

CHAPTER-6

SYSTEM TESTING

6.1 Testing

In computer hardware and software development, testing is used at key checkpoints in the overall process to determine whether objectives are being met. For example, in software development, product objectives are sometimes tested by product user representatives.

Testing is a process rather than a single activity. This process starts from test planning then designing test cases, preparing for execution and evaluating status till the test closure.

6.2 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration.

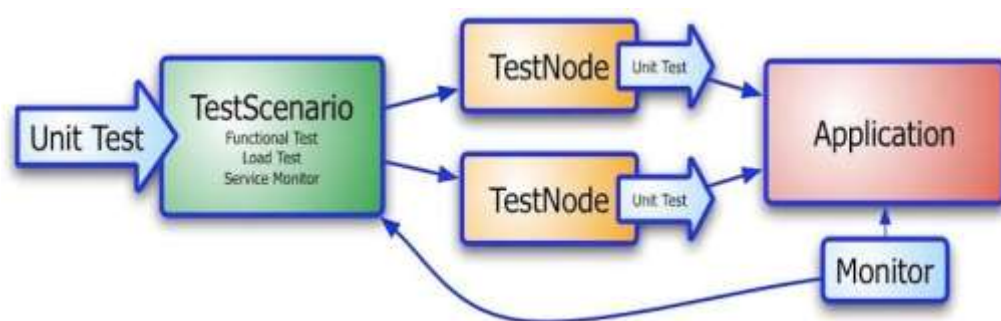


Figure 6.1 Unit testing

6.2.1 Unit testing techniques

- Black Box Testing
- White Box Testing
- Gray Box Testing

6.2.2 Applications

- Extreme programming
- Unit testing frameworks
- Language-level unit testing support

6.3 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.3.1 Two approaches

- Top-down approach
- Bottom-up approach

6.4 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- | | | |
|---------------|---|---|
| Valid Input | : | identified classes of valid input must be accepted. |
| Invalid Input | : | identified classes of invalid input must be rejected. |
| Functions | : | identified functions must be exercised. |
| Output | : | identified classes of application outputs must be |

exercised.

Systems : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.5 System testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying.

6.6 Black box testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

BLACK BOX TESTING APPROACH

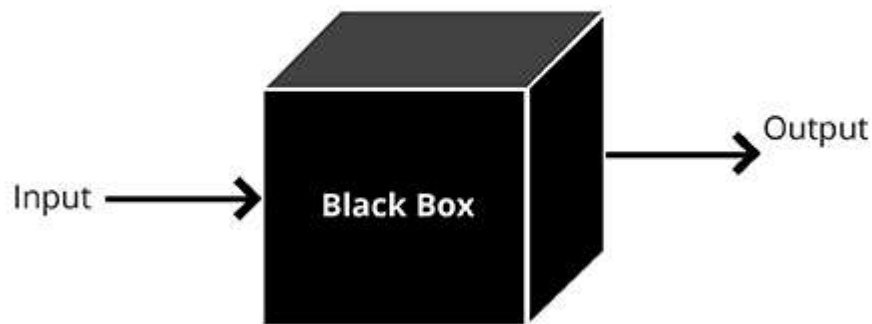


Figure 6.2 Black box testing

6.6.1 Security testing of automated attendance system

Security of the application very important to handle the database to reduce the fraudulent correction in the system. It is shown in Figure 6.3: Administration verification and Figure 6.4 Administration login confirmation.

The screenshot shows a web application window titled "AUTOMATED ATTENDANCE BY IMAGE PROCESSING USING IOT". Below the title is a section labeled "Admin Login". It contains two input fields: "User ID:" with the value "1" and "Password:" with masked characters "***". Below these fields are two buttons: "Login" and "Signin".

