ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 7, January 2013

Software Testing Process Using a Payroll System as a Case Study

Akinboyewa Nelson E.,Johnson O. Victor

Abstract*—Software testing involves running an implementation of the software with test data. One can examine the outputs of the software and its operational behavior to check that it is performing as required. But meanwhile testing is an expensive, difficult and time-consuming stage; however it is a dynamic technique of software verification and validation. This paper therefore takes a look at payroll system design with real-life test data and generates test cases for each of the tested modules using a black box technique. Each of the test scope and test deliverables was analyzed and the test cases result generated and tabulated.

***Index Terms*—NEMAK, Blackbox Testing, Whitebox Testing, Payroll, Akinboyewa Nelson, Johnson Victor,**

I. INTRODUCTION

Although the software development life cycle can be divided in different ways, it usually includes the following phases, which application developers can repeat iteratively: initialization, specification and design, implementation (coding), testing, deployment, and decommissioning [1]. A software test is an activity in which a system or a component is executed under specified conditions, the results observed, and an evaluation made on some aspect of the system or component [2]. "Reference [3] defines software testing as an empirical, technical, investigation conducted to provide stakeholders with information about quality of the product or service under test."

Test planning is the critical part of an effective software testing. It is the start point of an effective testing. Testing should not be considered merely as an evaluative process. It is also a process of exploring the meaning and implications of requirements [2]. A good testing plan should:

- Defines the objectives for each test phase.
- Establishes schedules and responsibilities for each activity.
- Builds the procedures and standards to be followed for planning and conducting the test reporting of the test results and,
- Sets the criteria for test completion as well as for the success of each test. After developing the test plan, the test cases should be created.

To achieve the goal of finding as many errors as possible, we must face the fact that the programmers who produce programs also produce bugs. That is a consequence of humanity. The programmers are often biased by their creative work, so they have difficulties finding out the errors themselves. The solution to this problem is to separate testing from program design and implementation.

This plan defined the general approach of the testing and defines the scope of the features to be tested and what will be excluded from the testing. Furthermore, we therefore discuss the test design and execution deliverables.

II. METHODOLOGY

We provided an overview of the technical testing strategy, the test processes and decided that the focus of the first testing of the initial prototype should be on the input/output handling of the system. There are two main test approaches: *Black box testing* and *White box testing*. For the implementation part of this research paper, we use Black box testing, to test the prototype against the end user's requirements in the GUI coding phase. This refers to the analysis of program execution from the external point of view [4][5]. In short it consists of comparing the software execution outcome with the expected result [6]. "Reference [3] highlighted ten dominating techniques to black box software testing in which Telenova station was used as test case." Black box testing concentrate on the *application system testing* in line with the view that a Contract Driven Development can be used a mechanism of extracting test cases from failure-producing runs that the programmers triggers for a unit testing approach [7]. Moreover application scope is used to test user requirements using extended use cases.

III. TEST ITEMS AND TECHNIQUES

The system to be tested is the requirement documentation of the first prototype of the NEMAK Payroll System. This prototype focuses on the creation and maintenance of timecards and the initialization of the run payroll sequence.

A. Application Scope Testing

In order to be able to effectively test a system scope you require a testable system specification. Testing an application system has three main goals:

1) To reveal bugs that are present only at system scope
2) To demonstrate that the system under test implements all required capabilities
3) To provide answer the question:"Is the system finished?"
4) To uncover any usability issues

These goals cannot be achieved by any other software engineering technique. Walkthroughs reviews and inspections on models and code do not exercise the actual system. Therefore Application scope testing is necessary. Three patterns are available that can be used to create a complete test suite:

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 7, January 2013

B. Extended Use Case Test

This involves the designs of test cases to exercise all relationships implied by a use case.

C. Allocate Test by Profile

This develops a quantity of test cases in proportion to the operational frequency of the use cases.

We use the extended use case test since the emphasis here is on testing the system requirements and functionality, as defined in the requirement document. Use cases reflect the inputs/outputs from the user's point-of-view [8] and are, therefore, focused on essential capabilities that will determine the success or failure of the System. Test cases will be based on the extended use cases that were defined during the requirement process [9].

D. Integration Testing

Software systems are built with components that must inter-operate. Three basic kinds of testing are needed to show that the components are minimally inter-operative: Test on individual components, test on the system resulting from the federation of components, and test of components interoperation. Integration testing is the search for component faults that cause inter-component failures. There are four focuses of integration testing:

1) **Method focus**, which focuses on class level and intra-class messages.

2) **Class focus**, which focuses on cluster level and inter-class messages.

3) **Cluster focus**, which focuses on subsystem level and inter-class/inter-packages messages.

4) **Subsystem focus**, which focuses on System level and inter-process communication and remote procedure calls.

Here we chose to focus on method, cluster and subsystem testing. Class focus is tedious and time consuming since we have to isolate each class by itself. Therefore, class testing is not a favorable one at this moment in time.

E. Method Scope Testing

Although a method cannot be tested apart from its class, test cases are applied by sending messages to methods. The design of method scope tests is fundamental to testing object-oriented systems. In some situations, a method scope test design becomes necessary when developing tests for individual methods. There are four method scope test patterns available:

1) **Category-Partition**, which can be used on any method or testable function

2) **Combinational Function test**, which is appropriate for methods that implement complex algorithms, business rules or similar case-based logic.

3) **Recursive Function test**, which focuses on functions that calls themselves

4) **Polymorphic message test**, which is the pattern used for a polymorphic server.

