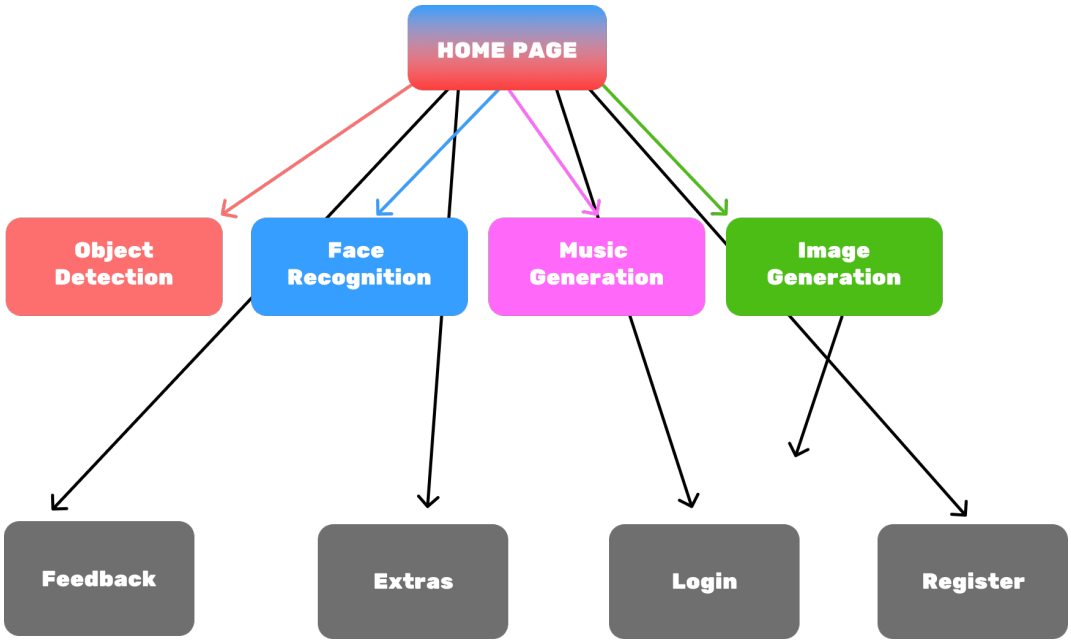
Criterion B: Design

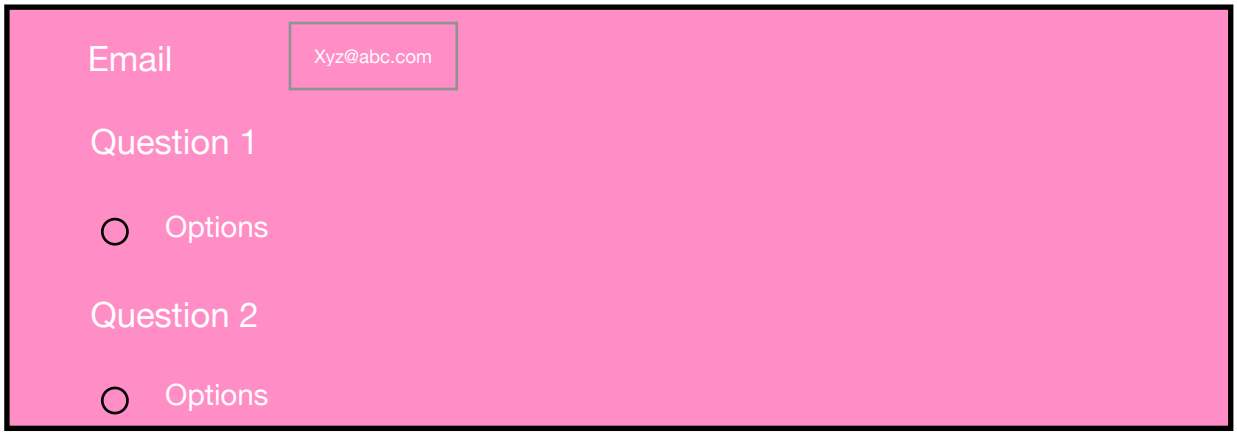
Site-Map



Fields required in the Menu Bar (Header)

Feedback:

Feedback links to a google form which will have the following content



Extras:

This page contains definitions of all the important terms used In the website

Header

Definition of All terms

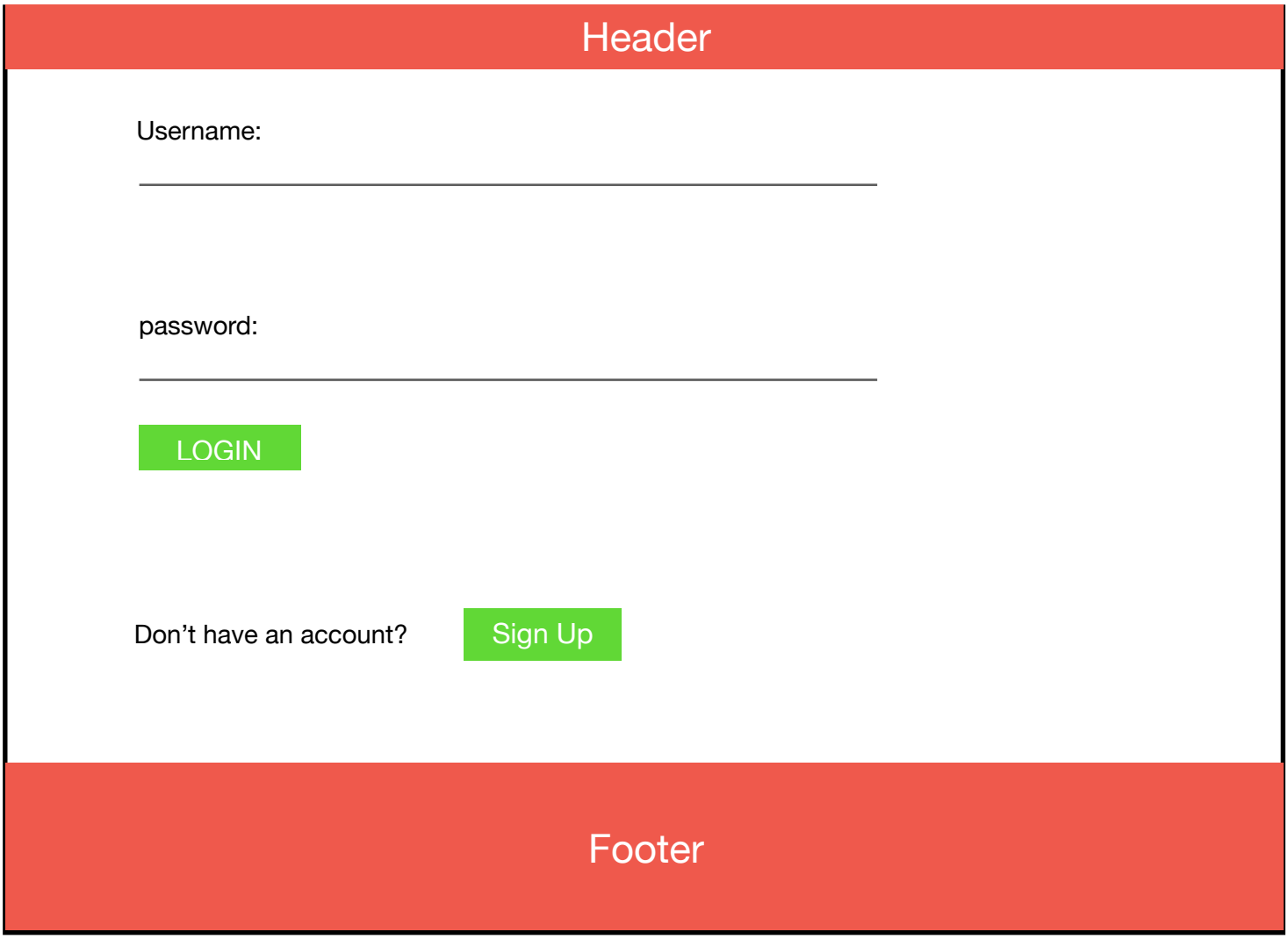
Term: Definition

Term: Definition

Term: Definition

Footer

Login Form:


Header
Username:
password:
LOGIN
Don’t have an account?
Sign Up
Footer

Register Form:

Header

Username:

Email:

Password:

