**Introduction to Software Engineering**

**Writing Test Cases**

A Test Plan contains a discussion of the testing activities to be conducted on the software product. It describes the way in which we will show that the software works correctly according to the stated requirements specifications. The Test Plan also identifies the people who will perform the test, the objectives of performing each type of tests, and the method to be employed in conducting the test.

For the first iteration of your software, only the Test Cases need to be submitted. The following are the guidelines for writing your Test Cases:

**Guidelines:**

1. Use the Project Coding Schedule / Progress Chart as reference document to Identify the priority user stories for Iteration 1 and the current development status of each of these stories.
2. Prepare test cases for the priority Iteration 1 User Stories (regardless of their current development status); the test cases should include the following types:
   1. Black box testing (required)
   2. Function testing
   3. Usability testing
   4. Design and Code review (reiteration of previous QA to check compliance with design principles of classes, UI standards for user forms, and coding standards for code)

**Reminders:**

1. Think of critical test cases that can potentially reveal functional defects
   1. Test cases must be effective – they will uncover defects
   2. Test cases must be efficient – fewest number of tests that can reveal the most number of defects
2. Make sure to include positive (valid) and negative (invalid) test cases to account for user errors
3. Make sure you provide accurate and honest results; grading is not based on the number of PASS/FAIL outcome, but on the **quality of the test cases and the reporting of defects**.
4. Make sure you have enough data to write your Test Report
5. Your test plan, specifically the function testing, should be executable for at most two (2) hours
6. The Test Cases will be presented on **March 9, 2015 (Mon)**. No need to print yet, but the file must be made available to your teacher via Google Docs.
7. A complete and printed Test Case will be submitted on **March 16, 2015 (Mon)**, as part of the Test Report. Prepare for class presentation.
8. DEV team: Bring a laptop with your running software already installed and ready for class presentation and testing on **March 9 and 16**.
9. Late groups /group members will receive corresponding deductions.



De La Salle University

College of Computer Studies

Software Technology Department

**<SOFTWARE NAME>**

TEST CASES

|  |  |
| --- | --- |
| **Team Name** | **Blue Barracudas/Red Jaguars** |
| **Section** |  |
| **Team Members** | In alphabetical order based on surname  <surname>, <firstname> <m.i.>  <surname>, <firstname> <m.i.>  : |
| **Date Submitted** |  |

**Test Cases**

**For each priority Iteration 1 User Story, prepare a test case table containing >1 test cases, following the format below:**

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story ID / Title:** | | **<#> / <Verb> <Noun>** | **Priority: <N>** |
| **Type of Test** | | **{Black box, Function, Usability, Design Review, Code Review}** | |
| **Purpose of Test** | | ***Briefly describe what you intend to find out by doing this test.*** | |
| **Test Case Number** | | ***<user story id#> - <sequence #>*** | |
|  | Objective of Test Case | *Briefly describe the equivalence class or function you are testing for this test case* | |
|  | Input | *Specify the input values or test data to be used* | |
|  | Expected Results | *Describe the expected output* | |
|  | Actual Results | *Describe the actual outcome when you run the software with the given input* | |
|  | Findings | *PASS or FAIL 🡺 any difference, no matter how slight, between Expected and Actual results is a FAIL* | |
|  | Remarks | *What steps will reproduce the problem? What else would you like to tell the developers regarding the implementation of this user story?* | |
| Test Case Number | | *<user story id#> - <sequence #> ==> 1 User Story should have >1 Test Cases* | |

**Sample**

|  |  |  |
| --- | --- | --- |
| **User Story ID / Title:** | | **1 / Add Shirt** |
| **Type of Test** | | Black box |
| **Purpose of Test** | | Check if the software allows new shirt records to be created and stored |
| **Test Case Number** | | **1-1** |
|  | Objective of Case | Check that the software will not accept duplicate shirt code |
|  | Input | TSM-0001 (an existing shirt code) |
|  | Expected Result | Software will not add the new shirt record and will inform the user that “Shirt TSM-0001 already exists in the database. Please use another code.” |
|  | Actual Result | Software added the duplicate shirt code into the database |
|  | Conclusion | Fail |
|  | Remarks | Shirt code is a primary key and should be unique. |
| **Test Case Number** | | **1-2** |
|  | Objective of Case | Check that the software will correctly store the new shirt record into the database and will inform the user about the successful operation |
|  | Input | Valid shirt details:  Shirt Code: TSM-0002  Type: T-Shirt  Gender: Male  Size: M  Color: Green  Quantity: 10 |
|  | Expected Result | Software will insert the new shirt record into the database and will inform the user that “Shirt TSM-0002” has been successfully added to the database.” |
|  | Actual Result | Software inserted the new shirt record into the database but did not display any message |
|  | Conclusion | Fail |
|  | Remarks | The testers verified that the shirt record has been inserted into the database through the MySQL Workbench. |