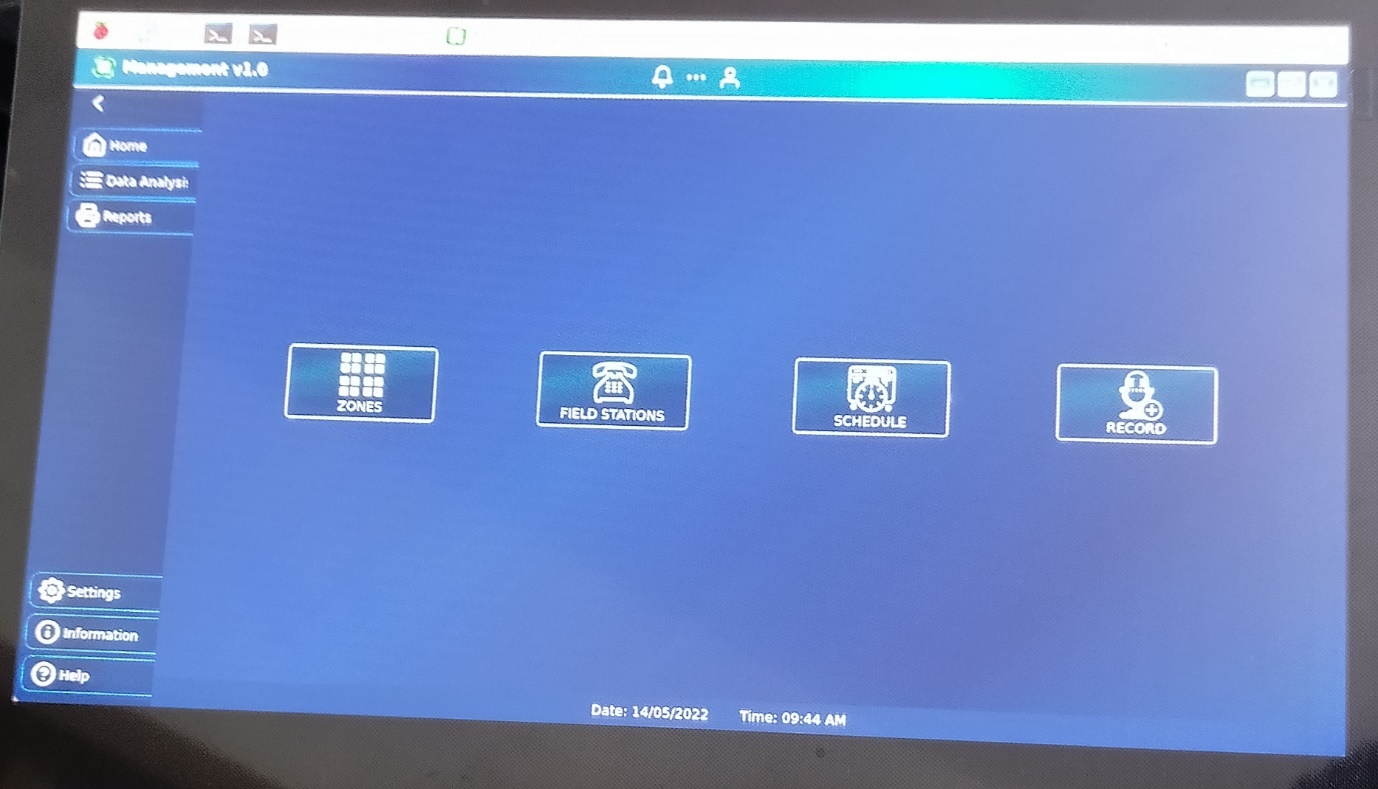
1. Add an additional Everyday button, if that is checked, it should discard the calendar settings and repeat every day instead of a specific date. That is the plan.
2. Add an additional column, a list of schedules. This should be taken from an actual Python list (or some other Python datatype).
3. Select Audio should list all the files in the ‘Recorded’ directory in the current directory.

Python Programming task:

1. On pressing ‘Add to schedule’ button, it should make an entry to a list of ‘schedules’, a list (or another datatype). That should have fields to,

* Store the date and time
* Whether it repeats every day
* Selected Audio file name
* Selected Audio volume level