Password

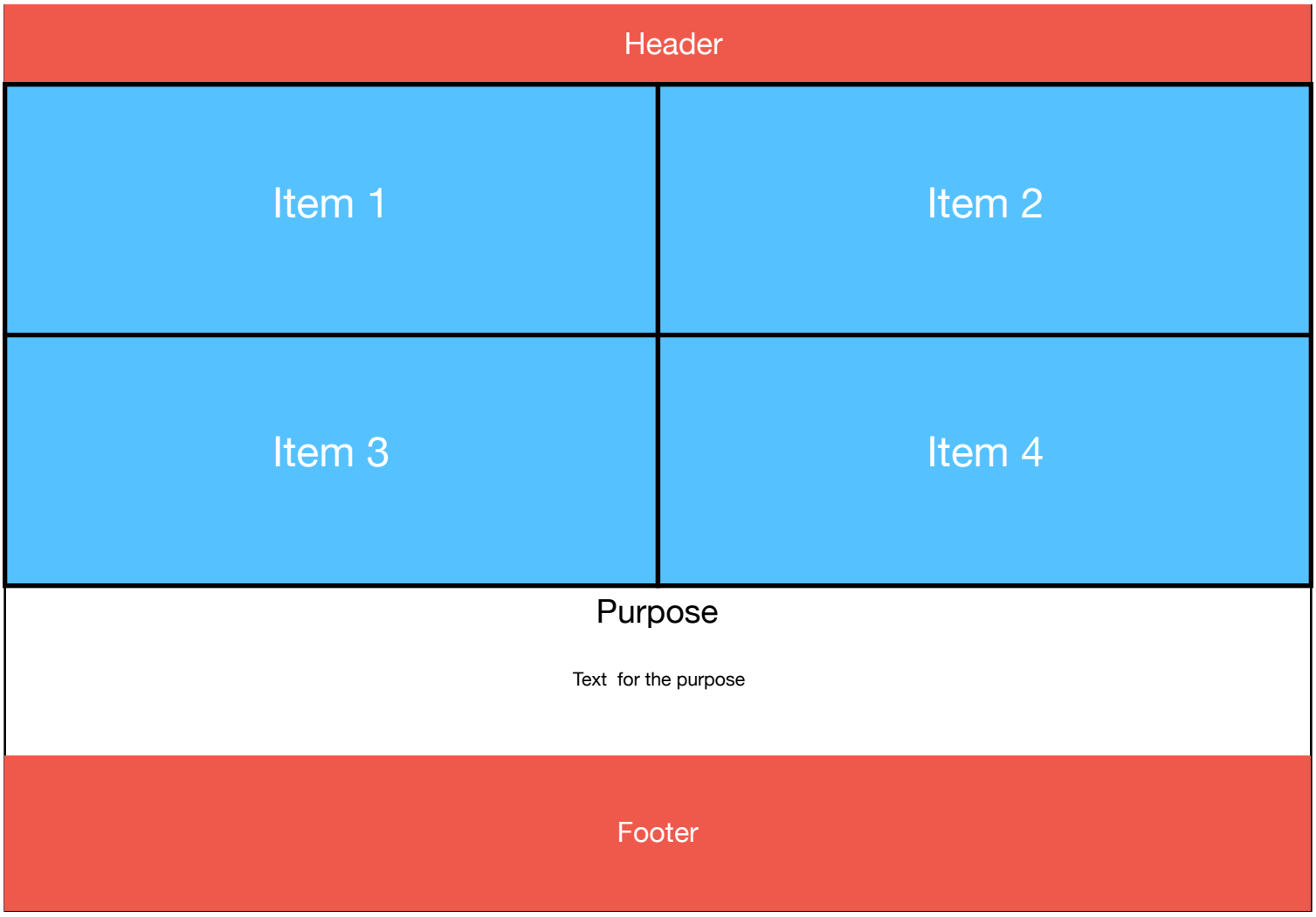
SIGNUP

Already have an account? LOGIN

Footer

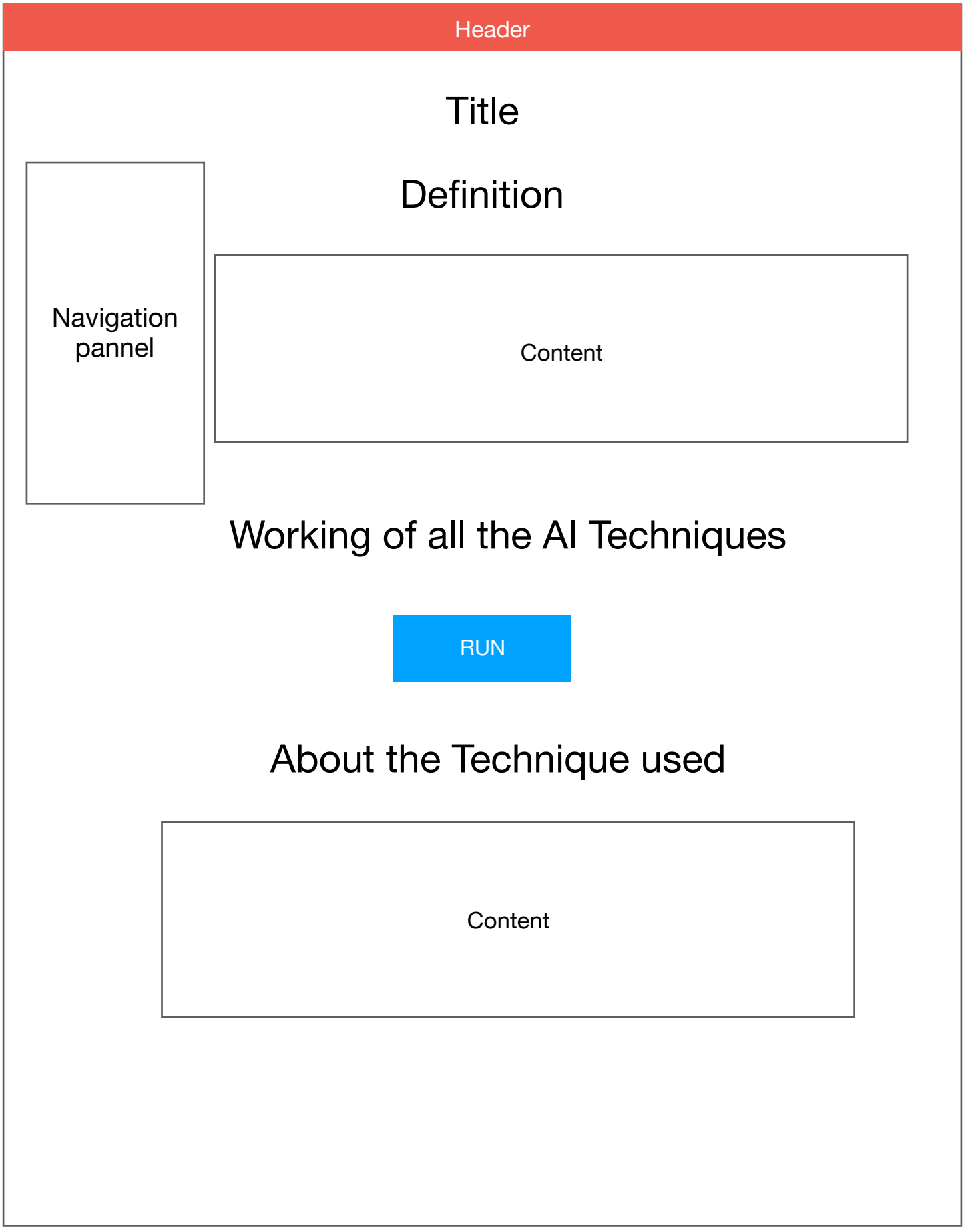
Fields required in Footer

Home Page



Items Page

All the Items have a common page design



Explanation of other Techniques

Name:

Name: Explanation

Explanation

Table containing the Details about the Technique used

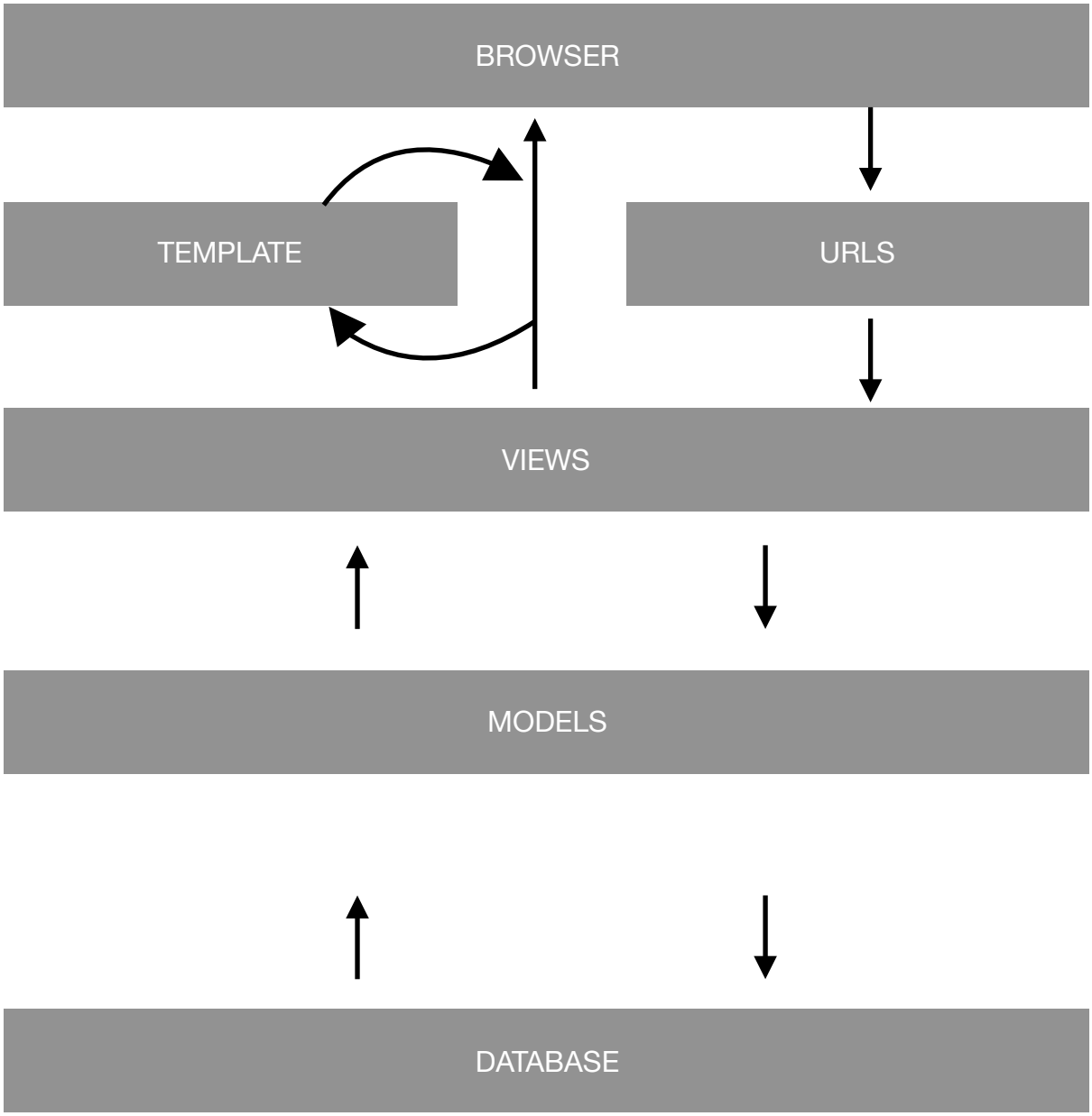
Abcd	Abcd	Abcd	Abcd	Abcd	Abcd
Abcd					
Abcd					
Abcd					