Since the methods that will be tested do not contain any coding but, few line of algorithms where needed for the purpose of this paper, no distribution will be considered. However, we chose Category-Partition based approach to our method scope testing.

F. Cluster Scope Testing

At cluster level, the system is the cluster under test. The components to be integrated are the objects used by the cluster under test [10]. The available test patterns are:

1) **Bottom-up integration**, Interleave component and integration testing by following usage dependencies

2) **Top-down integration**, Interleave component and integration testing by following application control hierarchy

3) **Collaboration integration**, Choose the order of integration according to collaborations and their dependencies, by testing one collaboration at a time.

4) **Big Bang integration** attempt to demonstrate system stability by testing all components at the same time

5) **Controlled Exception integration** can be used if the cluster under test catches exceptions.

In this scope we decided to use the collaboration integration. The cluster scope test pattern will be applied on the collaboration diagram of the use case. The collaboration diagram is used to specify the implementation of the use case and it depicts the participating classes in a design pattern. Collaboration diagrams represent a cluster of classes involved in the sequence of interactions relating to a single use case. Testing of collaboration diagram can be accomplished with:

1) **Class association test** shows how to design the test suite that will exercise the associations defined in a class or object model.

2) **Controlled exception test** shows how to design a test suite that will exercise exception handling.

3) **Round trip scenario test** shows how to design a test that will cover all event-response paths in a sequence or collaboration diagram.

4) **Mode machine test** tells us how to model the aggregate state behavior of a cluster and how to develop a state-based test suite.

On this level we decided to use the Round-trip Scenario testing on the collaboration diagrams since we are interested in event-response paths that the cluster uses to call its subparts. Collaborations to be tested are Change Period and Get Employee Periods.

G. Subsystem Scope Testing

A subsystem is any testable collection of classes, objects, components and modules. A subsystem is executable and testable as a whole and has parts that can be tested in isolation. The interest for testing at this scope can arise from many reasons. The major one is that it is a precondition for testing at system level (application scope testing).

A subsystem test normally answers two key questions:

1) What features and capabilities should be tested?

2) How should the test plan and test suite be organized?

The key problem of subsystem testing is to develop a testable model.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 7, January 2013

IV. TEST DELIVERABLES AND ANALYSIS

We describe below the overall test plan, individual test design and test cases specifications and test procedures for each of the test items as described above

A. Application Scope Test Suite

This section focuses on the coverage of input/output relationships that occur between the user and the system. In order to develop test cases the extended use cases defined during the requirement engineering process.

1) Entry Criteria

Testing may begin when following criteria's are met: Extended Use Cases have been developed and verified Extended Use Case Test Cases have been developed and verified The SUT (System under Test) has passed the integration test suite. This shows that the system can meet the minimal requirements for a functioning system.

2) Exit Criteria

The testing is considered finished when All Test cases have been tested successfully at least once All Use Cases have been tested at least once

3) Automation

Test suite will be manually executed so no automation is applicable for the purpose of this project.

4) Pass criteria

A test case scenario is considered "passed" when: All conditions of the test case have been met The System response equals what is defined in expected outcome

5) Fail Criteria

A test case scenario is considered "failed" when: The test case could not be run All conditions were not met The system response is not equal to what was defined in expected outcome

B. Method Scope Test Suite

This test suite tests the input/output handling of the system at a method level of the system under test.

1) Entry Criteria

Testing may begin when following criteria are met: The classes have reached the operative threshold.

2) Exit Criteria

The testing is considered finished when every combination of the method's choices have been tested at least once All branches of the method should have been executed at least once.

3) Automation

None

4) Pass criteria

A test case scenario is considered "passed" when: All conditions of the test case have been met The System response equals what is defined in expected outcome

5) Fail Criteria

A test case scenario is considered "failed" when: The test case could not be run All conditions were not met The system response is not equal to what was defined in expected outcome

V. TEST CASES ANALYSIS AND RESULTS

A. Login:

1) Description:

These test case scenarios test the login function of the Run Payroll System. The requirements are that if the user specifies a valid user ID and password he will be logged in to the system. Otherwise he will be refused access to the system. The test result is presented below (see Table I).

2) Procedure specifications: In order for the Test Cases to be initialized, the tester must have installed the Run Payroll System on his personal Computer, executed the Run Payroll system by clicking on the run payroll icon and be logged in to the Payroll system with username/password .

B. Logout:

1) Description:

These test case scenarios test the logout function of the Run Payroll System. The requirements are that that if the User logs out and his timecard has been changed it will be saved. If the user has created a report he will be asked to save it. The test result is presented below (see Table II).

2) Procedure specifications: as in A (2) above.

C. Maintain Timecard

1) Description:

These test case scenarios test the Maintain timecard function of the Run Payroll System. The requirements are that the user should be able to select a period, choose a project charge number and enter how many hours he has worked on that project. The test result is presented below (see Table III).

2) Procedure specifications: as in A (2) above.

D. Change Password

1) Description:

These test case scenarios test the Change password function of the Run Payroll System. The requirements are the user should be able to change password to another than the default. The test result is presented below (see Table IV).

2) Procedure specifications: as in A (2) above.

E. Create Employee Report

1) Description:

These test case scenarios test the Create employee report function of the Run Payroll System. The requirements are the employee should be able to generate different kinds of reports and be able to save them. The test result is presented below (see Table V).

2) *Procedure specifications:* as in A (2) above.

