EXPERIMENT – 4

TESTING

APARTMENT MANAGEMENT SYSTEM

Aim

To perform black box testing and white box testing along with its testing strategies and to execute hands on experience on unit testing.

Software testing

Testing software is an important part of the development life cycle of a software. Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is defect free. According to the ANSI/IEEE 1059 standard, the definition of testing is the process of analyzing a software item, to detect the differences between existing and required to evaluate the features of the software item.

The purpose of testing is to verify and validate a software and to find the defects present in a software. The purpose of finding those problems is to get them fixed.

Verification is the checking or we can say the testing of software for consistency and conformance by evaluating the results against pre-specified requirements.

Validation looks at the systems correctness, i.e. the process of checking that what has been specified is what the user actually wanted.

Defect is a variance between the expected and actual result. The defect's ultimate source may be traced to a fault introduced in the specification, design, or development (coding) phases.

PROJECT DESCRIPTION:

Product Perspective

Hardware Interface

Hard disk: The database connectivity requires a hardware configuration that is on-line. This makes it necessary to have a fast database system running on high rpm hard disk permitting complete data redundancy and back-up systems to support the primary goal of reliability. The system must interface with the standard output device, keyboard and mouse to interact with this software.

Software Interface

Front end : HTML5, CSS, PHP

Back end : PHP, Apache, WAMP server With MYSQL

Memory Constraints

No specific constraints on memory.

Operations

The software allows two modes of operations

- Enquire about the availability of blocks for rent in apartment.
- By extracting the username and password the software allows the user to make the payments.

Product Functions

Enquire about the availability of apartments. Search the availability of apartments by new user and the existing can make the payments. The software validates the authentic user by extracting their user name and password. After the validation of the user software allows the admin to rent the house for tenants after the credentials verification.

User characteristics :

The intended users of this software need not to have specific knowledge as to what is the internal operation of the system. Thus the end user is at a high level of abstraction that allows easier, faster operation and reduces the knowledge requirement of end user .The Product is absolutely user friendly, so the intended users can be the naive users. The product does not expect the user to possess any technical background. Any person who knows to operate the mouse and the keyboard can successfully use this product.

Constraints

The user can login only using their unique username and password. In case, the user has forgotten the password ,it can be retrieved through their registered email.

Types of Software Testing

- UNIT TESTING
- INTEGRATION TESTING
- SYSTEM TESTING

1)Unit Testing:

Unit testing is done at the lowest level. It tests the basic unit of software that is the smallest testable piece of software. The individual component or unit of a program is tested in unit testing. Unit testing are of two types.

1. Black Box Testing
2. White Box Testing

TEST SUITE:

Module 1: Verify "User Login" functionality

#	**TS1**
Title	**Verify "User Login" functionality**
Description	**To test the different scenarios that might arise while an user is trying to login**

#	Summary	Dependency	Pre-condition	Post-condition	Execution Steps	Expected Output
TC1	Verify that user already registered with the AMS is able to login with correct user ID and password		CUSTOMER ID *1234* is a registered user of AMS; user's password is *this_is_password*	User is logged in	1. Type in customer id as *1234* 2. Type in password *this_is_password* 3. Click on the 'Login' button	"Home" page for the user is displayed
TC2	Verify that an unregistered user of AMS is unable to login		customerID *1566XX* is not a registered user of LIS	User is not logged in	1. Type in customer ID as *1566xx* 2. Type in password *whatever* 3. Click on the 'Login' button	The "Login" dialog is shown with a *error invalid user credentials*

#	TS1
Title	Verify "User Login" functionality
Description	To test the different scenarios that might arise while an user is trying to login

#	Summary	Dependency	Pre-condition	Post-condition	Execution Steps	Expected Output
TC3	Verify that user already registered with the AMS is unable to login with incorrect password		customer ID is a registered user of AMS user's password is *this_is_password*	User is not logged in	*1.* Type in customer ID as *4567* *2.* Type in password *whatever* 3. Click on the 'Login' button	The "Login" dialog is shown with a *error invalid user credentials*
TC4	Verify that user already registered with the AMS is unable to login with incorrect password given twice consecutively	TC3	This test case is executed after execution of TC3 before executing any other test case	User is not logged in	*1.* Type in employee ID as *4567* *2.* Type in password *whatever2* 3. Click on the 'Login' button	The "Login" dialog is shown with a *error invalid user credentials*
TC5	Verify that user already registered with the AMS is unable to login with incorrect password given thrice consecutively	TC4	This test case is executed after execution of TC4 before executing any other test case	User is not logged in	*1.* Type in customer ID as *4567* 2. Type in password *whatever3* 3. Click on the 'Login' button	The "Login" dialog is shown with a *error invalid user credentials*

Module 2: Verify "search according to your needs" functionality

2#	TS2
Title	**Verify "search according to your needs" functionality**
Description	**To test the different scenarios that might arise while a new tenant is trying to searchfor an apartment**