Code

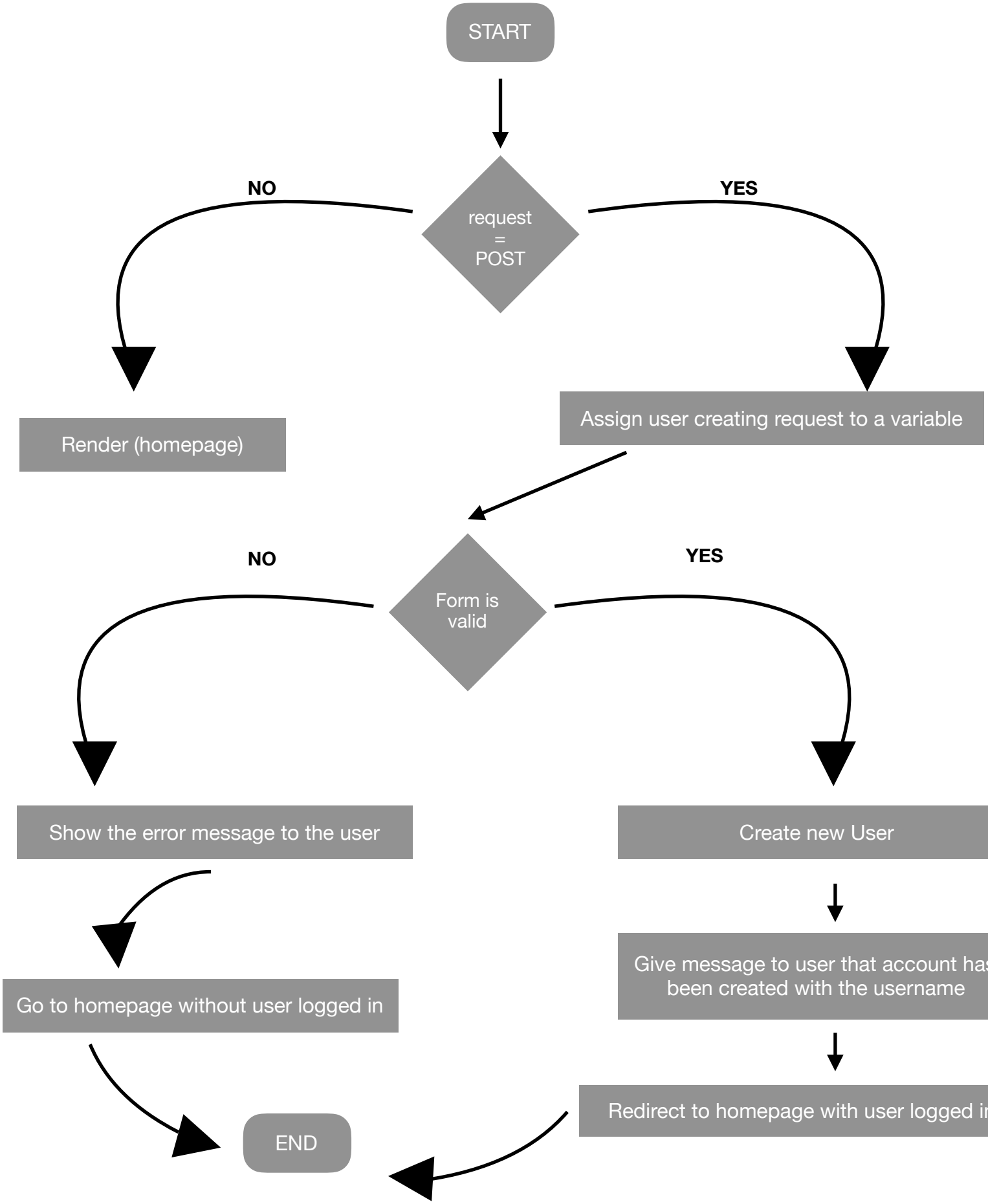
```
print("Code for all the techniques")
```

Flowchart of all the functions

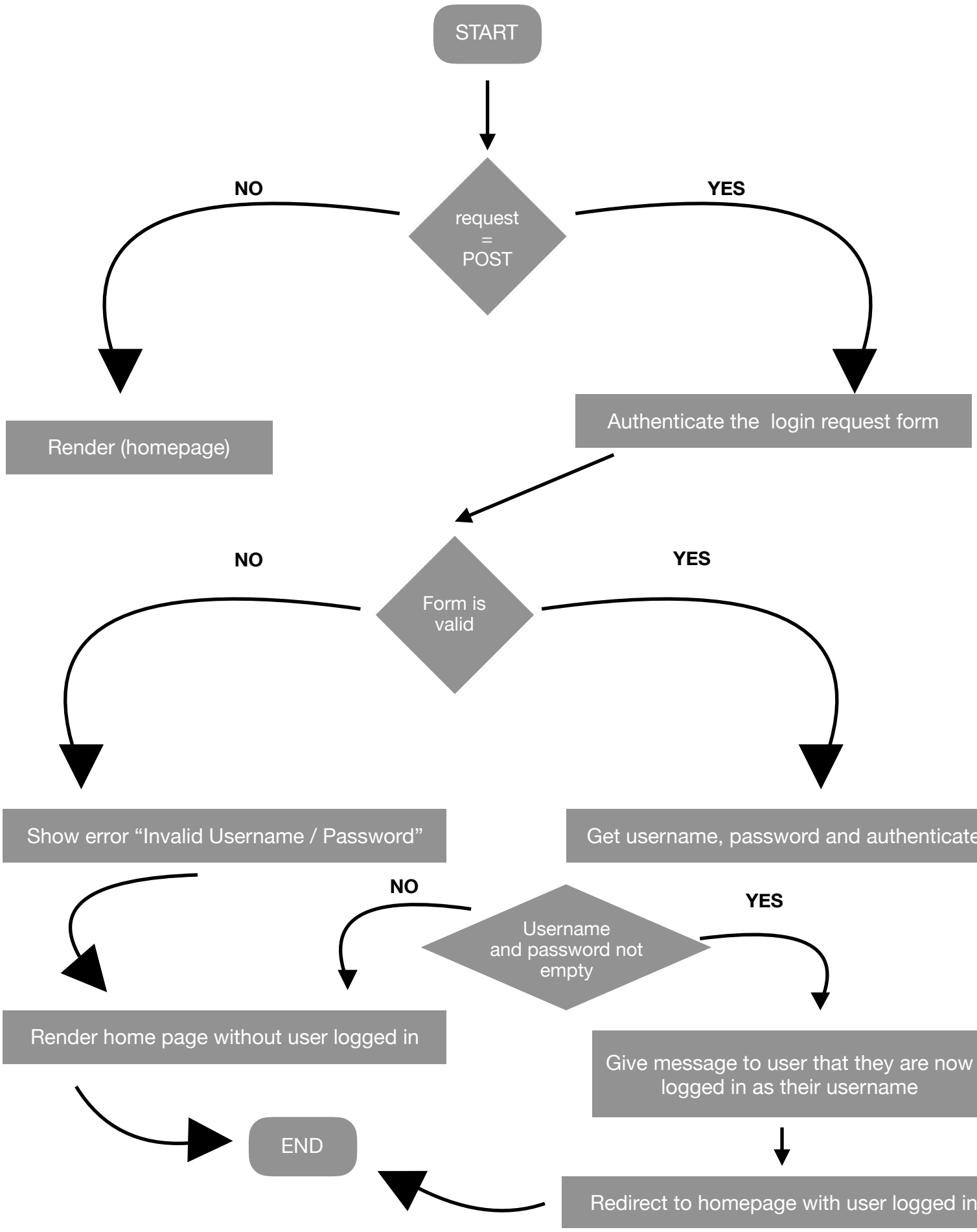
Django uses model-template-view architectural pattern, it gets data from the model, gives to the view which can have various functions to manipulate the data before passing to the template which then displays it on the website.



Register

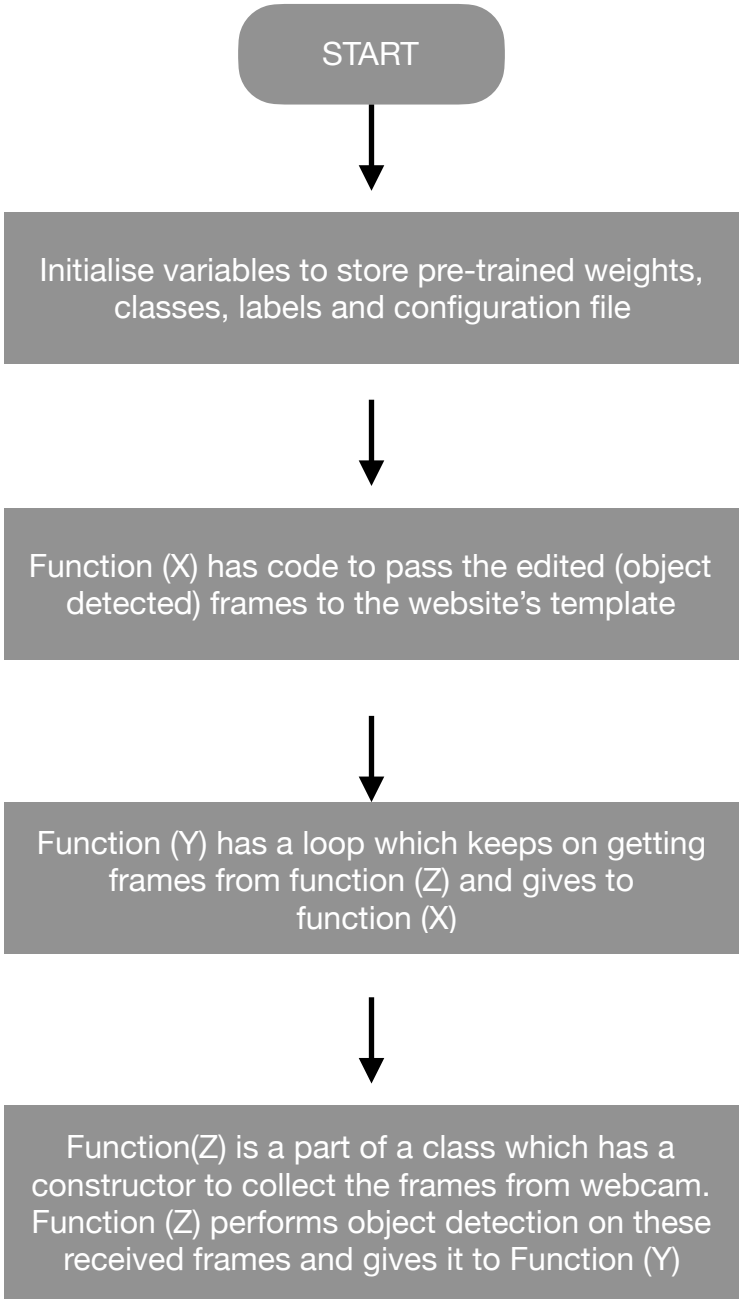

START
request = POST
NO
YES
Render (homepage)
Assign user creating request to a variable
Form is valid
NO
YES
Show the error message to the user
Create new User
Go to homepage without user logged in
Give message to user that account has been created with the username
Redirect to homepage with user logged in
END