Table I: Showing the Login test cases, test scenario and the expected outcome

Test Case	Test Scenario	Expected Outcome
TC.1.1. Valid Username and Password	TC.1.2. Enter "akinboyewa" in User field TC.1.3. Enter "akinboyewa" in Password field TC.1.4. Click on "Log in"	TC.1.5. Login successful, User is logged in to the system.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 7, January 2013

TC.1.6. Wrong password	TC.1.7. Enter "akinboyewa" in User field TC.1.8. Enter "akinbo" in Password field TC.1.9. Click on "Log in"	TC.1.10. Error TC.1.11. Login failure TC.1.12. User asked to login again.
TC.1.13. Wrong Username	TC.1.14. Enter "akinboye" in User field TC.1.15. Enter "akinboyewa" in Password field TC.1.16. Click on "Log in"	TC.1.17. Error TC.1.18. Login failure TC.1.19. User asked to login again.

Table II: Showing the Logout test cases, test scenario and the Expected outcome

Test Case	Test Scenario	Expected Outcome
TC.1.20. Log out with Timecard Changed.	TC.1.21. Open Maintain Timecard. TC.1.22. Select first row, first column. TC.1.23. Enter "5". TC.1.24. Click on "Logout". TC.1.25. Press "Yes" if asked to save the timecard	TC.1.26. Employee asked if he/she wishes to save the timecard. TC.1.27. Timecard Saved TC.1.28. Logged out
TC.1.29. Log out with Report created and don't save the report	TC.1.30. Open Report form TC.1.31. Select "New". TC.1.32. Choose "Pay Year-To-Date Report". TC.1.33. Click on "Logout". TC.1.34. Click on "No" when asked to "Save Report"	TC.1.35. Employee Logged out TC.1.36. No report saved.
TC.1.37. Log out with report Created, Save the report and specify invalid Directory	TC.1.38. Open Report form TC.1.39. Select "New". TC.1.40. Choose "Pay Year-To-Date Report". TC.1.41. Click on "Logout". TC.1.42. Click on "Yes" when asked "SaveReport" TC.1.43. Directory: type in "C:/Mydoc/" TC.1.44. Filename: type in "MyReport.doc" TC.1.45. Press "Save"	TC.1.46. Error TC.1.47. Directory does not exist TC.1.48. Prompt to specify another directory.
TC.1.49. Log out with report Created, Save the report and specify invalid filename	TC.1.50. Open Report form TC.1.51. Select "New". TC.1.52. Choose "Year-To-Date Report". TC.1.53. Click on "Logout". TC.1.54. Click on "Yes" when asked "Save Report?" TC.1.55. Directory: type in "C:/Mydocuments/" TC.1.56. Filename: type in "My.Report.doc"	TC.1.58. Error TC.1.59. Filename is incorrect. TC.1.60. Prompt for specify filename
	TC.1.57. Press "Save"	
TC.1.61. Log out with report Created, Save the report and specify valid filename and Directory	TC.1.62. Open Report form TC.1.63. Select "New". TC.1.64. Choose "Pay Year-To-Date Report". TC.1.65. Click on "Logout". TC.1.66. Click on "Yes" when asked "Save Report?" TC.1.67. Directory: type in "C:/Mydocuments/" TC.1.68. Filename: type in "MyReport.doc" TC.1.69. Press "Save"	TC.1.70. Report saved. TC.1.71. Logged out
TC.1.72. logout without Timecard or Report changes	TC.1.73. Open "Maintain Timecard" TC.1.74. Click on "Logout".	TC.1.75. Logged out

F. Maintain Employee Information

1) Description:

These test case scenarios test the Maintain employee function of the Run Payroll System. The requirements are the System administrator should be able to add/delete employees and change employee information. The test result is presented below (see Table VI).

2) *Procedure specifications:* as in A (2) above.

G. Create Administrative Report

1) Description:

These test case scenarios test the Create administrative report function of the Run Payroll System. The requirements are that the System administrator should be able to create and save administrative reports of all/or specified employees. The test result is presented below (see Table VII).

2) *Procedure specifications:* as in A (2) above.

H. Change Payment Method

1) Description:

These test case scenarios test the Change payment method function of the Run Payroll System. The requirements are that the user should be able to change his payment method to either pickup or direct deposit. The test result is presented below (see Table VIII).

2) Procedure specifications: as in A (2) above.

Table III: Showing the Maintain Timecard test cases, test scenario and the Expected outcome

Test Case	Test Scenario	Expected Outcome
TC.1.76. Maintain Timecard, retrieve old timecard.	TC.1.77. Open "Maintain Timecard". TC.1.78. Choose Period: TC.1.79. Choose Start Date "07/15/2012" TC.1.80. Choose End Date "07/21/2012"	TC.1.81. Timecard is retrieved if already existing if not a new one is created

TC.1.82. Maintain	TC.1.83. Open	TC.1.89.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 7, January 2013