Figure 6.3 Administration verification

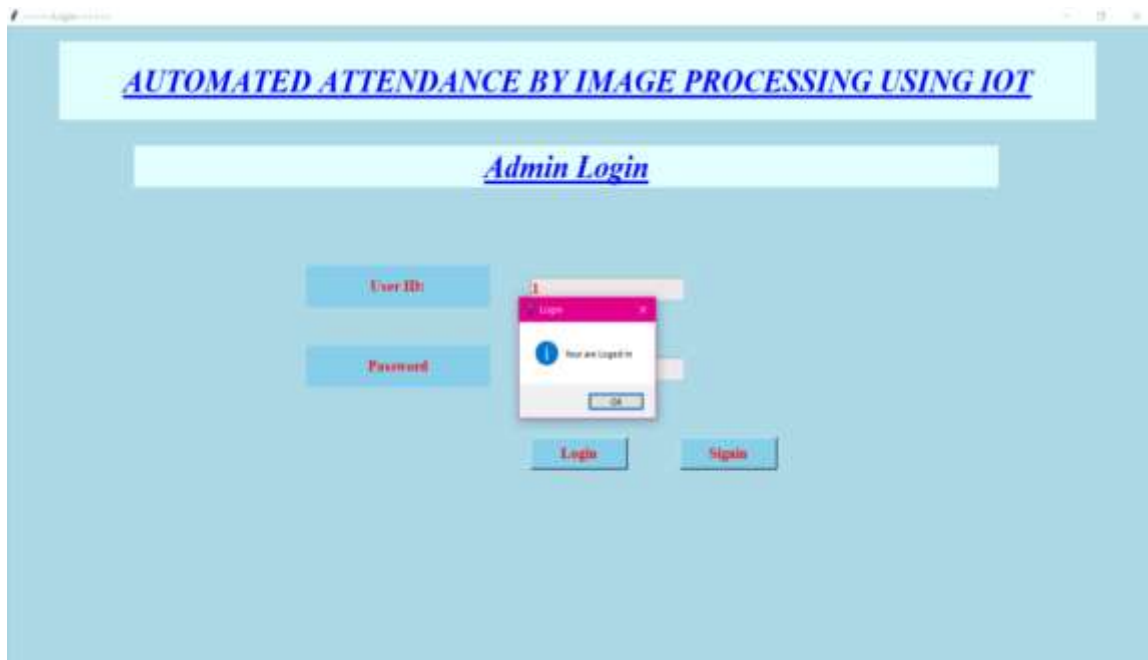


Figure 6.4 Administration login confirmation

6.6.2 Class room face detection testing

Detecting faces in the class room is important to notice the facial detection Figure 6.5: shows the face detection in the class room.