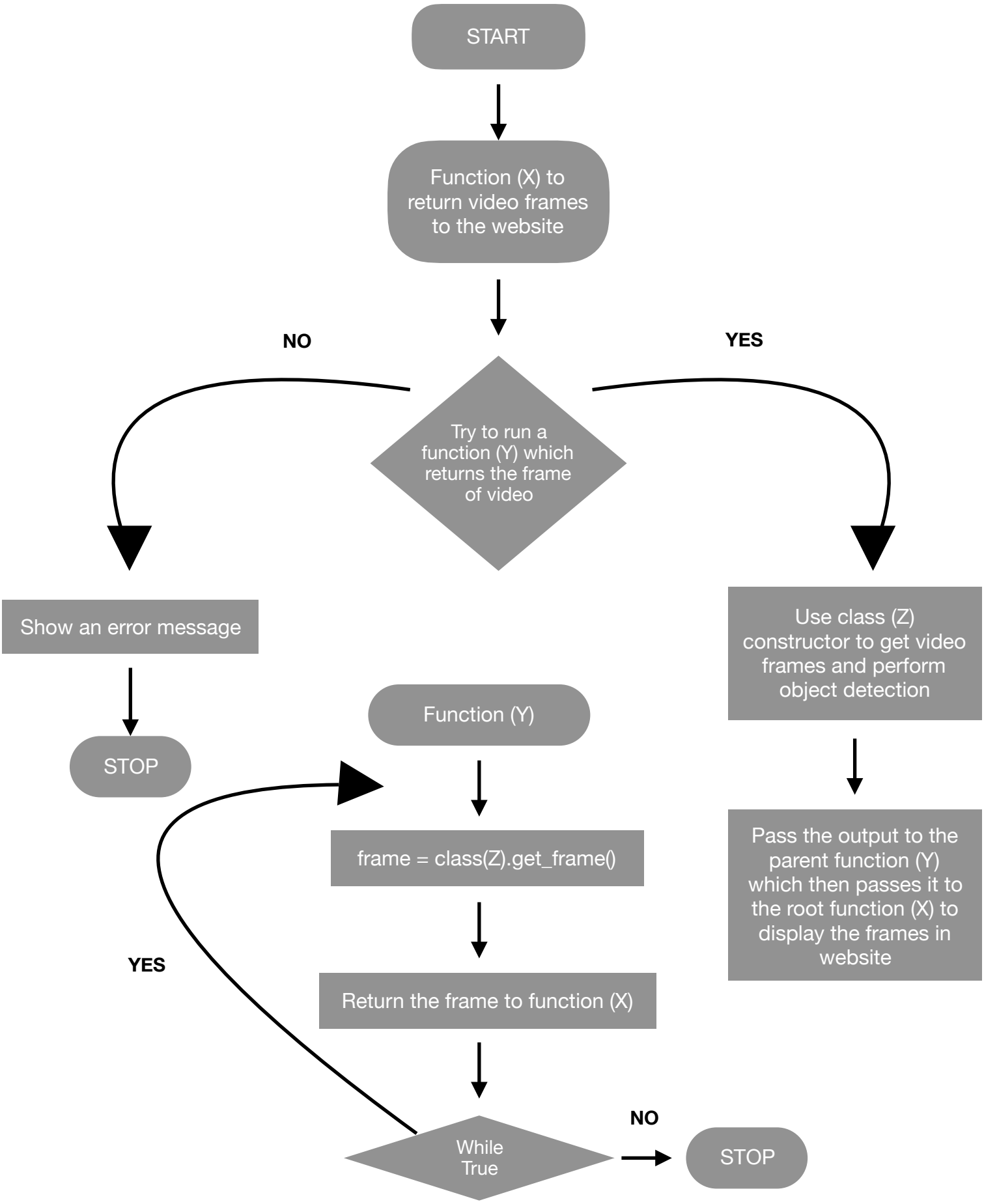
Login

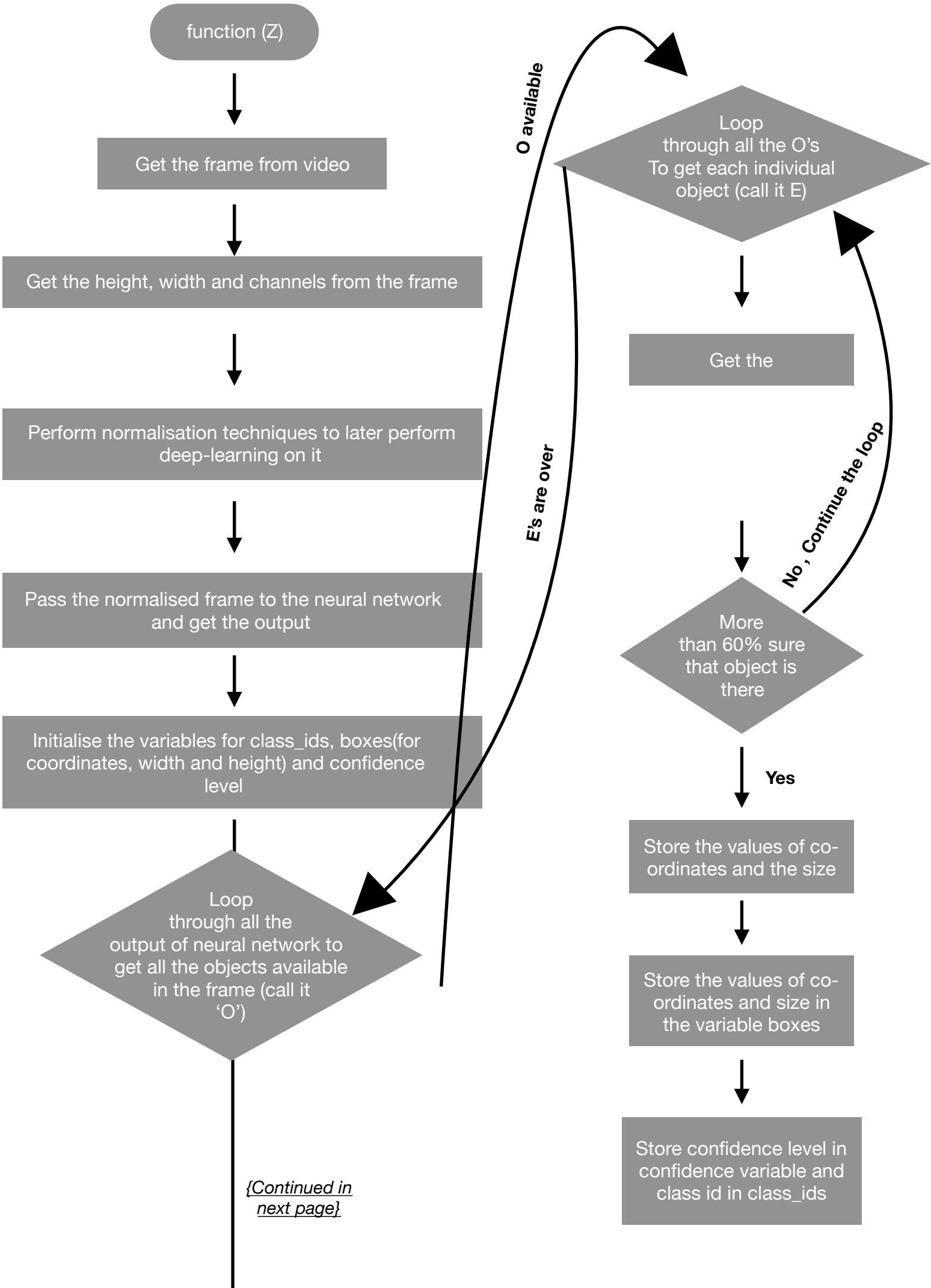

START
request = POST
NO
YES
Render (homepage)
Authenticate the login request form
Form is valid
NO
YES
Show error “Invalid Username / Password”
Get username, password and authenticate
Username and password not empty
NO
YES
Render home page without user logged in
Give message to user that they are now logged in as their username
END
Redirect to homepage with user logged in