Timecard	"Maintain Timecard". TC.1.84. Choose current Period TC.1.85. Add a new project row TC.1.86. Choose a Charge Number from the list TC.1.87. Choose day on the project row TC.1.88. Enter Hours worked "8"	Timecard updated
TC.1.90. Maintain Timecard with invalid charge number	TC.1.91. Open "Maintain Timecard". TC.1.92. Choose current Period: TC.1.93. Add a new project row TC.1.94. Choose Charge Number "XXXXXX" TC.1.95. Choose day "07/18" TC.1.96. Enter Hours worked "8"	TC.1.97. Error TC.1.98. Charge number don't exist TC.1.99. Enter valid charge number
TC.1.100. Maintain Timecard TC.1.101. With invalid hours worked	TC.1.102. Open "Maintain Timecard". TC.1.103. Choose current Period: TC.1.104. Choose an existing project and day TC.1.105. Enter Hours worked "-8"	TC.1.106. Error. TC.1.107. Invalid number of hours worked. Prompted to enter Valid hours. TC.1.108. Time card remains unchanged.
TC.1.109. Maintain Timecard TC.1.110. With too many hours worked per day.	TC.1.111. Open "Maintain Timecard". TC.1.112. Choose current Period: TC.1.113. Choose an existing project and day TC.1.114. Enter Hours worked "25"	TC.1.115. Error. TC.1.116. Hours worked must be less than 24. TC.1.117. Prompted to enter Valid hours. TC.1.118. Time card remains unchanged
TC.1.119. Save Timecard	TC.1.120. Open "Maintain Timecard". TC.1.121. Choose current Period: TC.1.122. Choose an existing project and day TC.1.123. Enter Hours worked "8" TC.1.124. Click on "Save"	TC.1.125. Time card successfully saved:
TC.1.126. Logout	TC.1.127. Test case covered by the Logout test cases	
TC.1.128. Submit Timecard	TC.1.129. Open "Maintain Timecard". TC.1.130. Choose current Period: TC.1.131. Click on submit TC.1.132. Choose "Yes" when asked "Submit?"	TC.1.133. Time card successfully submitted TC.1.134. No more editing allowed.
TC.1.135. Change Submitted Timecard	TC.1.136. Open "Maintain Timecard". TC.1.137. Choose a submitted timecard: TC.1.138. Try to change any numbers on the Timecard TC.1.139. Try to add a new project row	TC.1.140. Can not enter hours worked TC.1.141. Timecard remains unchanged

Table IV: Showing the Change Password test cases, test scenario and the Expected outcome

Test Case	Test Scenario	Expected Outcome
TC.1.142. Change Password.	TC.1.143. Open "Preferences". TC.1.144. Chose "Change Password". TC.1.145. Enter "whatsup" in field "New password" TC.1.146. Enter "whatsup" in field "Confirm Password" TC.1.147. Click on "Change Password" button.	TC.1.148. Password is changed in database TC.1.149. The user is notified through a message box that the operation was carried out successfully.
TC.1.150. Change Password with invalid confirmation	TC.1.151. Open "Preferences". TC.1.152. Chose "Change Password". TC.1.153. Enter "whatsup" in field "New password" TC.1.154. Enter "what" in field "Confirm Password" TC.1.155. Click on "Change Password" button.	TC.1.156. No change in password takes places. TC.1.157. The password fields are emptied.

Table V: Showing the Create Employee Report test cases, test scenario and the Expected outcome

Test Case	Test Scenario	Expected Outcome
TC.1.158. Create "Total Hours Worked" Report	TC.1.159. Open "Reports" TC.1.160. Choose "New" TC.1.161. Choose Report Type "Total	TC.1.165. A Report for total hours worked generated.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 7, January 2013

	Hours Worked” TC.1.162. Choose Start Date “10/15/2012” TC.1.163. Choose End Date “10/30/2012” TC.1.164. Click “Create”	
TC.1.166. Create “Total hours worked for a project” Report.	TC.1.167. Open “Reports” TC.1.168. Choose “New” TC.1.169. Choose Report Type “Total Hours Worked” TC.1.170. Choose Start Date “01/03/2012” TC.1.171. Choose End Date “31/03/2012” TC.1.172. Enter “AB1234” in charge number field TC.1.173. Click “Create”	TC.1.174. A report for total hours worked on a project generated
TC.1.175. Create “Vacation or Sick leave” Report	TC.1.176. Open “Reports” TC.1.177. Choose “New” TC.1.178. Choose Report Type “Vacation or Sick Leave” TC.1.179. Choose Start Date “10/01/2012” TC.1.180. Choose End Date “10/02/2012” TC.1.181. Click “Create”	TC.1.182. Report for total vacation and sick leaves generated
TC.1.183. Create “Year-To-Date” Report	TC.1.184. Open “Reports” TC.1.185. Choose “New” TC.1.186. Choose Report Type “Year-To-Date” TC.1.187. Choose Start Date “01/03/2002” TC.1.188. Choose End Date “31/02/2012” TC.1.189. Click “Create”	TC.1.190. “Year-To-Date” report generated
TC.1.191. Create Report with invalid Start Date	TC.1.192. Open “Reports” TC.1.193. Choose “New” TC.1.194. Choose Report Type “Total Hours Worked” TC.1.195. Choose Start Date “10/00/2012” TC.1.196. Choose End Date “10/30/2012”	TC.1.198. Error - Invalid Start date. TC.1.199. No Report Created.
	TC.1.197. Click “Create”	
TC.1.200. Create Report with invalid end date	TC.1.201. Open “Reports” TC.1.202. Choose “New” TC.1.203. Choose Report Type “Total Hours Worked” TC.1.204. Choose Start Date “01/03/2012” TC.1.205. Choose End Date “40/02/2012” TC.1.206. Click “Create”	TC.1.207. Error TC.1.208. Invalid End date TC.1.209. No Report Created
TC.1.210. Create Report with no type set.	TC.1.211. Open “Reports” TC.1.212. Choose “New” TC.1.213. Choose Start Date “01/10/2012” TC.1.214. Choose End Date “30/10/2012” TC.1.215. Click “Create”	TC.1.216. Error TC.1.217. Must enter a valid report type TC.1.218. No Report Created
TC.1.219. Create “Total hours worked for a project” Report with invalid charge number	TC.1.220. Open “Reports” TC.1.221. Choose “New” TC.1.222. Choose Report Type “Total Hours Worked” TC.1.223. Choose Start Date “01/10/2012” TC.1.224. Choose End Date “30/03/2012” TC.1.225. Enter “XXXXX” in charge number field TC.1.226. Click “Create”	TC.1.227. Error TC.1.228. Invalid charge number. TC.1.229. Charge number do not exist. TC.1.230. No Report Created.
TC.1.231. Create “Total hours worked for a project” Report with no charge number entered	TC.1.232. Open “Reports” TC.1.233. Choose “New” TC.1.234. Choose Report Type “Total Hours Worked” TC.1.235. Choose Start Date “01/03/2012” TC.1.236. Choose End Date “30/03/2012” TC.1.237. Click “Create”	TC.1.238. Error TC.1.239. Must enter a valid charge number
TC.1.240. Save Report	TC.1.241. Open “Reports” TC.1.242. Choose “New” TC.1.243. Choose Report Type “Total Hours Worked”	TC.1.250. Report Saved in directory specified