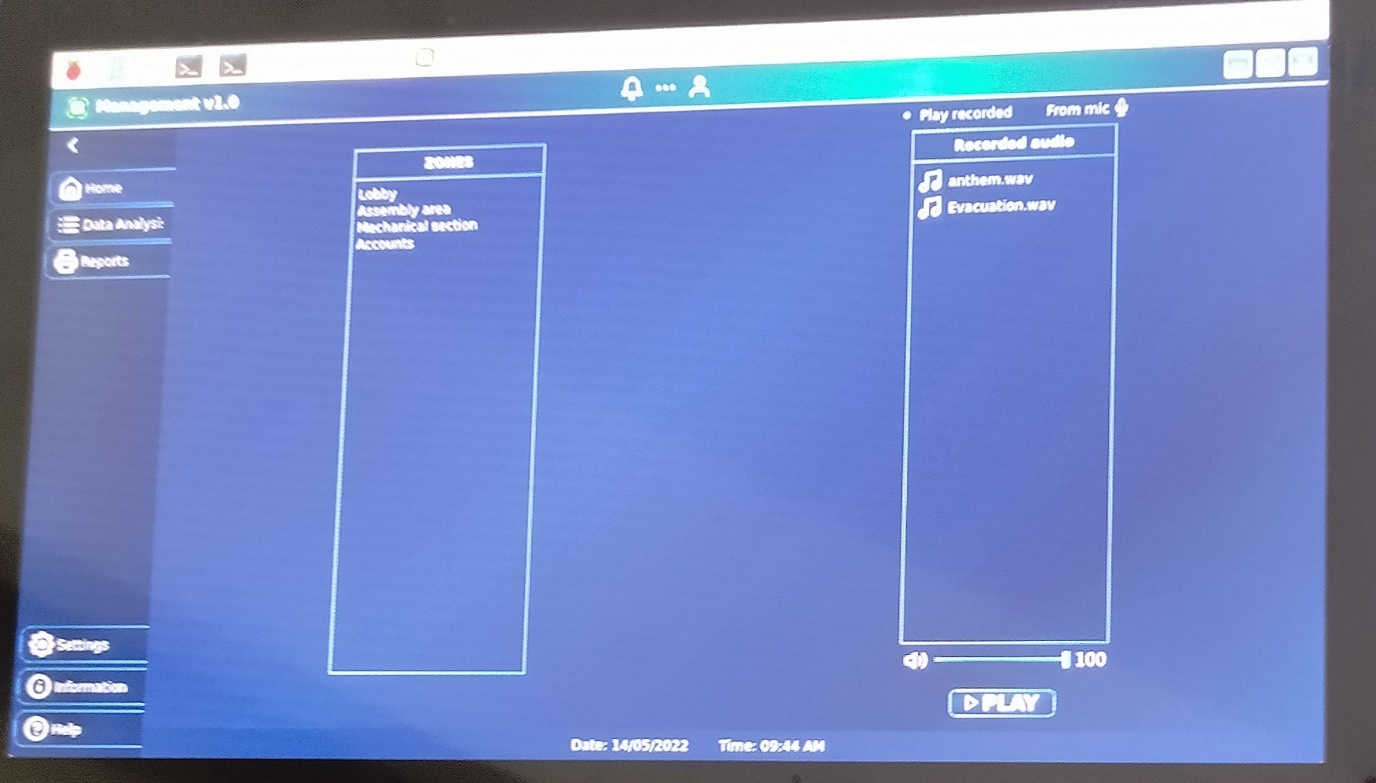
1. A Thread should run every minute and check whether any of the scheduled tasks are matching. If it is matching, just print the time there. I will add the code there with the help of a local engineer.
2. Select Audio should list all the files in the ‘Recorded’ directory in the current directory.
3. SCREEN MAIN, SCREEN 2

