**NEWSPAPER AGENCY**

**APPLICATION DEVELOPMENT – IV**

**IMPLEMENTATION PHASE**

*Submitted in partial fulfilment for the award of the degree*

*of*

**Master of Science**

***in***

**Information Technology**

*by*

**SANKET SURESH PETHKAR**

**REG NO: 14MIN2879**



**School of Information Technology and Engineering**

March 2016

Table of Contents

[5.3 SCREEN SHOTS 7](#_Toc458785295)

[6 TESTING AND DOCUMENTATION 14](#_Toc458785296)

[6.1 TEST PLAN 14](#_Toc458785297)

[6.1.1 INTRODUCTION 14](#_Toc458785298)

[6.1.2 OBJECTIVE 14](#_Toc458785299)

[6.1.3 DEFINITIONS 14](#_Toc458785300)

[6.1.4 TEST ITEMS 15](#_Toc458785301)

[6.1.5 APPROACH 15](#_Toc458785302)

[6.1.5.1 Unit Testing 15](#_Toc458785303)

[6.1.5.2 Integration Testing: 16](#_Toc458785304)

[6.1.5.3 Regression Testing: 16](#_Toc458785305)

[6.1.5.4 Acceptance Testing: 16](#_Toc458785306)

[6.1.5.5 System Testing: 16](#_Toc458785307)

[6.2 TEST DESIGN/TEST CASE SPECIFICATIONS 17](#_Toc458785308)

[6.2.1 Registration Successful 17](#_Toc458785309)

[6.2.2 Registration Failed 17](#_Toc458785310)

[6.2.3 Login Successful 18](#_Toc458785311)

[6.2.4 Login Failed 18](#_Toc458785312)

[6.2.5 Change Password Successful 19](#_Toc458785313)

[6.2.6 Change Password Failed 19](#_Toc458785314)

[6.2.7 Add Customer requirements](#_Toc458785315)

[6.2.8 Maintain Stock 20](#_Toc458785318)

[6.2.9 User Interface Testing: 21](#_Toc458785319)

[6.2.10 Pass or Fail Criteria 22](#_Toc458785320)

[6.3 TEST REPORTS 22](#_Toc458785321)

[6.3.1 Introduction 22](#_Toc458785322)

[6.3.2 Test Case Result Summary 22](#_Toc458785323)

[6.3.3 Detailed Test Results 22](#_Toc458785324)

[6.3.3.1 Manual Testing 22](#_Toc458785325)