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 7, January 2013

	TC.1.244. Choose Start Date "15/03/2012" TC.1.245. Choose End Date "30/03/2012" TC.1.246. Click "Create" TC.1.247. Click "Save". TC.1.248. Enter "MyReport.doc" in filename field TC.1.249. Enter "C:/mydocuments" in directory field	
TC.1.251. Save Report with invalid Directory TC.1.252.	TC.1.253. Open "Reports" TC.1.254. Choose "New" TC.1.255. Choose Report Type "Total Hours Worked" TC.1.256. Choose Start Date "15/03/2012" TC.1.257. Choose End Date "30/03/2012" TC.1.258. Click "Create" TC.1.259. Click "Save". TC.1.260. Enter "MyReport.doc" in filename field TC.1.261. Enter "mydocuments" in directory field	TC.1.262. Error TC.1.263. Invalid directory TC.1.264. Report not saved TC.1.265.
TC.1.266. Save Report with invalid filename	TC.1.267. Open "Reports" TC.1.268. Choose "New" TC.1.269. Choose Report Type "Total Hours Worked" TC.1.270. Choose Start Date "15/03/2012" TC.1.271. Choose End Date "30/03/2012" TC.1.272. Click "Create" TC.1.273. Click "Save". TC.1.274. Enter "My.Report.doc" in filename field TC.1.275. Enter "C:/mydocuments" in directory field	TC.1.276. Error TC.1.277. Invalid filename TC.1.278. Report not saved

Table VI: Showing the Maintain Employee Information test cases, test scenario and the Expected outcome

Test Case	Test Scenario	Expected Outcome
TC.1.279. Add new employee	TC.1.280. Open "Maintain employee" TC.1.281. Choose "New" TC.1.282. Enter "100920103" in "Employee ID" field TC.1.283. Enter "Philip Ikuesan" in "Name" field TC.1.284. Enter nelson@gmail.com" in "Email" field TC.1.285. Enter "010180-201P" in "Social security number" field TC.1.286. Enter "25.50" in "standard deduction" field TC.1.287. Enter "5.0" in "Other deductions" field TC.1.288. Enter "(041) 1234567" in "Phone number" field TC.1.289. Enter "3000.00" in "Salary" field TC.1.290. Enter "Philip" in "Username" field TC.1.291. Enter "Philip" in "Password" field TC.1.292. Click "Add employee" TC.1.293.	TC.1.294. Employee Added TC.1.295. Payment method set to Pickup
TC.1.296. Add employee with invalid information	TC.1.297. Open "Maintain employee" TC.1.298. Choose "New" TC.1.299. Enter "100920103" in "Employee ID" field TC.1.300. Enter "nelson@gmail.com" in "Email" field TC.1.301. Enter "010180-201P" in "Social security number" field TC.1.302. Enter "25.5" in "standard deduction" field TC.1.303. Enter "5.0" in "Other deductions" field TC.1.304. Enter "(081) 1234567" in "Phone number" field TC.1.305. Enter "philip" in "Password" field TC.1.306. Click "Add	TC.1.307. Error TC.1.308. The information not completed TC.1.309. Show information needed TC.1.310. Employee not added TC.1.311.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 7, January 2013