#	Summary	Dependency	Pre-condition	Post-condition	Execution Steps	Expected Output
TC1	Select the desired block from the apartment select filter/list		Desired product is 2bhk.If we select 2bhk it displaysthe available apartments.	Varieties of blocks are displayed.	4. Select the variety from the list. 5. Click on the 'GO' button	"RESULT" page for thevariety is displayed
TC2	Search without selecting any desired module from the filter/list.		Displays all the apartments available.	Varieties of all blocks in apartmentsare displayed	4. Without selecting any variet. 5. Click on the 'GO' button	"RESULT" page for variety of all products is displayed.
TC3	Search for selected block details.	TC2	This test case is executed after execution of TC2 before executing any other test case	It displays the apartment image, block sub variety, rent and button to request the block and review.	4. Click the apartment image.	Displayed the result page
TC4	Verify that the user request for the apartment	TC3	This test case is executed after execution of TC3 before executing any other test case	If we request, it displays the details of admin approval or rejection. If it is approved by admin, status is changed to apartment booked for rent and after the payment of initial it is removed from the available.	4. Click 'REQUEST' button 5. Click on 'submit' button	1)If it is approved by admin, new user can make the payment and ensure it 2) If we request for the block thenit is added to waiting list of admin 3) If approved it is displayed as booked for rent

Module 3: Verify " update apartment and tenant" functionality

#	TS3
Title	**Verify " upload apartment and tenant" " functionality**
Description	**To test the different scenarios that might arise while an tenant/admin is trying to update**

#	Summary	Dependency	Pre-condition	Post-condition	Execution Steps	Expected Output
TC 1	Admin wants to update the block details		User should be registered in theAMS website.	Apartment updated successfully.	6. Click update apartment 7. Click block details and Update the apartment 8. Type the block name 9. Type the price of the block in apartment 10. Enter submit button	" Digital apartment page isdisplayed"
TC2	Admin updates the block details without the necessary information		User should registered in theAMS website.	Apartment is not updated	6. Without typing necessary information 2. Enter submit button	The result page shows error you cannot update the apartment.
TC 3	Admin updates the apartment withoutchoosing the image file		User should be registered in theAMS website	Apartment is not updated	5. Without Updating the apartment image 6. Click the submit button	The result page shows error you cannot update the apartment
TC4	Admin updates the apartment without typing the priceof the block		User should be registered in theAMS website	Apartment is not updated	7. Without typing the price 2.Click the submit button	. The result page shows error you cannot update the apartment

Module 4: Verify " PAYMENT" functionality

#	TS4
Title	Verify " **PAYMENT** " " functionality
Description	To test the different scenarios that might arise while an user is trying during the payment process

#	Summary	Dependency	Pre-condition	Post-condition	Execution Steps	Expected Output
TC 1	Tenant wants to make a payment from list of bills allotted to them for their rental apartment		Tenant should registered inthe AMS website.	Payment done successfully	11. Select the payment from the given category 12. Click pay for the lit of bills according to your wish. 13. 3. Fill the transaction details 14. 4.enter confirm	" Result page displayed with SUCCESS ..payment done successfully!
TC2	Tenant wants to make the payment without filling the necessary fields in the transaction details	TC1	Tenant should registered in theAMS website.	Oops.Payment failed	7. Without typing necessary field in the transaction details 2. Enter confirm	The result page show error you have not make the payment.

BLACK BOX TESTING:

This is also known as functional testing , where the test cases are designed based on input output values only. There are many types of Black Box Testing but following are the prominent ones.

Equivalence class partitioning: In this approach, the domain of input values to a program is divided into a set of equivalence classes. e.g. Consider a software program that computes whether an integer numberis even or not that is in the range of 0 to 10. Determine the equivalence class test suite. There are three equivalence classes for this program. - The set of negative integer - The integers in the range 0 to 10 - The integer larger than 10.

Boundary value analysis : In this approach, while designing the test cases, the values at boundaries of different equivalence classes are taken into consideration. eg. In the above given example as in equivalence class partitioning, a boundary values based test suite is { 0, -1, 10, 11

2.WHITE BOX TESTING:

It is also known as structural testing. In this testing, test cases are designed on the basis of examination of the code. This testing is performed based on the knowledge of how the system is implemented. It includes analyzing data flow, control flow, information flow, coding practices, exception and error handling within the system, to test the intended and unintended software behavior. White box testing can be performed to validate whether code implementation follows intended design, to validate implemented security functionality, and to uncover an exploitable vulnerabilities. This testing requires access to the source code. Though white box testing can be performed any time in the lifecycle after the code is developed, but it is a good practice to performwhite box testing during the unit testing phase.

2) Integration Testing

Integration testing is performed when two or more tested units are combined into a larger structure. The main objective of this testing is to check whether the different modules of a program interface with each other properly or not. This testing is mainly of two types:

1) Top-down approach

2)Bottom-up approach

In bottom-up approach, each subsystem is tested separately and then the full system is tested. But the top-down integration testing starts with the main routine and one or two subordinate routines in the system. After the top-level ‘skeleton’ has been tested, the immediately subroutines of the ‘skeleton’ are combined with it and tested.

3) System Testing

Alpha testing is done by the developers who develop the software. This testing is also done by the client or an outsider with the presence of developer or tester.

Beta testing is done with very few end users before the delivery, where the change requests are fixed, if the user gives any feedback or reports any type of defect.

User Acceptance testing is also another level of the system testing process where the system is tested for acceptability. This test evaluates the system's compliance with the client requirements and assess whether it is acceptable for software delivery

An error correction may introduce new errors. Therefore, after every round of error-fixing, another testing is carried out, i.e. called regression testing. Regression testing does not belong to unit testing, integration testing, or system testing instead, it is a separate dimension to these three forms of testing.

SUBMITTED BY
Arifa shirreen .M – 1817106
Bhuvaneswari. S – 1817109
Deepika. S– 1817111
Dhanam. S – 1817112
B.E CSE III YEAR