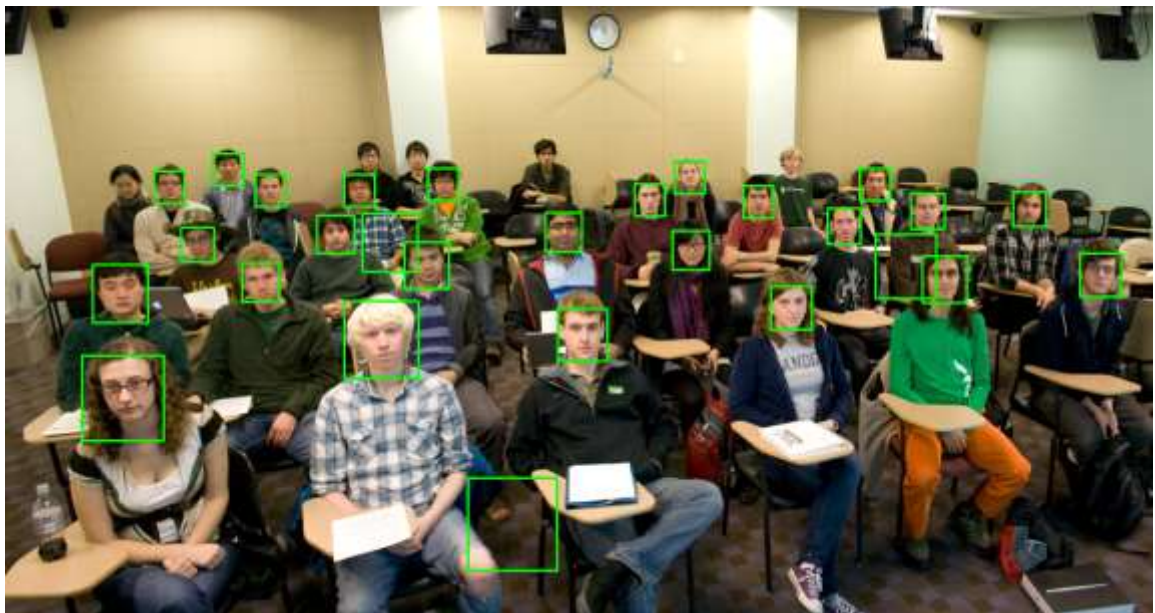


Figure 6.5: Class room face detection

6.6.3 Registration testing

If user ID already exist then it will alert the registration already exist in the GUI. It is shown in Figure 6.6 Registration verification



Figure 6.6 Registration verification

6.6.4 Face detection sample testing

Verification of student will be checked using Face detection sample. It is shown in the Figure 6.7 Sample face detection.



Figure 6.7: Sample face detection

6.7 White box testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level. It is shown in Figure 6.8 white box testing.

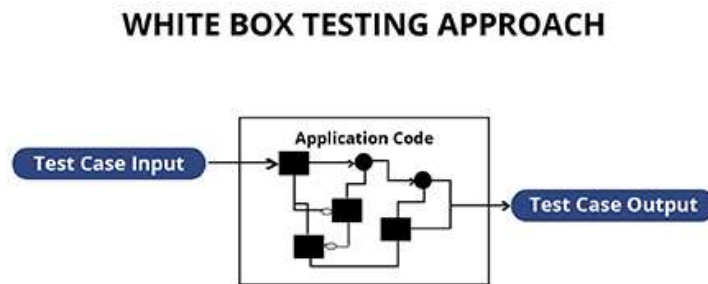


Figure 6.8 White box testing

Flow of the network will be identified using the database information. Figure 6.9 shows the database verification after registration