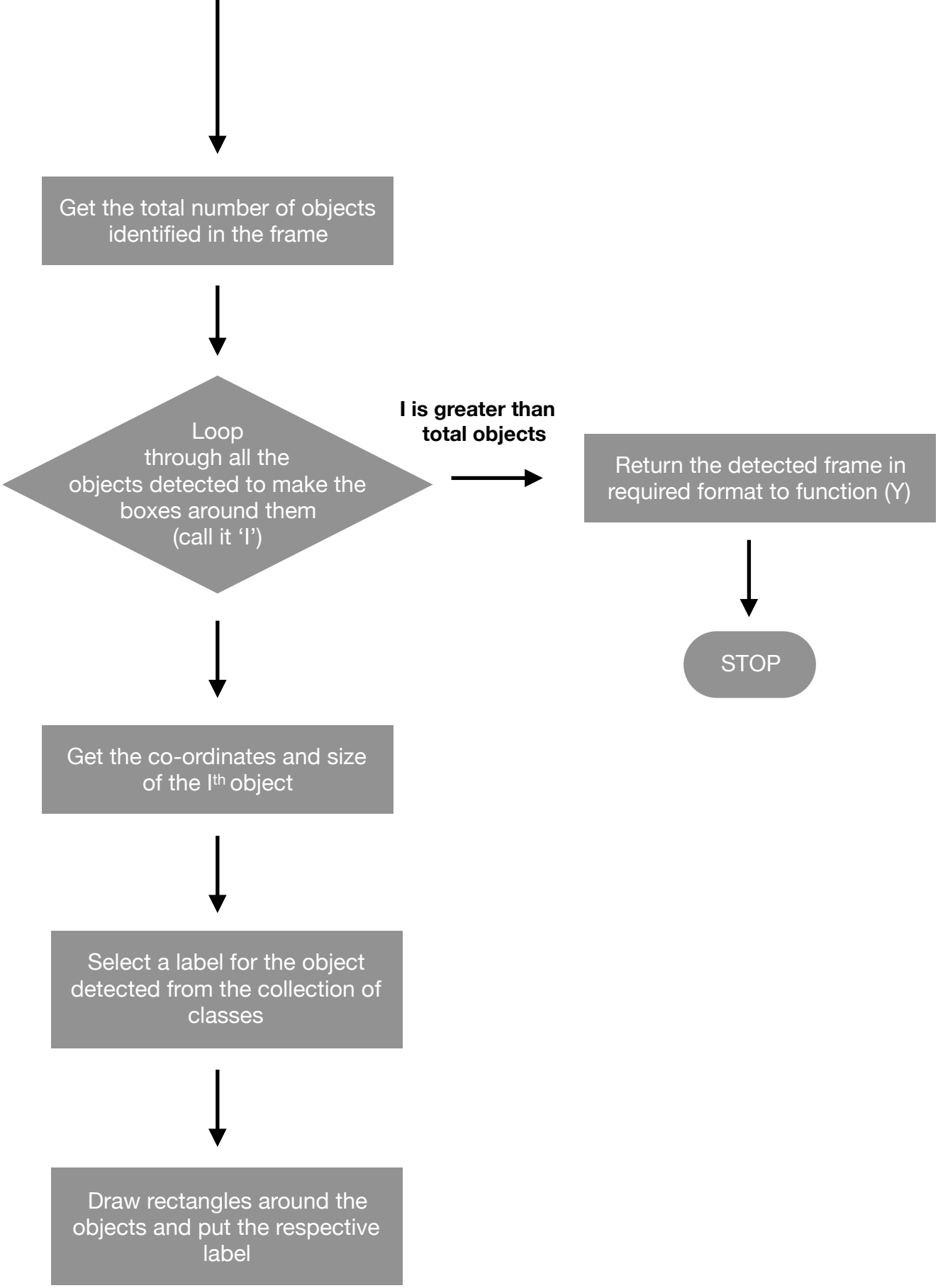
Object Detection

The following flowcharts shows the working of object detection code. The below flow chart shows the overview of how each function works




START
Function (X) to return video frames to the website
NO
YES
Try to run a function (Y) which returns the frame of video
Show an error message
STOP
Use class (Z) constructor to get video frames and perform object detection
Pass the output to the parent function (Y) which then passes it to the root function (X) to display the frames in website
Function (Y)
frame = class(Z).get_frame()
Return the frame to function (X)
YES
While True
NO
STOP

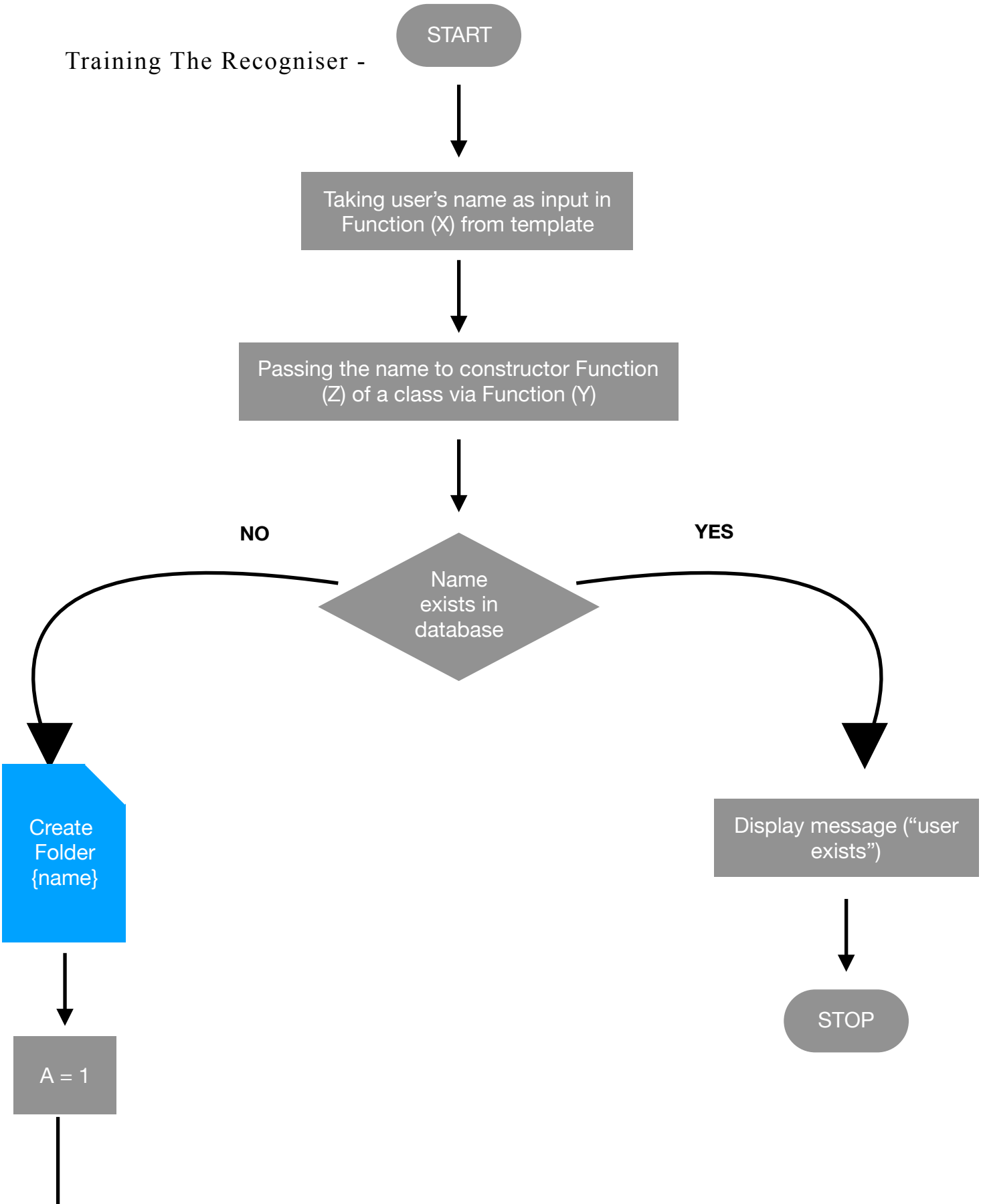

function (Z)
Get the frame from video
Get the height, width and channels from the frame
Perform normalisation techniques to later perform deep-learning on it
Pass the normalised frame to the neural network and get the output
Initialise the variables for class_ids, boxes(for coordinates, width and height) and confidence level
Loop through all the output of neural network to get all the objects available in the frame (call it 'O')
{Continued in next page}
O available
E's are over
Loop through all the O's To get each individual object (call it E)
Get the
More than 60% sure that object is there
No , Continue the loop
Yes
Store the values of co-ordinates and the size
Store the values of co-ordinates and size in the variable boxes
Store confidence level in confidence variable and class id in class_ids