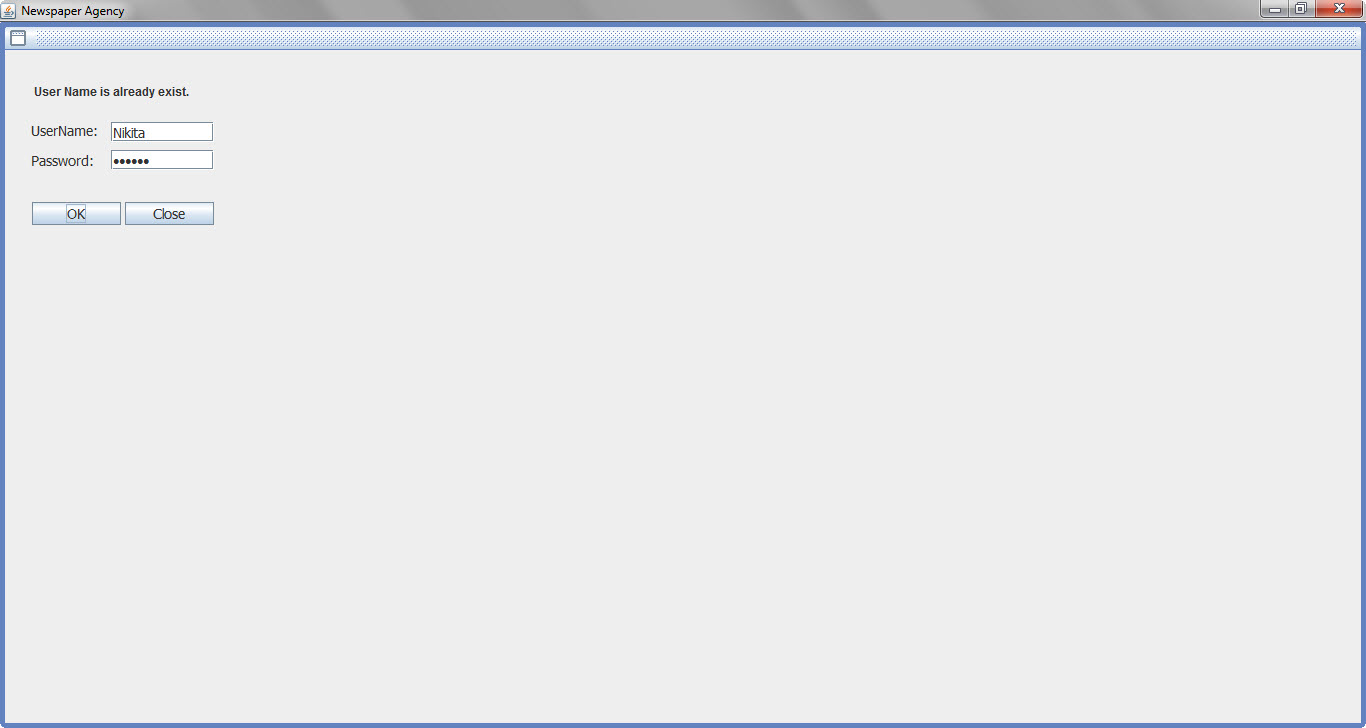
[6.3.3.2 TC # 1 – Registration 23](#_Toc458785326)

[6.3.3.3 TC # 2 – Login 23](#_Toc458785327)

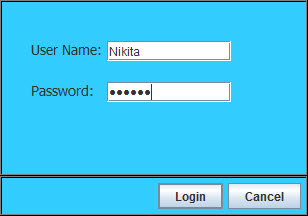
[6.3.3.4 TC # 5 – Admin Add/View Stock. 24](#_Toc458785330)

