Master Test Plan

1. Test Plan Identifier: WMS_TEST_ASSGN6

2. Authors:

Sarika Bishnoi (21CS10058), Sreejita Saha (21CS30052), Anwesha Das (21CS30007)

3. References:

a. SRS submitted as a part of Assignment

b. Project Description

c. https://jmpovedar.files.wordpress.com/2014/03/ieee-829.pdf

4. Introduction:

This is the Master Test Plan for the Work Management System (WMS) submitted. This plan will address testing of all items and elements that are related to the WMS, both directly and indirectly. The project will have unit testing, the details for each are addressed in the appropriate section.

5. Test Items:

a. Worker Class:

- name, contact, worker_id, role, skill_level, gender, availability, noofhours_worked, works_completed, last_active_work_date
- availability_status()

b. Work Class:

- name, status, work_id, noofworkers, start_date, requirement, priority, duration
- work_status()

c. Global Functions:

- addworker()
- editworker()
- delete_a_worker()
- viewedit_a_worker()
- display_all_workers()
- assignment_of_workers()
- addwork()
- editwork()
- viewdetails_work()
- view_all_works()

6. Features to be tested:

a. Add/modify/delete the worker details
b. Add/modify the work details
c. Assign workers to the works
d. Mark work status as completed
e. Display worker and work detail
f. User Interface (GUI)

7. Features not to be tested:

a. The database deployed via sqlite3 would NOT be tested
b. The MySQL server should not be tested for compatibility with unsupported operating system or hardware.
c. The syntax and semantics of Python code used to interact with the GUI should not be tested, as they are a part of the Python language and not the software being developed.

8. Pass/Fail Criteria:

In the test run, a test case will be considered as Pass if after running the cases, the software functions as designed. Any test case which does otherwise is considered as Fail.

We will be providing the test cases, which are to be run in the software for testing, along with the result of the test run.

9. Test Plans and Scenarios:

1. Worker Class:

1. name → String
2. contact → String
3. worker_id →Integer
4. role →String
5. skill_level →String
6. gender →String
7. availability →Bollean
8. noofhours_worked →Integer
9. works_completed →List
10. last_active_work_date →Date

Plan:

1. All the functionalities of this class are displayed in the Worker Window.
2. Click on Add Worker to go to the Add Worker window and fill all the details (user is given prompts to enter the details) to add a new worker in the system database.
3. Click on Delete Worker to go to the Delete Worker window and enter the worker id to delete the specified worker from the system database.
4. Click on Edit Worker to go to the Edit Worker window. Enter the worker id to specify the worker. Choose and edit the attributes you wish to edit of the selected worker.
5. Click on Display Worker to go to the Display Worker window and enter the worker id to display the details of a selected worker.
6. Click on the Exit button to go to the previous window. This is applicable to the Worker Window and all the windows the user opens from it subsequently.

Output:

1. Displays the message “Worker added successfully!” when the addition of a worker is completed.
2. Displays the message “Worker deleted successfully!” when the deletion of a selected worker is completed.
3. Displays the message “Worker edited successfully!” when the user completes editing the details of a worker.
4. Displays all the details(data members of the worker class) of the selected worker when the worker id is entered in the Display Worker Window.

2. Work Class:

1. name → String
2. status → String
3. work_id → Integer
4. noofworkers → Integer
5. start_date →Date
6. requirement →Table
7. priority → Integer
8. duration →Integer (number of days)

Plan:

1. All the functionalities of this class are displayed in the Work Window.
2. Click on Add Work to go to the Add Work window and fill all the details (user is given prompts to enter the details) to add a new work in the system database.
3. Click on Edit Work to go to the Edit Work window. Enter the work id to specify the work. Choose and edit the attributes you wish to edit of the selected work.
4. Click on Display Work to go to the Display Work window and enter the work id to display the details of a selected work.
5. Click on the Exit button to go to the previous window. This is applicable to the Work Window and all the windows the user opens from it subsequently.

Output:

1. Displays the message “Work added successfully!” when the addition of a work is completed.
2. Displays the message “Work edited successfully!” when the user completes editing the details of a work.
3. Displays all the details(data members of the work class) of the selected work when the work id is entered in the Display Work Window.

.

3. Global Functions:

a. Display all Works

Plan :
Click on the Display All Work Work window to display.

Output :

Displays all the works currently in the system in a tabular form, the fields/columns of the table being the data members of the class Work.

b. Display all Workers

Plan :
Click on the Display All Work Work window to display.

Output :

Displays all the workers currently in the system in a tabular form, the fields/columns of the table being the data members of the class Worker.

c. Assignment

Plan :

The assignment function can be run in two ways :

a. When a work is completed and all the workers engaged in that work are now free.
b. When the user clicks on Assignment in the Assignment Window.

In both these cases, the Assignment function does the same work :

1. Gets the list of all the works which have not started yet (checks their status) sorted according to their priority.
2. Gets the list of all the workers wich are available.
3. Allots the required workers to the works starting from highest priority work and displays all the details related to the assignments of the works.

For the automatic Assignment, when the user ends a work, the Assignment function also uses the workers which have been freed for the assignment. If the requirements of any work is fulfilled, then that work is automatically started.

Output :

a. When the user ends a work (Assignment will happen automatically)
Displays the works assigned and their respective requirements (number of workers of particular roles and skill levels) and the works which have started(if any) and their details.

b. When the user clicks on Assignment (Assignment done by user manually)
Displays the works assigned and their respective requirements (number ofworkers of particular roles and skill levels).

Interface Testing

- **Check Basic GUI elements**

  For the following Tkinter GUI elements, we describe the basic properties that must be tested for appropriate/error-free behavior wherever they appear in our GUI

  1. **Buttons**
  Check if all buttons are clickable and active

  2. **RadioButtons**
  Check if exactly one is selected

  3. **TextBoxes**
  Check for text entry is not empty

  4. **ListBoxes/ComboBoxes**
  Check if atleast one option is selected

  5. **Multiple Windows**
  Check if all the windows are opening/closing properly and in correct order

- **Check Common GUI features**

  1. **'Exit' Buttons**
  Check if provided in every page to Go Back to the previous page

  2. **OK Buttons**
  Check if all 'Required' text entries are filled before execution