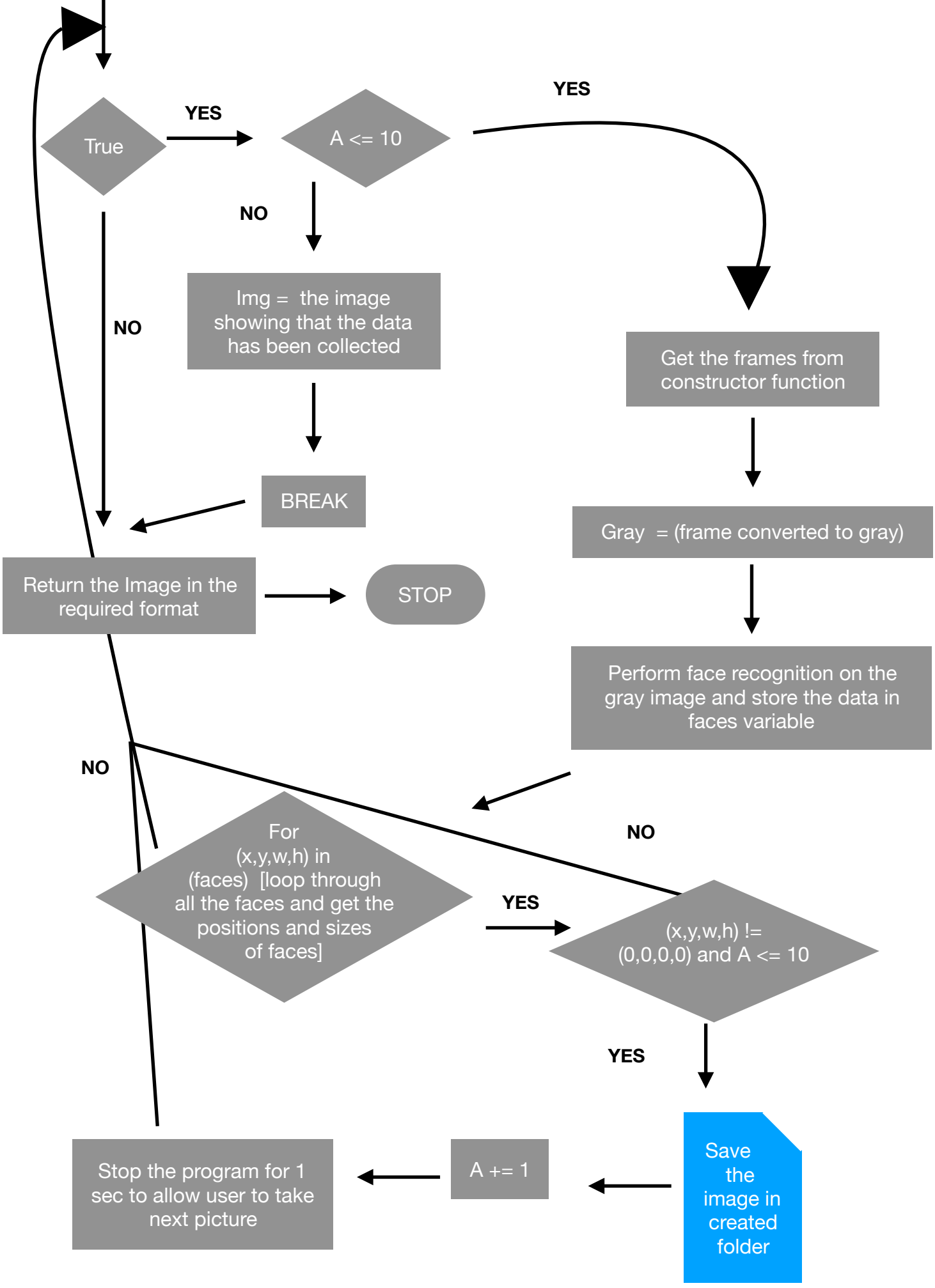

Get the total number of objects identified in the frame
Loop through all the objects detected to make the boxes around them (call it 'I')
I is greater than total objects
Return the detected frame in required format to function (Y)
STOP
Get the co-ordinates and size of the Ith object
Select a label for the object detected from the collection of classes
Draw rectangles around the objects and put the respective label

Face Recognition

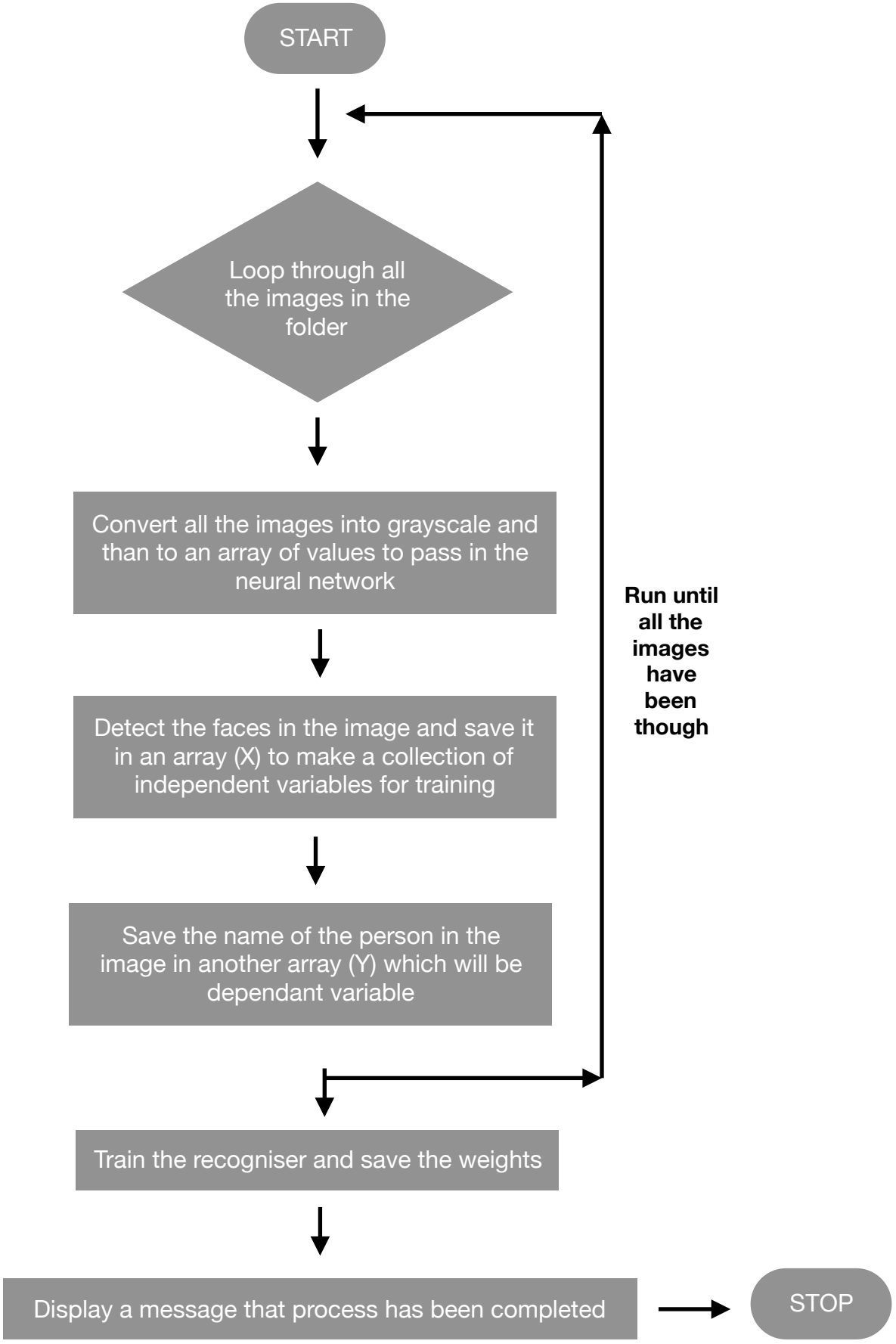
The following Flowchart displays how the data collection for the Face Recognition will work. (The pipeline to perform Face Recognition is similar to that of Object Detection)

Training The Recogniser -




True
YES
A <= 10
YES
NO
Img = the image showing that the data has been collected
NO
Get the frames from constructor function
BREAK
Gray = (frame converted to gray)
Return the Image in the required format
STOP
Perform face recognition on the gray image and store the data in faces variable
NO
NO
For (x,y,w,h) in (faces) [loop through all the faces and get the positions and sizes of faces]
YES
(x,y,w,h) != (0,0,0,0) and A <= 10
YES
Save the image in created folder
A += 1
Stop the program for 1 sec to allow user to take next picture

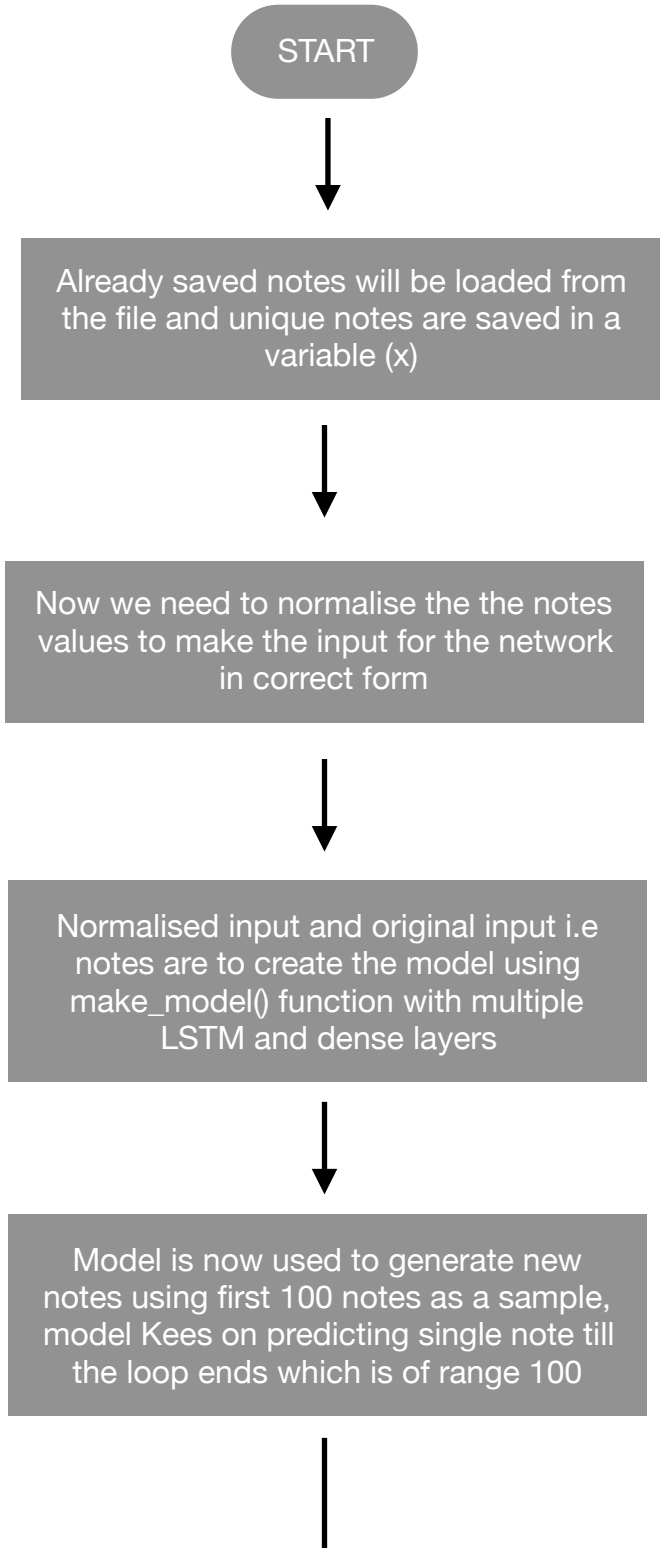
After saving the images, recogniser has to be trained on those saved pages and weights have to be updated which is done in update_weights function.