# **SCREEN SHOTS**

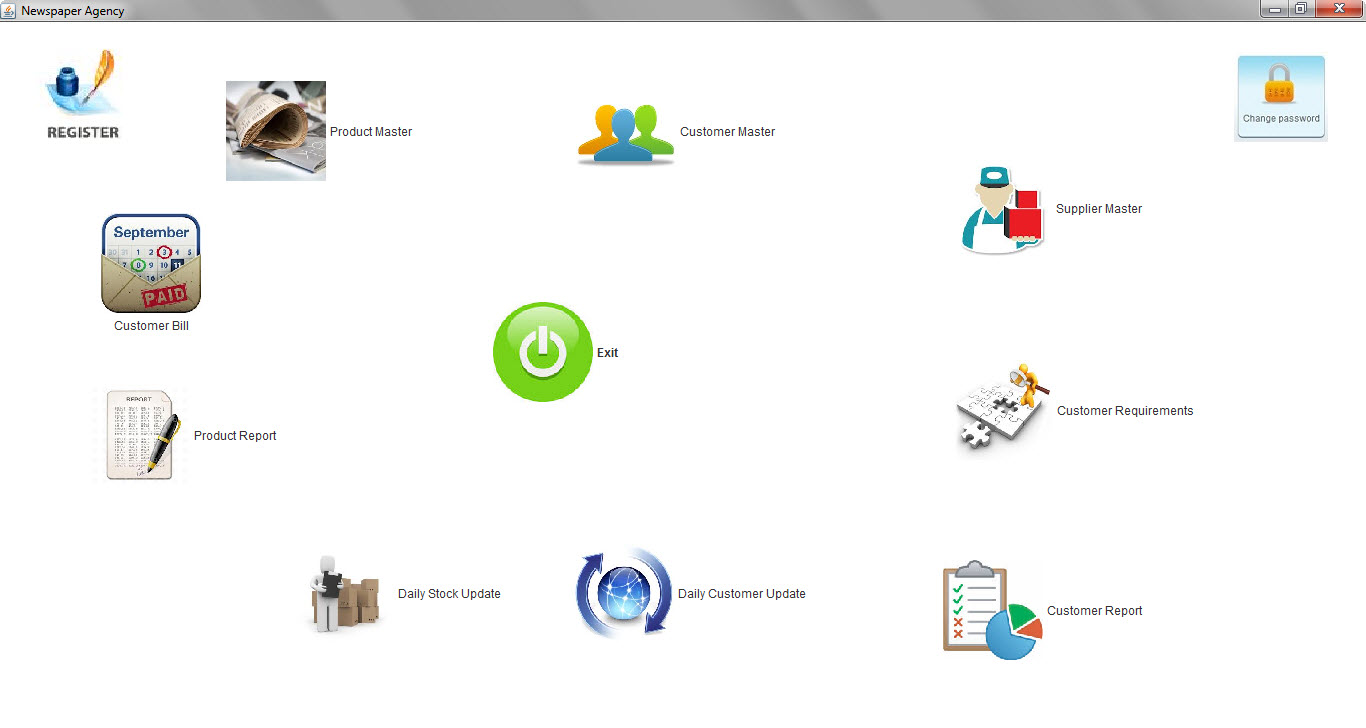
**New User Registration**