	employee”	
TC.1.312. Update employee	TC.1.313. Open “Maintain employee” TC.1.314. Choose “Update” TC.1.315. Enter “100920103” in “Employee ID” field TC.1.316. Click “Ok” TC.1.317. Change “Email” field to “nelson@gmail.com” TC.1.318. Change “Phone Number” to “(081) 7654321” TC.1.319. Click on “Update” TC.1.320. Click on “Ok”	TC.1.321. Employee Updated
TC.1.322. Cancel “Update employee”	TC.1.323. Open “Maintain employee” TC.1.324. Choose “Update” TC.1.325. Click “Ok” TC.1.326. Enter “100920103” in “Employee ID” field Change “Email” field to “nelson@gmail.com” TC.1.327. Change “Phone Number” to (080) 1324534” TC.1.328. Click on “Update” TC.1.329. Click on “Cancel”	TC.1.330. Employee not Updated .
TC.1.331. Update Employee with invalid field information	TC.1.332. Open “Maintain employee” TC.1.333. Choose “Update” TC.1.334. Enter “100920103” in “Employee ID” field TC.1.335. Click “Ok” TC.1.336. Change “Email” field to “nelsongtmail.com” TC.1.337. Click on “Update” TC.1.338. Click on “Cancel”	TC.1.339. Error TC.1.340. Employee not Updated TC.1.341. Information not complete
TC.1.342. Enter invalid Employee ID	TC.1.343. Open “Maintain employee” TC.1.344. Choose “Update” TC.1.345. Enter “100923010” in “Employee ID” field TC.1.346. Click “Ok”	TC.1.347. Error TC.1.348. Employee not found TC.1.349. Invalid Employee ID
TC.1.350. Cancel “Delete Employee”	TC.1.351. Open “Maintain employee” TC.1.352. Choose “Delete” TC.1.353. Enter “100920103” in “Employee ID” field	TC.1.357. Employee is not marked for deletion
	TC.1.354. Click “Ok” TC.1.355. Click “Mark for Deletion” TC.1.356. Click “Cancel” when asked to confirm	
TC.1.358. Delete employee	TC.1.359. Open “Maintain employee” TC.1.360. Choose “Delete” TC.1.361. Enter “100920103” in “Employee ID” field TC.1.362. Click “Ok” TC.1.363. Click “Mark for Deletion” TC.1.364. Click “OK” when asked to confirm	TC.1.365. Employee is marked for deletion TC.1.366.
TC.1.367. Delete Employee with employee already marked for deletion	TC.1.368. Open “Maintain employee” TC.1.369. Choose “Delete” TC.1.370. Enter “100920103” in “Employee ID” field TC.1.371. Click “Ok” TC.1.372.	TC.1.373. Display “Employee already deleted”.

Table VII: Showing the Create Administrative Report test cases, test scenario and the Expected outcome

Test Case	Test Scenario	Expected Outcome
TC.1.374. Create “Total Hours Worked” Report with all employees	TC.1.375. Open “Reports” TC.1.376. Choose “New” TC.1.377. Choose Report Type “Total Hours Worked” TC.1.378. Choose Start Date “15/03/2012” TC.1.379. Choose End Date “30/03/2012” TC.1.380. Choose “All Employees” TC.1.381. Click “Create”	TC.1.382. A Report for total hours worked for all employees is generated
TC.1.383. Create “Total Hours Worked” Report with specified number of employees	TC.1.384. Open “Reports” TC.1.385. Choose “New” TC.1.386. Choose Report Type “Total Hours Worked” TC.1.387. Choose Start Date “15/03/2012” TC.1.388. Choose End Date “30/03/2012” TC.1.389. Choose employee ID:s “100909290”,	TC.1.391. A Report for total hours worked for specified employees is generated

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 7, January 2013

	"100727647" and "100633070" TC.1.390. Click "Create"	
TC.1.392. Create "Total hours worked for a project" Report with all employees	TC.1.393. Open "Reports" TC.1.394. Choose "New" TC.1.395. Choose Report Type "Total Hours Worked" TC.1.396. Choose Start Date "01/02/2012" TC.1.397. Choose End Date "28/02/2012" TC.1.398. Choose "All Employees" TC.1.399. Enter "AB1234" in charge number field TC.1.400. Click "Create"	TC.1.401. A report for total hours worked on a project for all employees is generated
TC.1.402. Create "Total hours worked for a project" Report with specified number of employees	TC.1.403. Open "Reports" TC.1.404. Choose "New" TC.1.405. Choose Report Type "Total Hours Worked" TC.1.406. Choose Start Date "01/03/2012" TC.1.407. Choose End Date "30/03/2012" TC.1.408. Choose employee ID:s "100909290","100727647" and "100633070" TC.1.409. Enter "AB1234" in charge number field TC.1.410. Click "Create"	TC.1.411. A report for total hours worked on a project for specified employees is generated
TC.1.412. Create "Vacation or Sick leave" Report with all employees	TC.1.413. Open "Reports" TC.1.414. Choose "New" TC.1.415. Choose Report Type "Vacation or Sick Leave" TC.1.416. Choose Start Date "01/03/2012" TC.1.417. Choose End Date "30/03/2012" TC.1.418. Choose "All Employees" TC.1.419. Click	TC.1.420. Report for total vacation and sick leaves for all employees is generated
	"Create"	
TC.1.421. Create "Vacation or Sick leave" Report for specified number of employees	TC.1.422. Open "Reports" TC.1.423. Choose "New" TC.1.424. Choose Report Type "Vacation or Sick Leave" TC.1.425. Choose Start Date "10/02/2012" TC.1.426. Choose End Date "28/02/2012" TC.1.427. Choose employee ID:s "100909290","100727647" and "100633070" TC.1.428. Click "Create"	TC.1.429. Report for total vacation and sick leaves for specified employees is generated
TC.1.430. Create "Year-To-Date" Report with all employees	TC.1.431. Open "Reports" TC.1.432. Choose "New" TC.1.433. Choose Report Type "Year-To-Date" TC.1.434. Choose Start Date "01/01/2012" TC.1.435. Choose End Date "30/03/2012" TC.1.436. Choose "All Employees" TC.1.437. Click "Create"	TC.1.438. "Year-To-Date" report for all employees is generated
TC.1.439. Create "Year-To-Date" Report for specified number of employees	TC.1.440. Open "Reports" TC.1.441. Choose "New" TC.1.442. Choose Report Type "Year-To-Date" TC.1.443. Choose Start Date "01/03/2012" TC.1.444. Choose End Date "30/03/2012" TC.1.445. Choose employee ID "100909290","100727647" and "100633070" TC.1.446. Click "Create"	TC.1.447. "Year-To-Date" report for specified employees is generated
TC.1.448. Create Report for specified number of employees with wrong employee ID	TC.1.449. Open "Reports" TC.1.450. Choose "New" TC.1.451. Choose Report Type "Year-To-Date"	TC.1.456. Error TC.1.457. Employee not found TC.1.458. No report created