Now, We'll look at how Face recognition works.

Face recognition works in the similar fashion, just instead of saving the images to a folder the image is displayed to the user with faces bounded in rectangles using the trained weights and open cv library.

Music Generation:



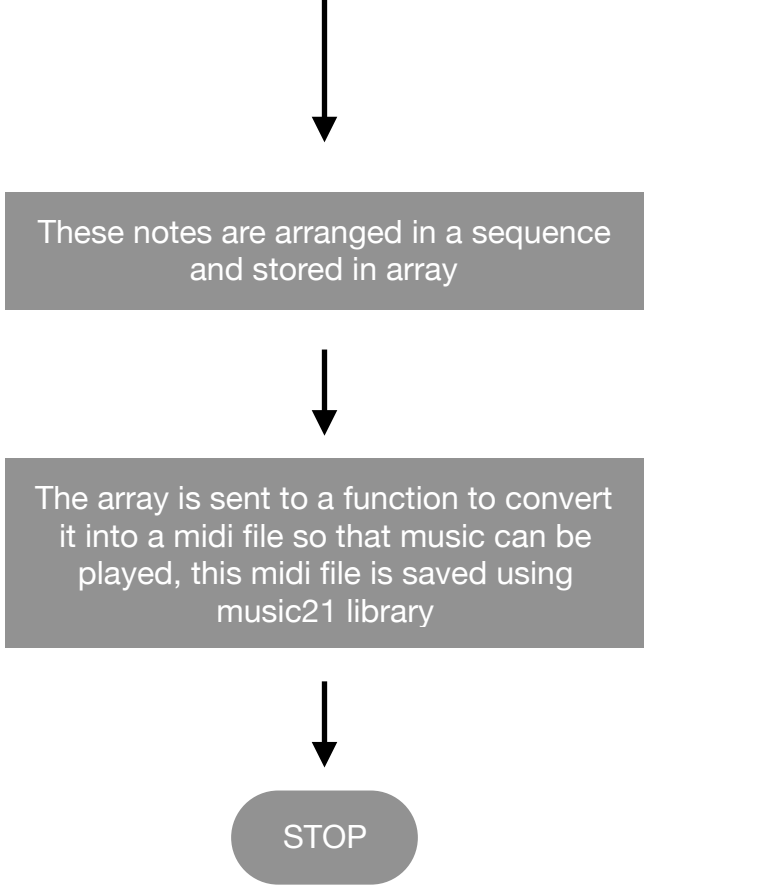
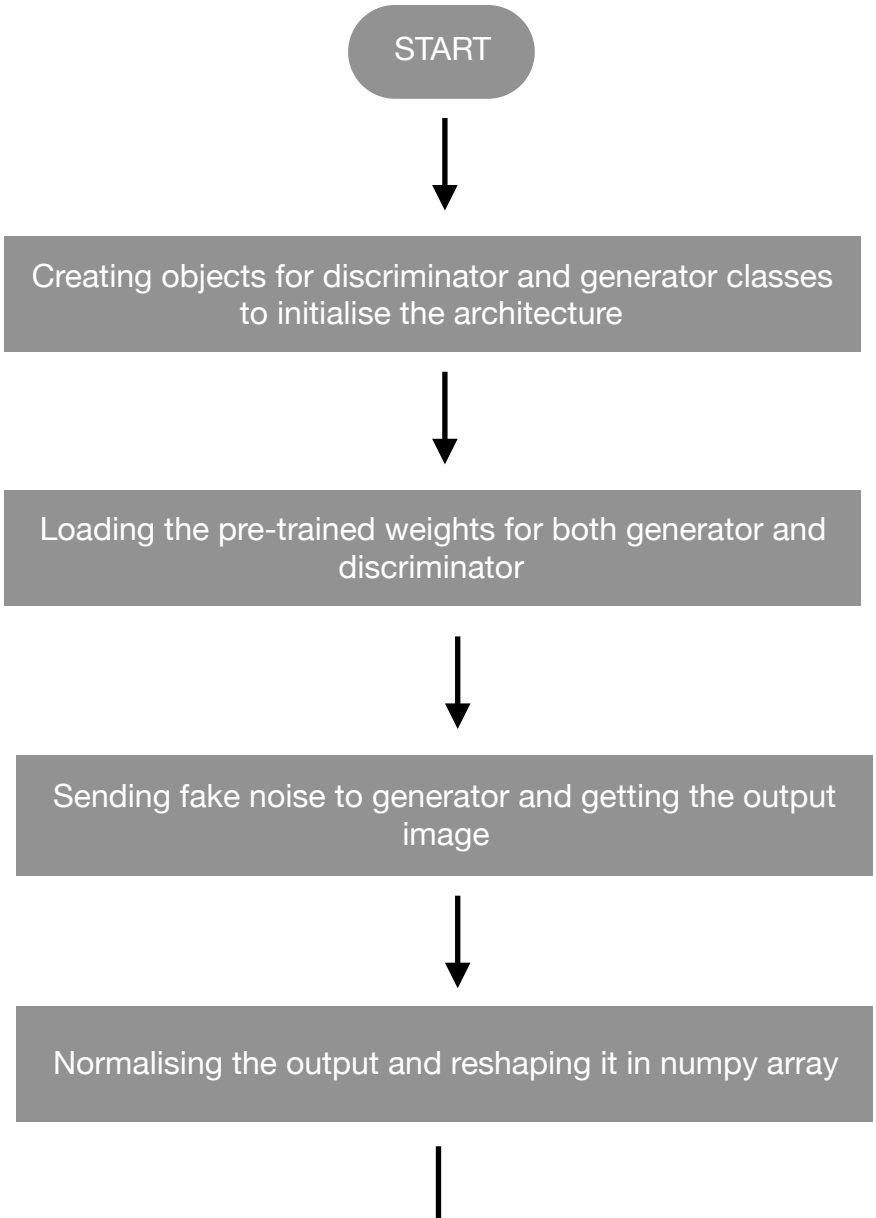
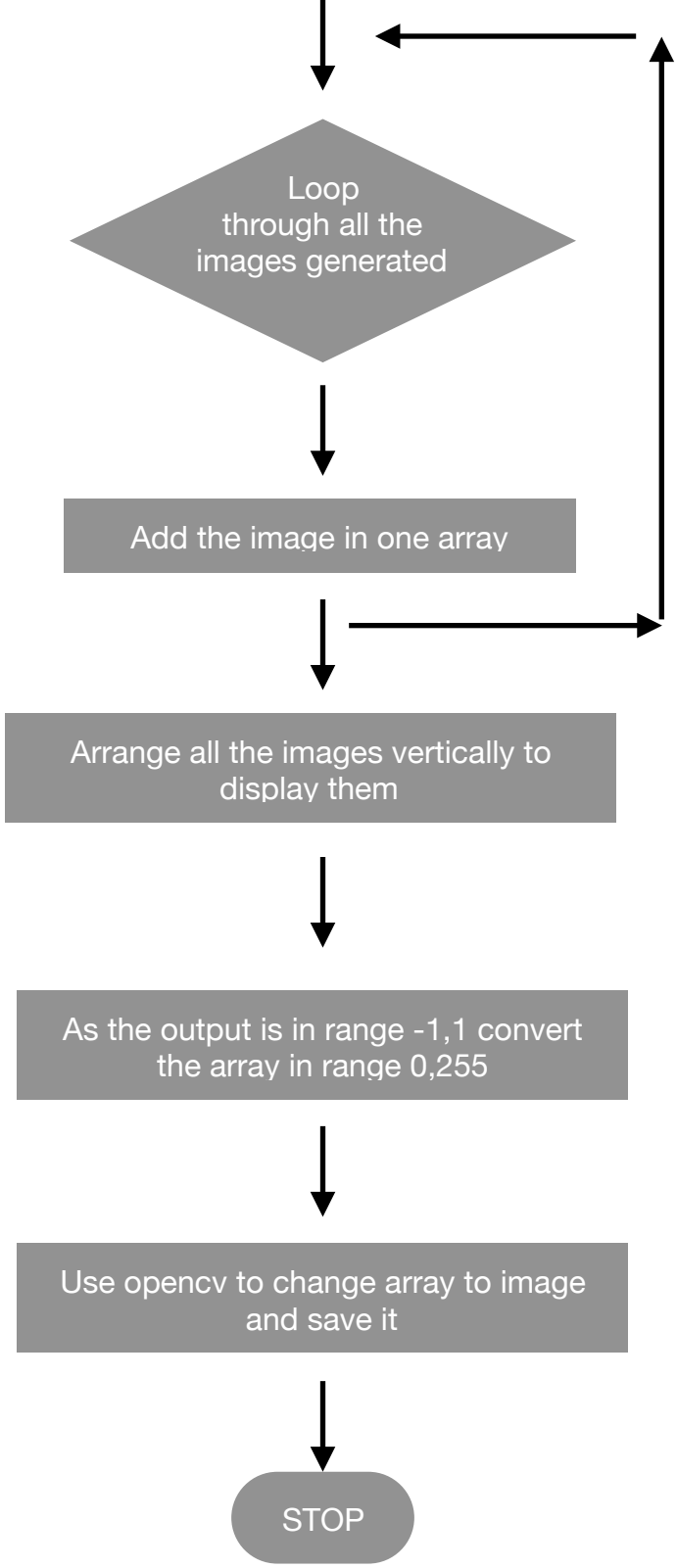
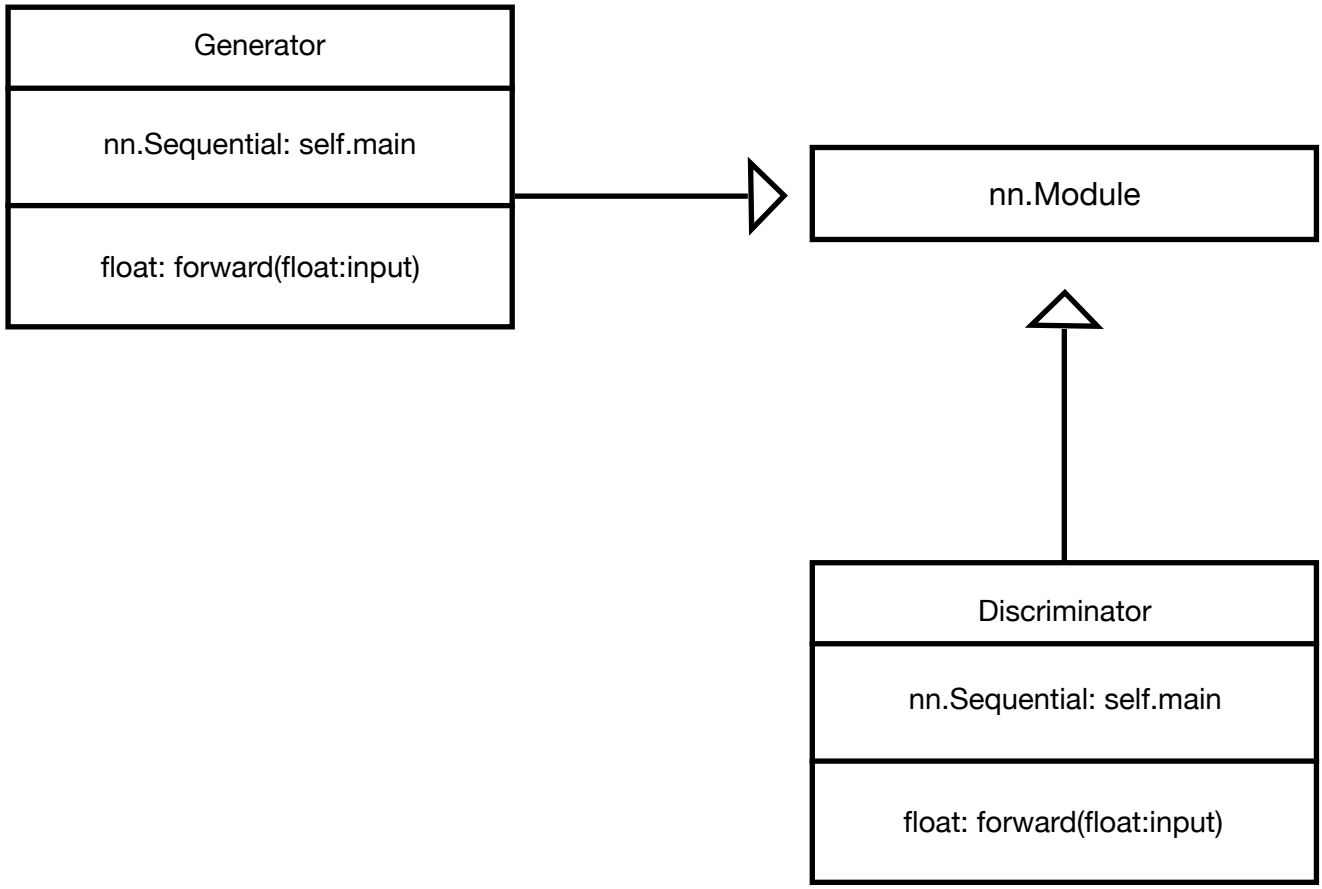
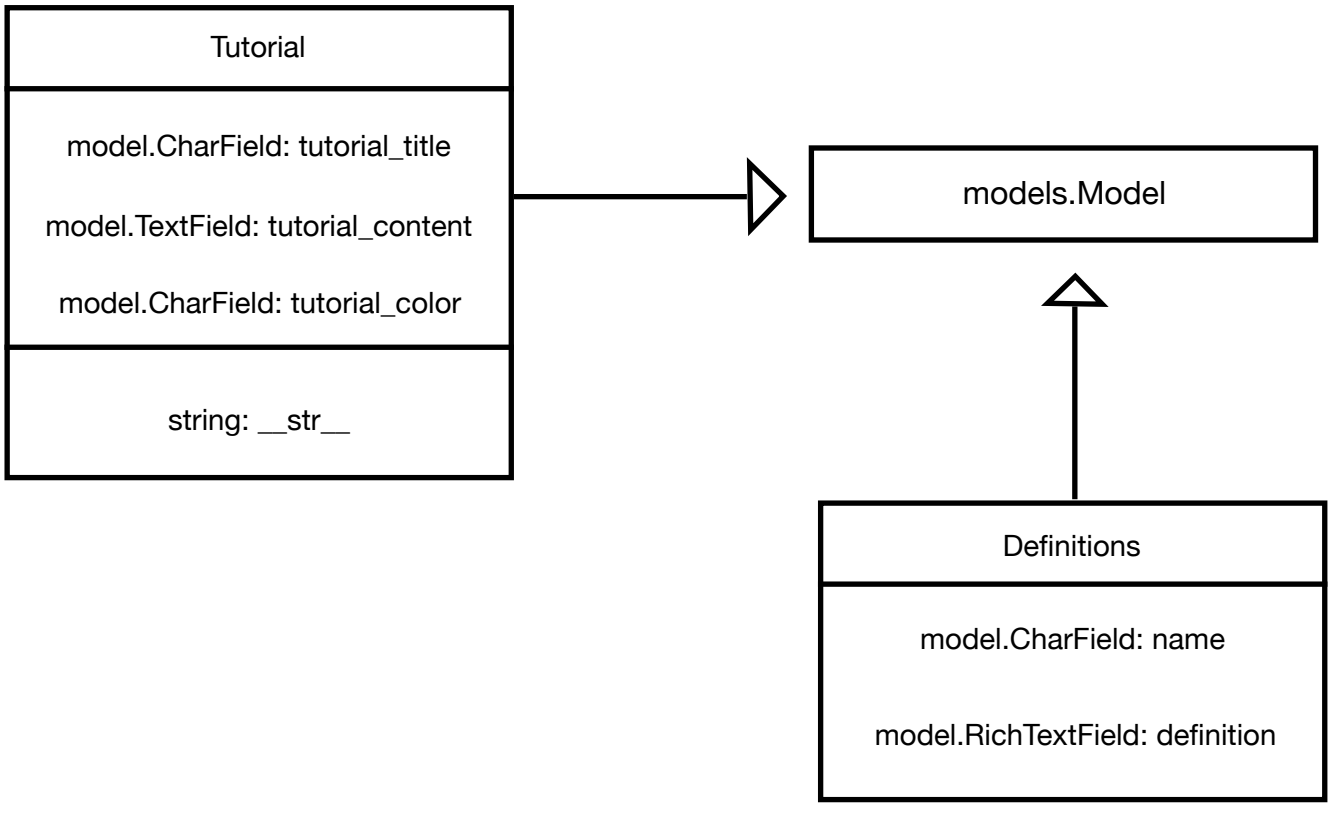