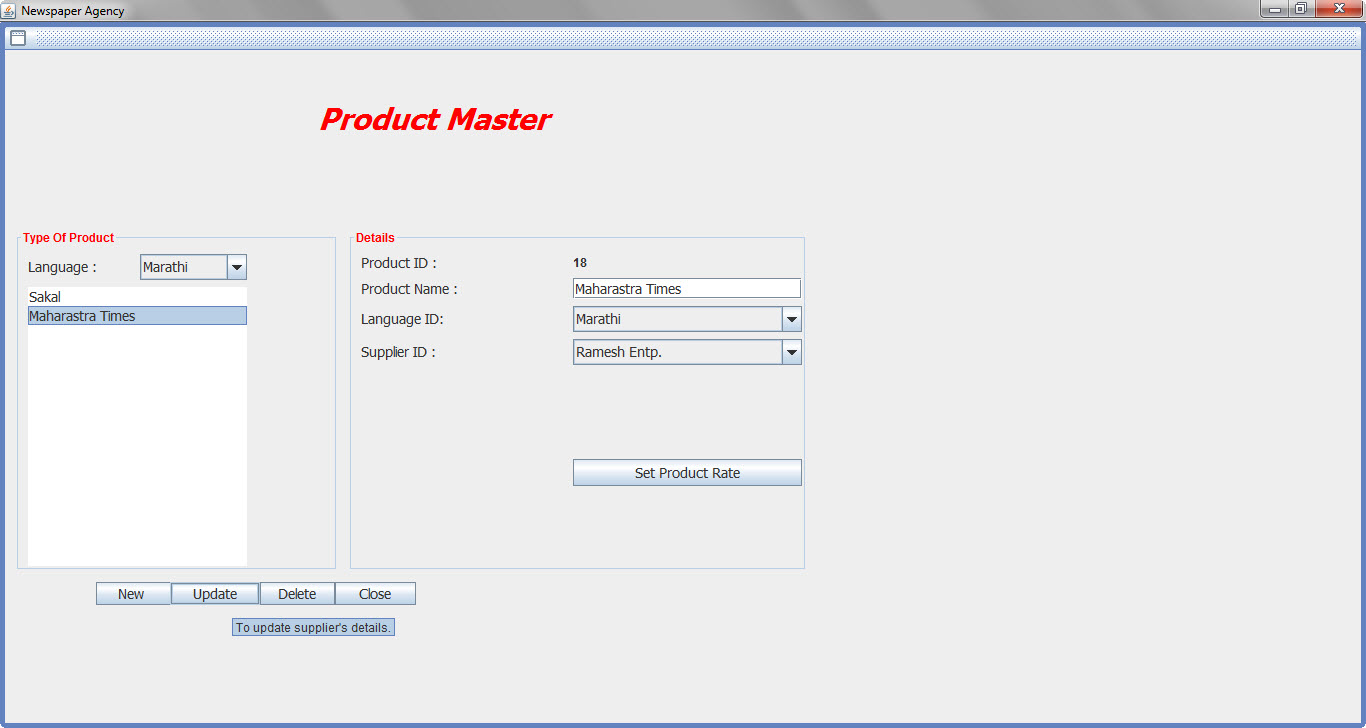
**User Login**



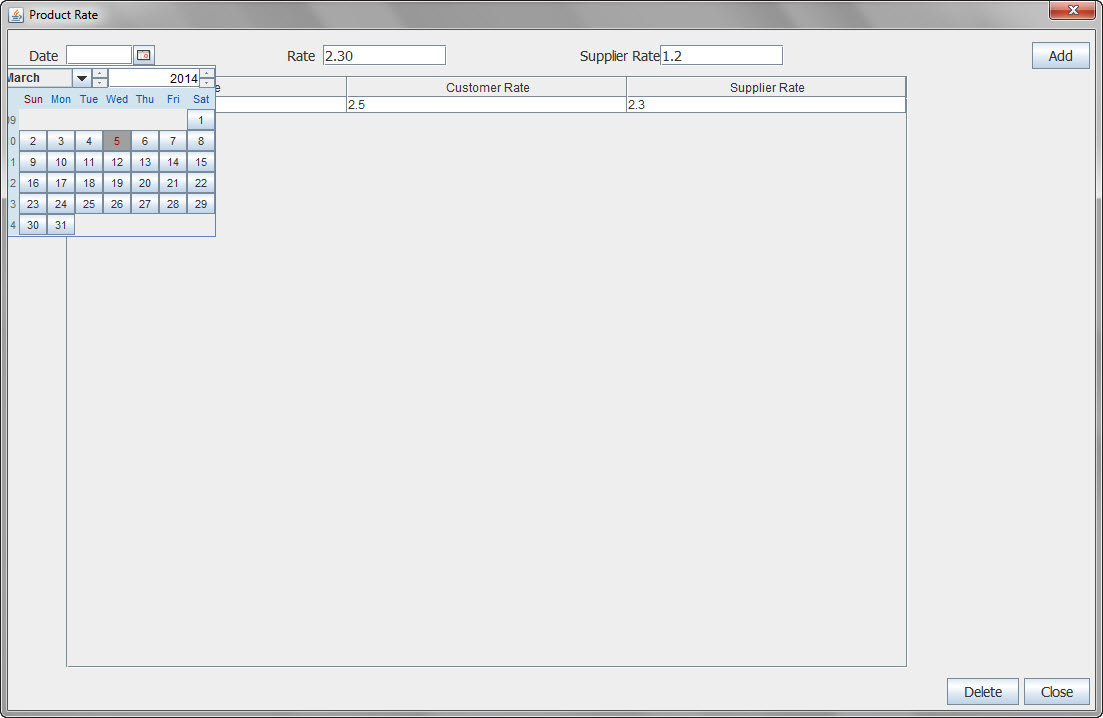