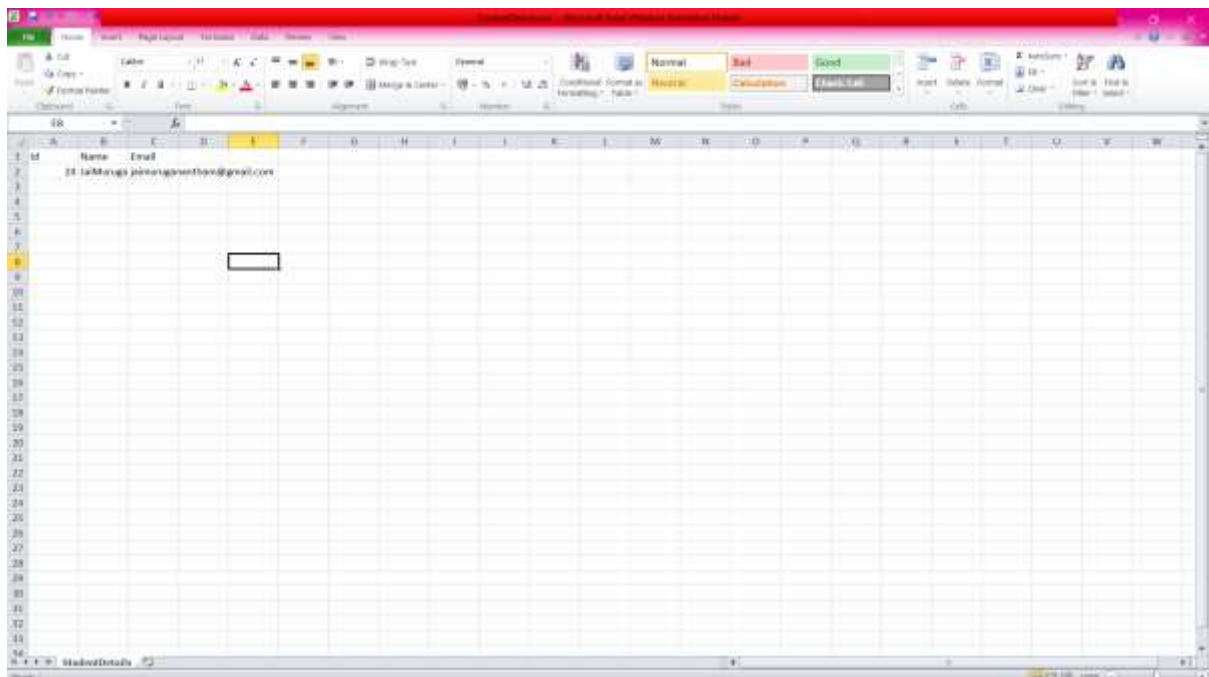


Figure 6.9 Database verification after registration

6.8 Unknown person details

Unknown persons details also updated to the database, recognize as “UN-KNOWN-PERSON”. It is shown in Figure 6.10.

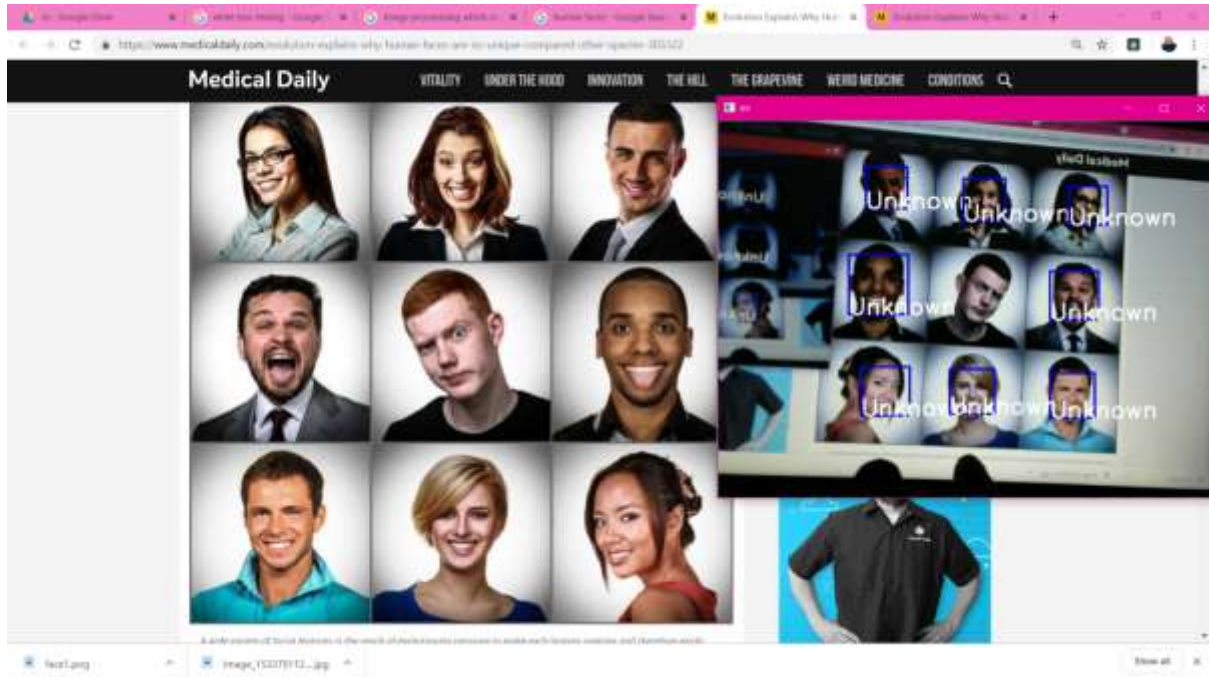


Figure 6.10 Unknown person details