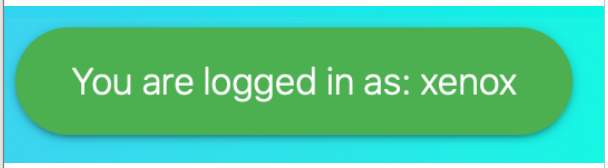
Image Generation:




Loop
through all the
images generated
Add the image in one array
Arrange all the images vertically to
display them
As the output is in range -1,1 convert
the array in range 0,255
Use opencv to change array to image
and save it
STOP


Generator
nn.Sequential: self.main
float: forward(float:input)
nn.Module
Discriminator
nn.Sequential: self.main
float: forward(float:input)


Tutorial
model.CharField: tutorial_title
model.TextField: tutorial_content
model.CharField: tutorial_color
string: __str__
models.Model
Definitions
model.CharField: name
model.RichTextField: definition

Action Test	Method of Testing and results
Check if the website shows all the elements in the home page	Run django server, copy the local address and use it in the website to open homepage and see if all the elements such as nav-bar, buttons are properly placed
Check if all the button in home page leads to desired pages	Click each button and check which page it leads to
Register feature actually saves the user's data such as username and password in database	Go to register page and register with a test name, head over to admin page provided by Django and check if the entry has Been properly made
Checking the Login page	Go to the login page and check if the user gets logged in when registered username and password is entered
User is able to logout	Clicking the logout button which appears when user is logged in. User should be presented with a message that he/she has been logged out successfully
All the correct alerts and messages are displayed at the right time	Clicking all the buttons associated with displaying any message to the user, buttons such as login, logout, each of the AI techniques. 
Object detection is working	Clicking the run button should turn on the camera and show all the objects detected in the frame
User is able to add their data in Face Recognition database	User should be able to enter their name and when they click configure, camera should turn on and store their images in the folder of their name.
Face Recognition model is training	Clicking the train button and checking whether the page is loading or not to notify that training is in progress. After the training is done it should display a image indicating that training is done

Action Test	Method of Testing and results
Face recogniser is able to recognise faces	Pressing the run button to check if the recogniser is able to recognise faces correctly
Music Generation	Generate button should start the process of generating music and once done it should display a message and load the webpage
Downloading Music	Pressing the download button to check if Music is downloaded in users computer
Image Generation	Clicking the generate button should generate images and save it in downloads folder of the user

Bibliography:-

Pre-Trained weights for all AI techniques were used.

Object Detection- https://pjreddie.com/darknet/yolo/

Face Recognition - open-cv Haar Cascades were used

Music Generation - https://github.com/Skuldur/Classical-Piano-Composer/blob/master/new_weights.hdf5

Image Generation - https://github.com/csinva/gan-pretrained-pytorch

Some parts of code was copied from internet and I have put comments for that in the code itself.

Sources such as stack-overflow and GitHub were used.