**Home Screen:**



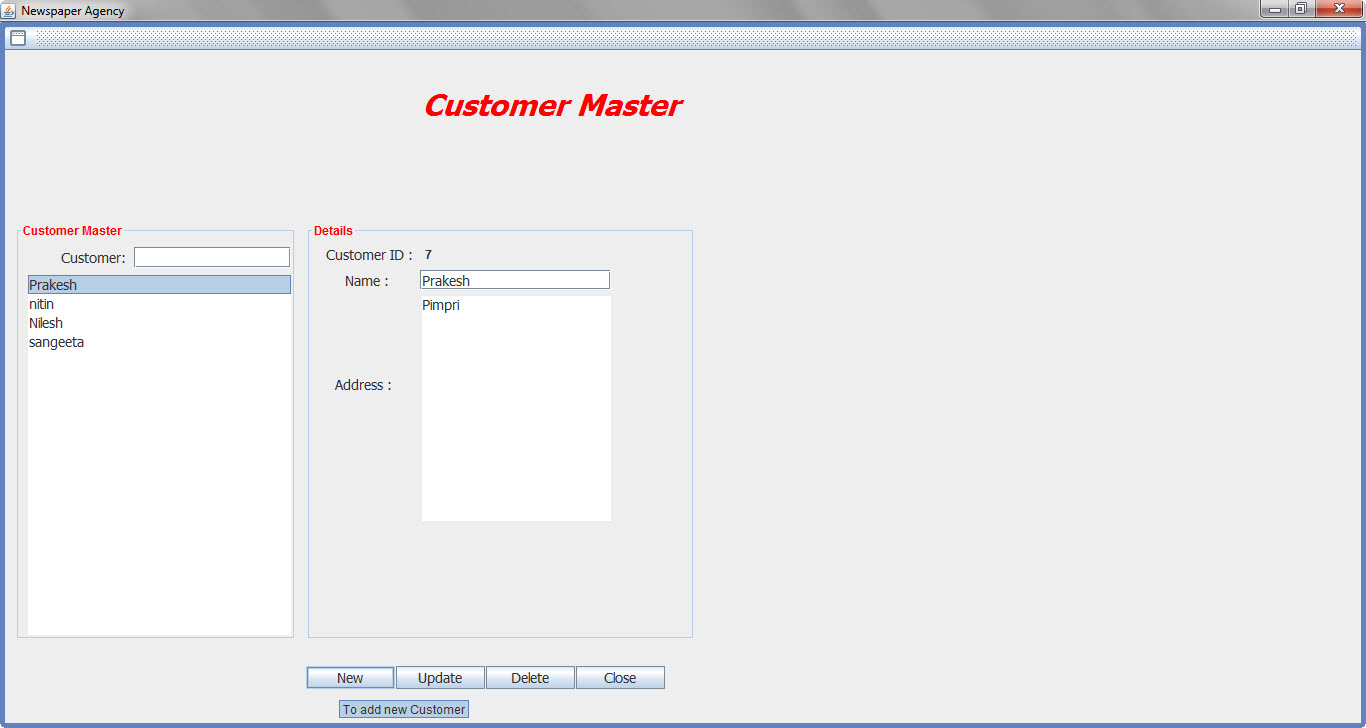
**Product Master:**