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 7, January 2013

	TC.1.452. Choose Start Date "01/03/2012" TC.1.453. Choose End Date "30/03/2012" TC.1.454. Choose employee ID "300909290","100727647" and "300633070" TC.1.455. Click "Create"	
TC.1.459. Create Report with invalid Start Date	TC.1.460. Open "Reports" TC.1.461. Choose "New" TC.1.462. Choose Report Type "Total Hours Worked" TC.1.463. Choose Start Date "00/03/2012" TC.1.464. Choose End Date "30/03/2012" TC.1.465. Choose "All Employees" TC.1.466. Click "Create"	TC.1.467. Error-Invalid Start date. TC.1.468. No Report Created.
TC.1.469. Create Report with invalid end date	TC.1.470. Open "Reports" TC.1.471. Choose "New" TC.1.472. Choose Report Type "Total Hours Worked" TC.1.473. Choose Start Date "01/03/2012" TC.1.474. Choose End Date "40/03/2012" TC.1.475. Choose "All Employees" TC.1.476. Click "Create"	TC.1.477. Error TC.1.478. Invalid End date TC.1.479. No Report Created
TC.1.480. Create Report with no type set.	TC.1.481. Open "Reports" TC.1.482. Choose "New" TC.1.483. Choose Start Date "01/03/2012" TC.1.484. Choose End Date "30/01/2012" TC.1.485. Choose "All Employees" TC.1.486. Click "Create"	TC.1.487. Error TC.1.488. Must enter a valid report type TC.1.489. No Report Created
TC.1.490. Create "Total hours worked for a project" Report with invalid charge number	TC.1.491. Open "Reports" TC.1.492. Choose "New" TC.1.493. Choose Report Type "Total Hours Worked" TC.1.494. Choose Start Date "01/03/2012" TC.1.495. Choose End Date "30/03/2012" TC.1.496. Enter "XXXXX" in charge number field TC.1.497. Choose "All Employees" TC.1.498. Click "Create"	TC.1.499. Error TC.1.500. Invalid charge number. TC.1.501. Charge number do not exist. TC.1.502. No Report Created.
TC.1.503. Create "Total hours worked for a project" Report with no charge number entered	TC.1.504. Open "Reports" TC.1.505. Choose "New" TC.1.506. Choose Report Type "Total Hours Worked" TC.1.507. Choose Start Date "01/03/2012" TC.1.508. Choose End Date "30/03/2012" TC.1.509. Choose "All Employees" TC.1.510. Click "Create"	TC.1.511. Error TC.1.512. Must enter a valid charge number TC.1.513.
TC.1.514. Save Report	TC.1.515. Open "Reports" TC.1.516. Choose "New" TC.1.517. Choose Report Type "Total Hours Worked" TC.1.518. Choose Start Date "15/03/2012" TC.1.519. Choose End Date "30/03/2012" TC.1.520. Choose "All Employees" TC.1.521. Click "Create" TC.1.522. Click "Save". TC.1.523. Enter "MyReport.doc" in filename field TC.1.524. Enter "C:/mydocuments" in directory field	TC.1.525. Report Saved in directory specified
TC.1.526. Save Report with invalid Directory TC.1.527.	TC.1.528. Open "Reports" TC.1.529. Choose "New" TC.1.530. Choose Report Type "Total Hours Worked" TC.1.531. Choose	TC.1.538. Error TC.1.539. Invalid directory TC.1.540. Report not saved TC.1.541.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 7, January 2013

	Start Date "15/03/2012" TC.1.532. Choose End Date "30/03/2012" TC.1.533. Choose "All Employees" TC.1.534. Click "Create" TC.1.535. Click "Save". TC.1.536. Enter "MyReport.doc" in filename field TC.1.537. Enter "mydocuments" in directory field	
TC.1.542. Save Report with invalid filename	TC.1.543. Open "Reports" TC.1.544. Choose "New" TC.1.545. Choose Report Type "Total Hours Worked" TC.1.546. Choose Start Date "15/03/2012" TC.1.547. Choose End Date "30/03/2012" TC.1.548. Choose "All Employees" TC.1.549. Click "Create" TC.1.550. Click "Save". TC.1.551. Enter "My.Report.doc" in filename field TC.1.552. Enter "C:/mydocuments" in directory field	TC.1.553. Error TC.1.554. Invalid filename TC.1.555. Report not saved

Table VIII: Showing the Change Payment Method test cases, test scenario and the Expected outcome

