1) [10 Points] Testing Web Applications

a) Gray box approach is the most appropriate approach in testing web applications. Briefly explain the difference between gray box approach with respect to black box and white box testing.

In Glass-Box testing or White-Box testing, the internal code is visible to us for testing. In Black-Box testing, one does not know how programs are implemented, it is tested using inputs, outputs, and then environment and server configurations. Functional testing (Black-Box) testing uses specification while structural testing (White-Box) uses the program when designing test cases. Gray-Box testing sits in between White-Box and Black-Box testing. Gray-Box testing is partially visible, we are aware of how the system is designed and what kind of technologies are used. This is so, so that a user can come-up with a proper set of test data to test the implementation. In Gray-Box testing we consider testing outcome on the end user, system specific technical knowledge, operating environment, application design in the context of interoperability of system components. Therefore, Gray-Box testing is the most appropriate testing technique for web application testing. (Using lecture from 11/10/2020 and PerspectiveOnSoftwareTesting\_Chapter1.pptx).

b) List and describe two challenges in testing client-server web applications that will not arise in non-web applications?

There are a few challenges with testing web applications, challenges such as web applications being loosely coupled, and data consistency (data pertaining to one application is scattered among different devices).

Loosely coupled web applications are likely to be difficult to examine due to having loose/independent components, i.e. different technologies can be used in implementing the client component, server component.

Like so, data consistency is a challenge because some components might be on the server side, some might be on the server side, and some might be in the database server. Now, this is likely to be a challenge because if some data is not upgraded accurately, another client might that data and it will not be up to date. Ex: client A purchases a ticket for flight, and data within the system does not get updated with a locking mechanism or similar computational implementation that shows the ticket has been reserved for client A. Another client, client B tries to purchase the same purchased flight ticket, data consistency issue will rise in this scenario. (Using WebApplicationTesting\_Lecture2.pptx)

2. [15] Designing Considerations and Testing Client Server Web Applications: For this problem, consider the online community newspaper application description given below.

Online community newspaper: This online community newspaper application is for a “close-knit” community such as a residential neighborhood, school club to maintain interested topics to that community. Each member in the community need to register in the community newspaper site and upon registration members can login using the username and the password. Each member can perform following tasks

• Post a news article

• Comment on a news article

• Owners of the post (person posted the article) can delete and make amendments to news articles

In addition to regular community members, there is admin who

• Approves member registration requests

• Monitors the newspaper and hide/delete inappropriate news articles/comments.

Assume that this application has been implemented using PHP, MySQL and deployed in an Apache server (Similar to the addressbook example discussed in the class).

1. Draw suitable software architecture for the above community newspaper application that complies with better design considerations discussed in the class. Identify functionalities you would put in each tier. Justify your design. (Consider client-server web application architecture.

Diagram

Description automatically generated

I would design my web application system this way due to the following reasons:

- This design has proper amount of layers, and it is easier for testing. User components and generations can be combined into one program. I.e. like shown in the design above(client and newsArticles.php) or (Admin with approveRegisteration.php). Layers will call each other’s interfaces/programs for information through this design, rather than exposing the manageArticles.php into the dbConnection, which will eliminate some security risks.

- To prevent malicious activity on this application, I would connect the database in the database.php layer instead of the manageArticles.php layers. For instance, if dbConnect.php gets connected in the manageArticles.php, a hacker can try to access the manageArticles.php and modify the system, instead of using the operation queries available like the system administrator would or the client would. Also, usernames and passwords of clients/admin may be in the dbConnect.php, so therefore, that would be a security issue if anyone can access it, this kind of design is open for SQL injections malicious activities.

database.php is the most important module within the system since it deals with the back end data storage, the query operations/components can access this module through public APIs only and there are verification methods that assist them(helper.php)

- helper.php function can sanitize the queries operations that helps ensure malicious code segments or malicious queries are not entered by the user. For instance, if we know the username, password of a user(u = dela, p = “222=222”) within the system, and that user tries to login, the helper can ensure by checking the password as follow to verify: “222=222”, data sent by the sanitizer will always be true for server side execution. Once it is sanitized, the system ensures that there are no malicious queries.

b) Briefly describe your approach in testing this application.

Tests can be defined for each layer within the system, for example:

Test for:

- Client layer(user-interface)

- Administrator layer (user-interface/admin-interface(approve registration/hide inappropriate comments))

- Server layer(Web application)

- Server layer(data transformation)

- Server layer(data management)

- Database layer(data access)

- Database

Sandwich approach can be used to test this application system since it is crucial for the admin module to perform tasks in the system and since the database.php module is the most important module layer that deals with the back end data storage.

One can generate test cases for all the operations within the database, to ensure all the implementations are correct. Then, one can test the administrator layer to ensure they can accurately and securely give registration approvals to clients depending on user registration forms and through user-input test submissions. Then the in-between operation queries can be tested by generative tests for each query available to ensure they work properly and check for harm against malicious activities.

(Reference: using Lecture 11/10/2020)

c) Name tools /frameworks that you would use in testing this application (name and describe the usage of at least three tools).

- DB Unit: is one tool in the unit family that tests database operations. This can be used for testing the database layer.

- Selenium IDE: plugin for the browser testing. Can be used for client/administrator user interactions testing.

- Selenium WebDriver: APIs that are open source, can be used to generate automated tests for web applications. Simulate user input in the manageArticles.php layer for testing.

(Reference: using Lecture 11/10/2020)