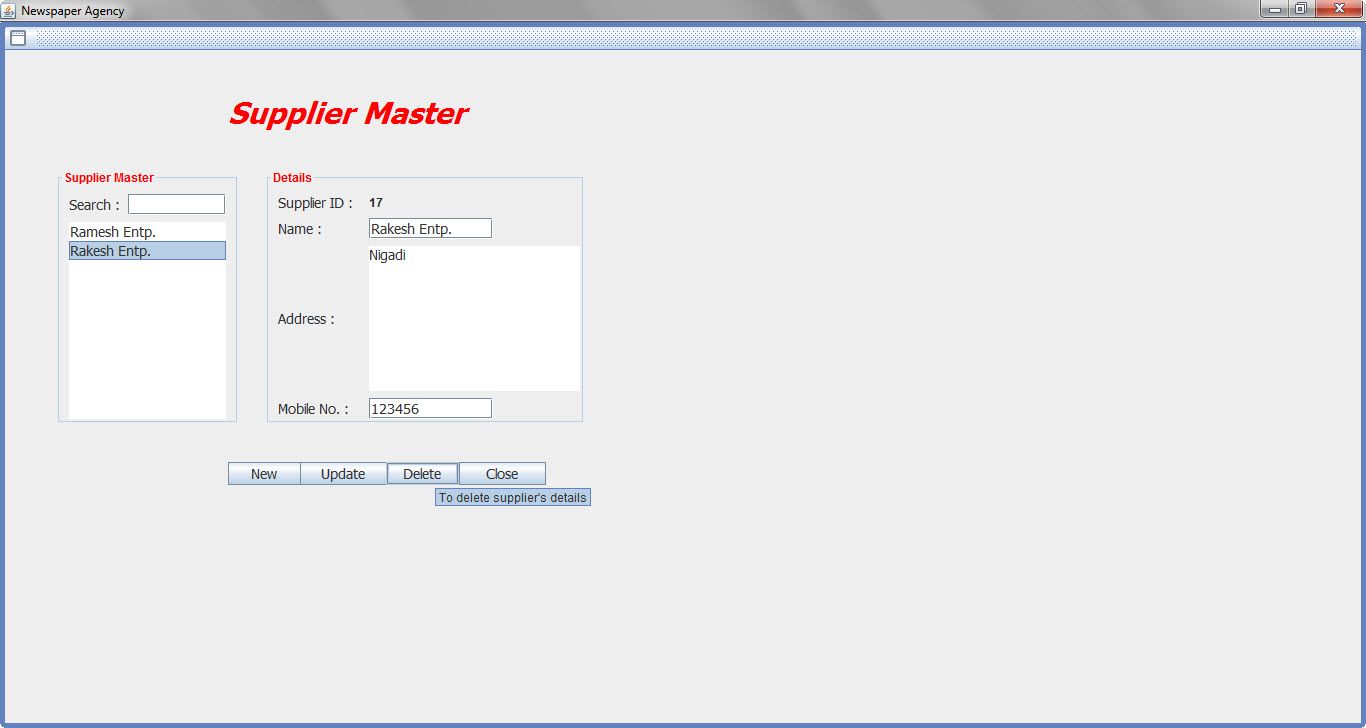
**Set Product Rate:**



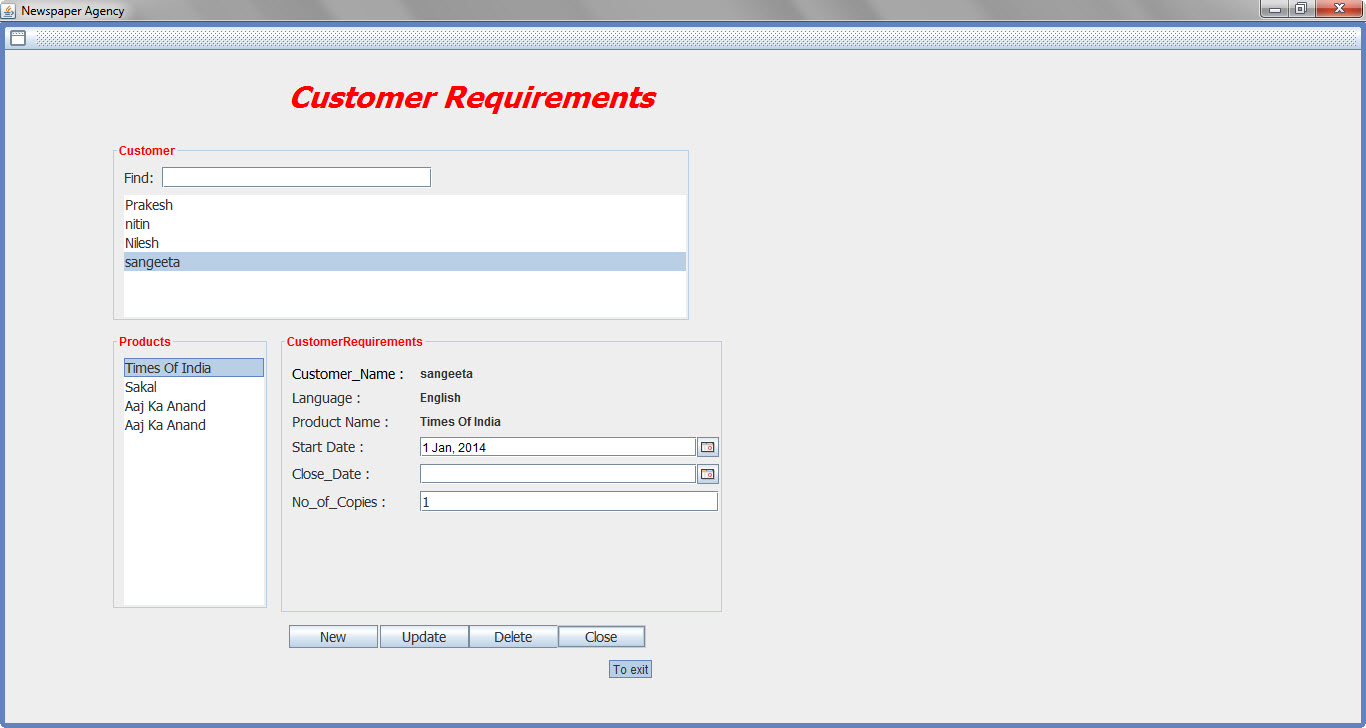
