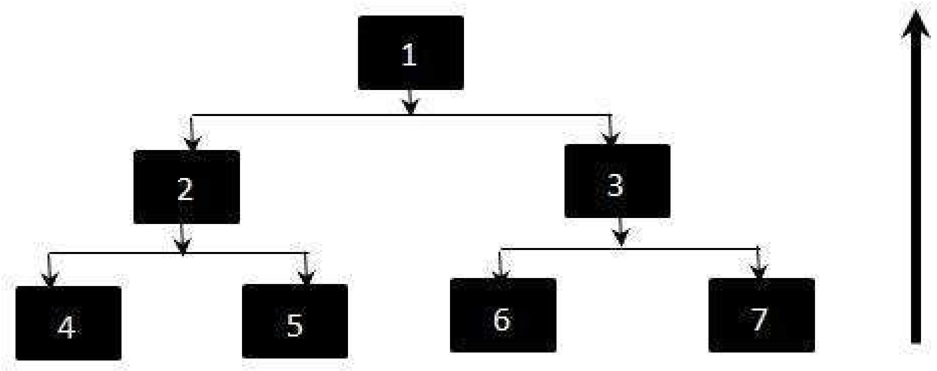
Testing Strategy

To fulfill our quality requirement, we will be performing two types of testing on our system: unit testing and integration testing. The overall testing strategy is **bottom up integration testing**. Following this strategy, our modules will divided into submodule and form a tree. Each modules at lower hierarchy is tested individually (unit testing) and then the modules that rely upon these modules are tested.



Following the diagram above, we will be testing 4, 5, 6 and 7 individually (unit testing). After testing all the modules at the bottom layer. We will start integration testing in pair with 4 and 2, 5 and 2 and so on. While testing 4 and 5, 2 are replaced with what is called a driver.

Types of Testing
In order to satisfy each of our quality requirement, we will be applying different types of testing to test our system.

Responsive Design Testing
Ensure all the elements in the user interface are displayed properly and doesn't affect the user's ability to understand and interact with it.
These elements include:

- Buttons
- Images
- Texts
- Modals
- Icons
- Forms

To test this, we will be using Chrome Developer Tools to simulate how every web page will display in 4 screen-width settings:

- Extra small devices Phones (<768px)
- Small devices Tablets (≥768px)
- Medium devices Desktops (≥992px)
- Large devices Desktops (≥1200px)

Data Integrity Testing

- Testing for accessibility of function:

  - Testing whether or not all user type can only access the functions that system allowed.
  - Eg: File management page can only accessed by students and supervisors but admin.
  - We will copy the url of the function (eg:/file/) and login as the user type that is not allowed to access the function to run the url. If the access is denied that means our testing is success.
- Testing for accessibility of data:
  - Testing whether or not user can access the data that related with them.
  - Eg: Student can view the file that upload by their supervisor or vice versa, but can't view the file uploaded by other students or supervisors.
  - To test this, we will copy the file url(eg:file/<file_id>/) of one of the user and run it using another user account. If the access is denied means our testing is success.

Security Testing

- Correct implementation of user login function
  - Username and password can login correctly
- Hashed and salted password will be checked if some users' password are same
  - Raw password is tested not to show in the user table which open the file by using SQLite
- Perspective of admin
  - No password is allow to show to anyone
- The functionality of password encryption

Functionality Testing

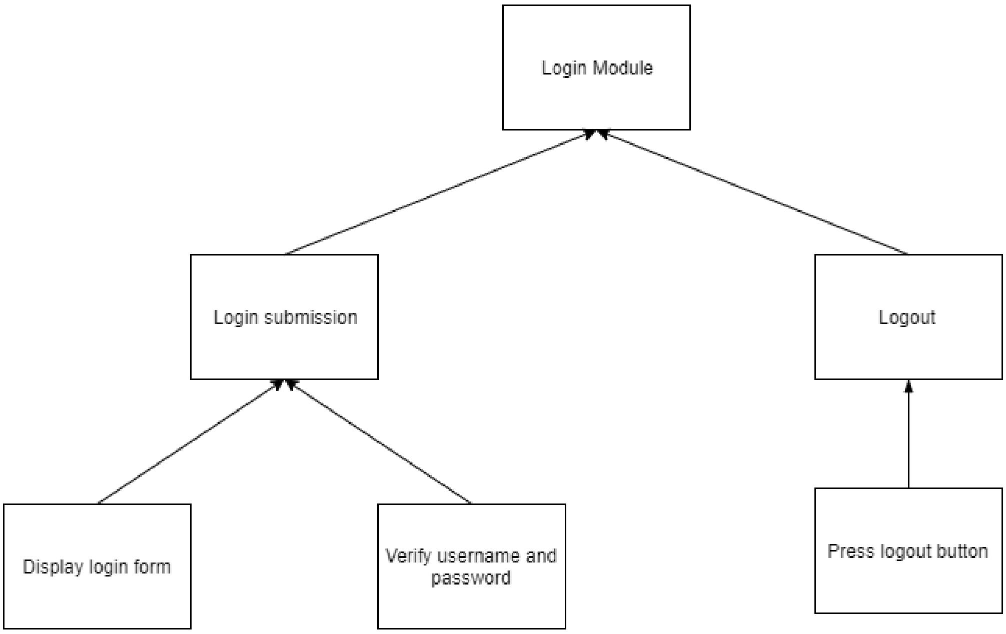
- Forms checking
  - Make sure that all forms are fill with correct input (e.g. file uploaded must be in a specific format only), if the forms have been input with invalid input, error message will pop up prompting user to fill the forms with the correct details.
  - Scenario : If user decided to upload/change their profile picture, user must only use file format such as PNG, JPEG and GIF. If user use other file format such as PDF, the system will notify the user to use on PNG, JPEG and GIF for their file input.
- Test all links
  - Check links to ensure user is directed to the section/page that it was intended.

Portability Testing

- To ensure user can access by multiple type of device and different web browser through the website.
  The device included:
  - Smartphone
  - Desktop
  - Laptop
  - Tablet

- The browser that will be tested on every device (if applicable) includes:
  - Google Chrome
  - Mozilla Firefox
  - Internet Explorer 11
  - Microsoft Edge
  - Safari
- To this test, we will use multiple device to access the website and ensure the output is works fine and display correctly as we expected in different type of device.

Example of Bottom Up Integration Testing



In our login module, the module itself is subdivided into sub-modules, namely Login Submission and Logout. So in order to test our Login Submission modules, we will have to test the sub-module's sub-modules, Display Login Form and Verify Username and Password.

We will start by testing the Display Login Form module. We will have to make sure the the labels for the form are displayed properly, and the forms work properly, eg. the username form will show everything the user types, while password form will hide the characters with '*' character.

Now to test Verify Username and Password, we could've test this module by inserting the password and username in variables and pass it to the login function to test it.

Now that both sub-modules of Login Submissions are working, we now can test Login Submission itself. To test this module we will use the form from the Display Login Form submodule and press the Login button. The input from the login form will be passed to the Verify Username and Password module and user will be logged or not logged in depending on the inputs.

However, we still have to check the Logout sub-module before we declare the Login module tested. We have make sure the Logout button is displayed to the user. If the user clicked the Logout button and is redirect to the Login page, it’s considered check. To test if the Logout truly works, we can try to access the homepage again and see if we got redirected back to login page. If it does, it means the Login sub-module is working.

With all the sub-modules tested, we now can declare the Login module tested.