Test Case	Test Scenario	Expected Outcome
TC.1.556. Choose Pick-up	TC.1.557. Open "Preferences" TC.1.558. Click on "Change payment method" TC.1.559. Choose "Pick-up" TC.1.560. Click "Update" TC.1.561. Click "OK" when asked to confirm	TC.1.562. System updated to default (pick-up) payment method TC.1.563. Message read "updated"
TC.1.564. Choose Direct Deposit TC.1.565.	TC.1.566. Open "Preferences" TC.1.567. Click on "Change payment method" TC.1.568. Choose "Direct Deposit" TC.1.569. Enter "SKYE" in "Bank Name" field TC.1.570. Enter "022-90987" in field "Account number" TC.1.571. Click "Update" TC.1.572. Click "OK" when asked to confirm	TC.1.573. Payment method changed to Direct Deposit. TC.1.574. Message read "updated"
TC.1.575. Choose Direct Deposit without specifying account number	TC.1.576. Open "Preferences" TC.1.577. Click on "Change payment method" TC.1.578. Choose "Direct Deposit" TC.1.579. Enter "SKYE" in "Bank Name" field TC.1.580. Click "Update" TC.1.581. Click "OK" when asked to confirm	TC.1.582. Error TC.1.583. Message read "Enter your account number" TC.1.584. Payment method not-changed.
TC.1.585. Choose Direct Deposit TC.1.586. Without specifying Bank name	TC.1.587. Open "Preferences" TC.1.588. Click on "Change payment method" TC.1.589. Choose "Direct Deposit" TC.1.590. Enter "022-90987" in field "Account number" TC.1.591. Click "Update" TC.1.592. Click "OK" when asked to confirm	TC.1.593. Error TC.1.594. Message read "Enter bank name" TC.1.595. Payment method not-changed.
TC.1.596. Cancel changes	TC.1.597. Open "Preferences" TC.1.598. Click on "Change payment method" TC.1.599. Choose "Direct Deposit" TC.1.600. Enter "022-90987" in field "Account number" TC.1.601. Click "Update" TC.1.602. Click "Cancel" when asked to confirm	TC.1.603. Payment method not-changed. TC.1.604.

VI. Conclusion

As software has grown more complex, the amount of errors in it, known as bugs, has increased. Market pressures can further compound this problem by causing a project to be developed with unskilled programmers or insufficient time or money. It is estimated that software errors lead to costs of tens of milliards of euros every year. Bugs are essentially a difference between the intended behaviour of the program and its actual behaviour. Thus,

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 7, January 2013

one way to end and eliminate bugs (an activity known as debugging) is to examine the operation of the program and compare this to the desired operation. This approach is called testing. This paper in its simplicity has been able to show case the design of payroll system and how it can well be tested with a detailed procedural analysis of its execution steps, therefore minimizing the risk of use of such application in organizations where the need is required. Future work should embrace other test approaches such class testing, round trip scenario test, polymorphic test in other to validate the correctness of this type of design. Also vein method of biometric for employee validity check is another aspect in which software developer might be interested for a secured and safe security platform.

REFERENCES

[1] C. Ghezzi, M. Jazayeri and D. Mandrioli (2002), Fundamental of software engineering," Prentice Hall Publication, sept 2002, 2nd edition, pp 604.

[2] James Bach (1999), "Risk and Requirements-Based Testing," IEEE Computer Society, http://www.cdainfo.com/Down/5-Test/Requirements.pdf

[3] Cam Kaner, J.D (2006), "Exploratory Testing, a Keynote at QAI," November 17, 2006.

[4] Nuno Antunes and Marco Vieira (2012), "Defending against Web Application Vulnerabilities," http://www.infoq.com/articles/defending-against-web-application-vulnerabilities.

[5] Nuno Antunes and Macro Vieira, "Comparing the Effectiveness of Penetration Testing and Static Code Analysis on the Detection of SQL Injection Vulnerabilities in Web Services," Proc. 15th IEEE Pacific Rim Int'l Symp. Dependable Computing (PRDC 09), IEEE CS, 2009, pp. 301-306.

[6] B. Arkin, S. Stender, and G. McGraw, "Software Penetration Testing," IEEE Security & Privacy, Jan.-Feb. 2005, pp. 84-87.

[7] Andreas Leitner, Uinca Ciupa, Manuel Oriol, Bertrand Meyer and Arno Fiva (2007), "Contract Driven Development-Test Driven Development-Writing Test Cases," ESEC/FSE'07, Cavtat Dubrovnik, Crotia, ACM 978-1-59593-811-4/07/009

[8] N. Ayewah and W. Pugh, "A Report on a Survey and Study of Static Analysis Users," Proc. Workshop Defects in Large Software Systems (DEFECTS 08) ACM, 2008, pp. 1-5.

[9] D.P. Freedman and G.M. Weinberg, "Handbook of Walkthroughs, Inspections, and Technical Reviews: Evaluating Programs, Projects, and Products," Dorset House, 2000.

[10] Robert V. Binder. (1999) "Testing Object-Oriented Systems," Addison Wesley Longman Inc.

AUTHOR BIOGRAPHY

Akinboyewa Nelson E. received his B.Eng degree in Information and Media Technology from Mikkeli University of Applied Science, Mikkeli, Finland in June 2004. His areas of research interest include Software development and testing, Web applications and Networking. He is Cisco system Networking academy certified, a member of Finnish Engineering society (Suomen Insiniöörin ammattilinen), and a member of Association of Computing Machinery (ACM).

Johnson O. Victor received his M.Sc degree in Computer Science from University of Benin, Nigeria in 2008. He is currently pursuing his PhD in the same institution with a focus on Software Success/Failure and Data Mining. He is also a member of Society of Computer Professionals of Nigeria (CPN), a member of IEEE and IEEE Computer Society, a member of Association of Computing Machinery (ACM) and a member of Association of Engineers (AENG).