**Customer Master:**



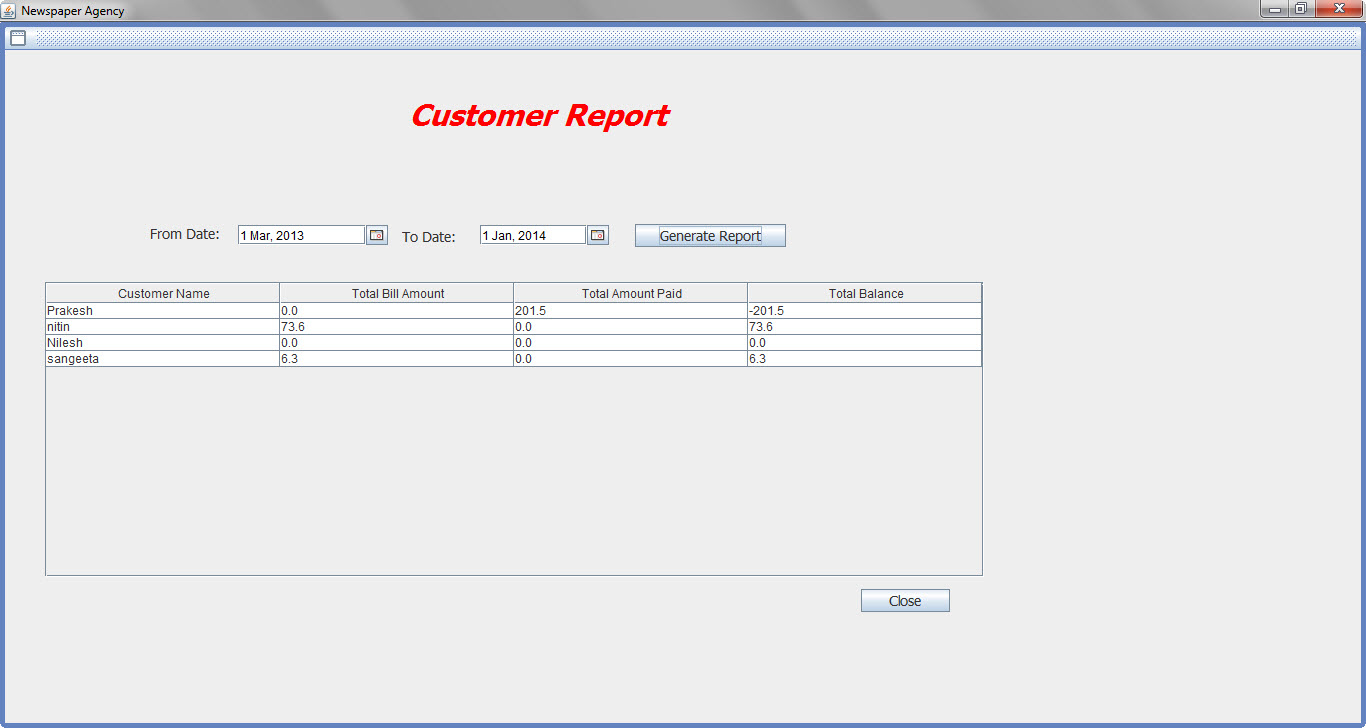