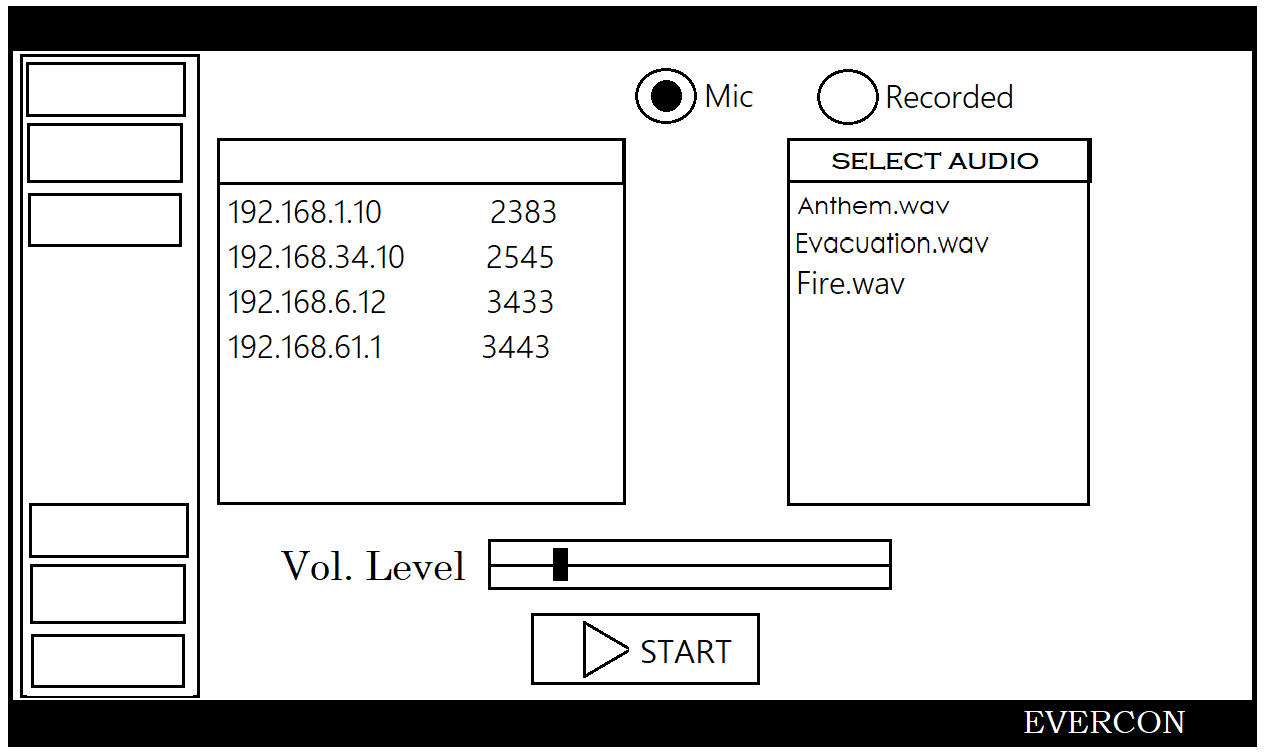


1. There are not many changes here, you can make fond of ZONES, FIELD Stations, etc. slightly bigger, not too much.

2. Add a new block right below ZONES named CAMERA and add an icon there.

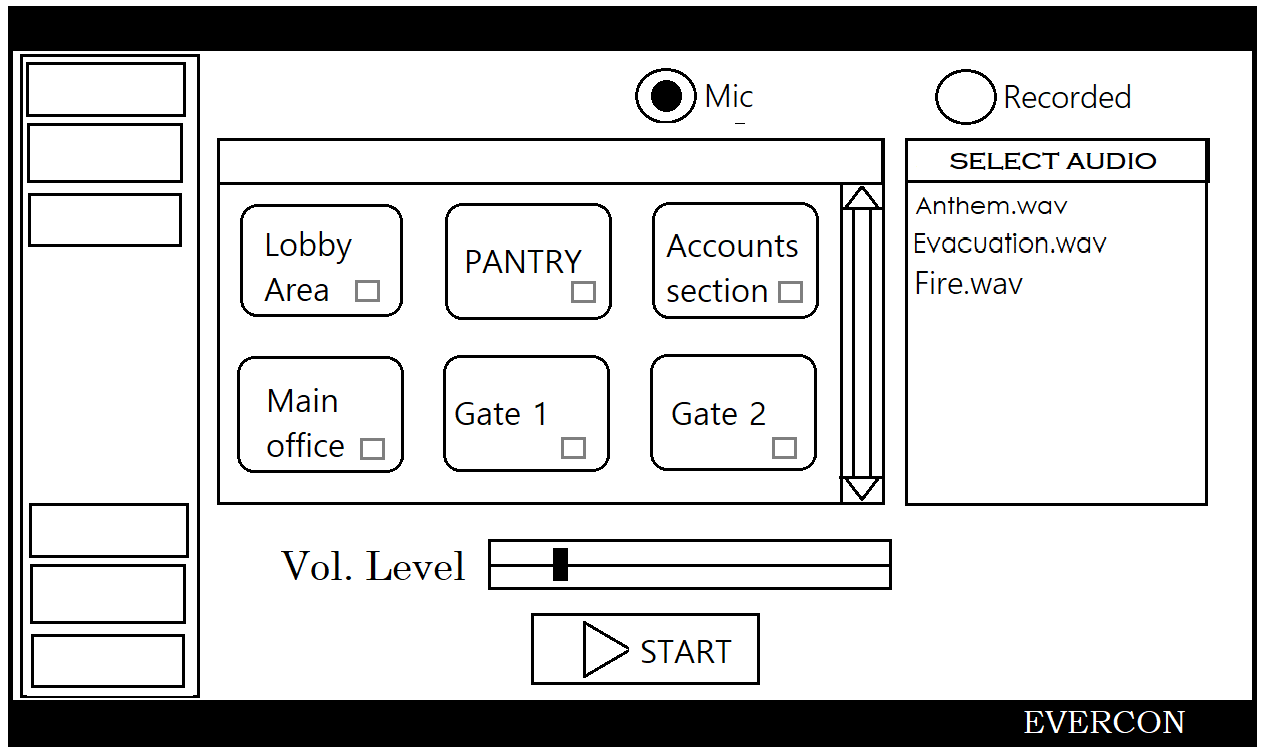
1. SCREEN STATIONS, SCREEN 2





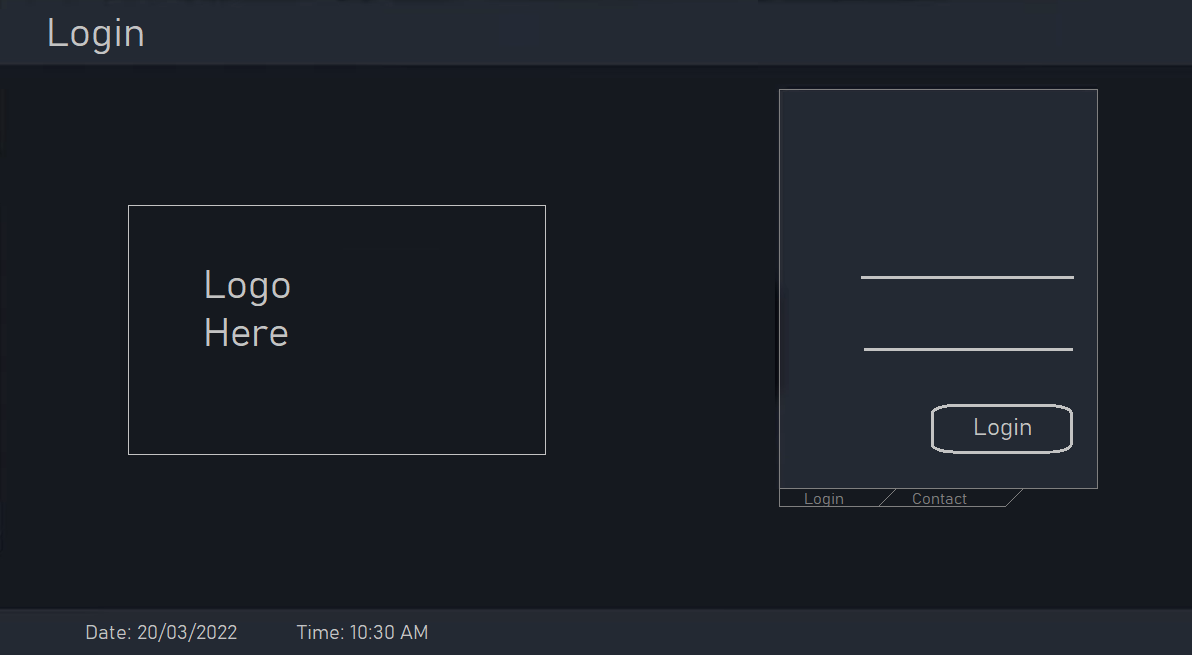
Changes.

1. Make the Font size bigger
2. Arrange things as shown above
3. When the START button is pressed, I should store the following information in any data structure and print a message to the console. I should be able to print this data by entering a command.
4. The data structure should have the following,
5. Flag indicating whether play from the mic or recorded file
6. Volume level
7. IP Address and extension
8. Recorded file name
9. If ‘Recorded’ is selected and no audio file is selected, then it should show an error popup
10. SCREEN ZONE STATION, SCREEN 3



1. If ‘Recorded’ is selected and no audio file is selected, then it should show an error popup.
2. When the Screen is loaded, it should create as many icons as in a list (Later, I should be able to create new zones through commands, for now, we can statically fill this data structure).
3. When the Start button is pressed, it prints a message to the Console, pushes the zones selected to a list, and Audio File name and Volume level to a data structure.
4. I should be able to print current zones using a command.
5. LOGIN MAIN, SCREEN 1

SCREEN 1, LOGIN SCREEN



1. An on-screen keyboard should be implemented to enter a username and password. Password characters should not get displayed. You should compare it with a list of usernames and passwords. Then load the next screen, only if it matches with one of those.
2. ‘Contact’ tab should be touchable, not as shown in the image.
3. When the Contact tab is selected, It should load the content from a text file. I should be able to modify the content of the file later.
4. Where it is marked as ‘Logo Here’ is should show logo and



1. Incoming Call, SCREEN 6



* + 1. This should be a very small screen and should come up as a pop-up screen over other screens. I should be able to invoke this by giving a command for now. A local engineer will integrate when the actual application is ready.
    2. It should display the Text ‘Incoming Call from Extension #2343’. The extension should be picked from a variable (which I will load it later, you can temporarily load a value which I can set by command).
    3. There should be ACCEPT and DISCARD buttons on the screen.