**Supplier Master:**



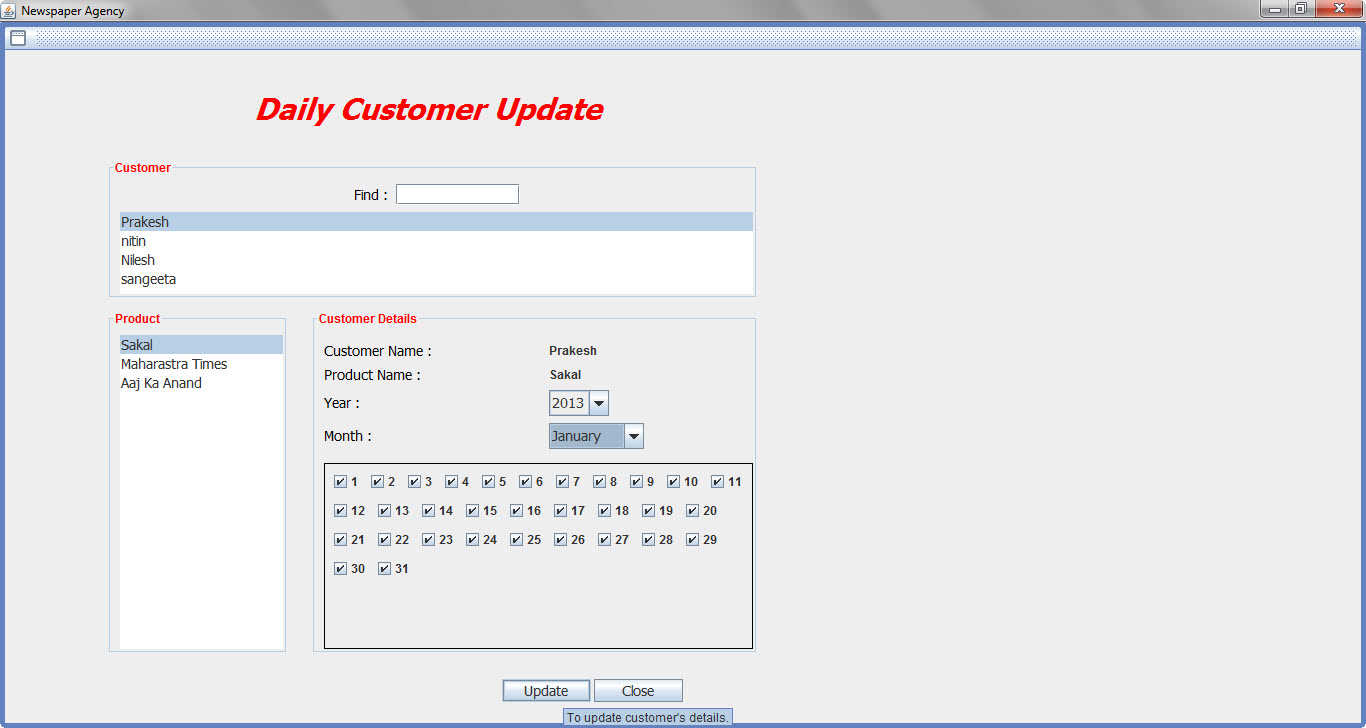
**Customer Requirements:**



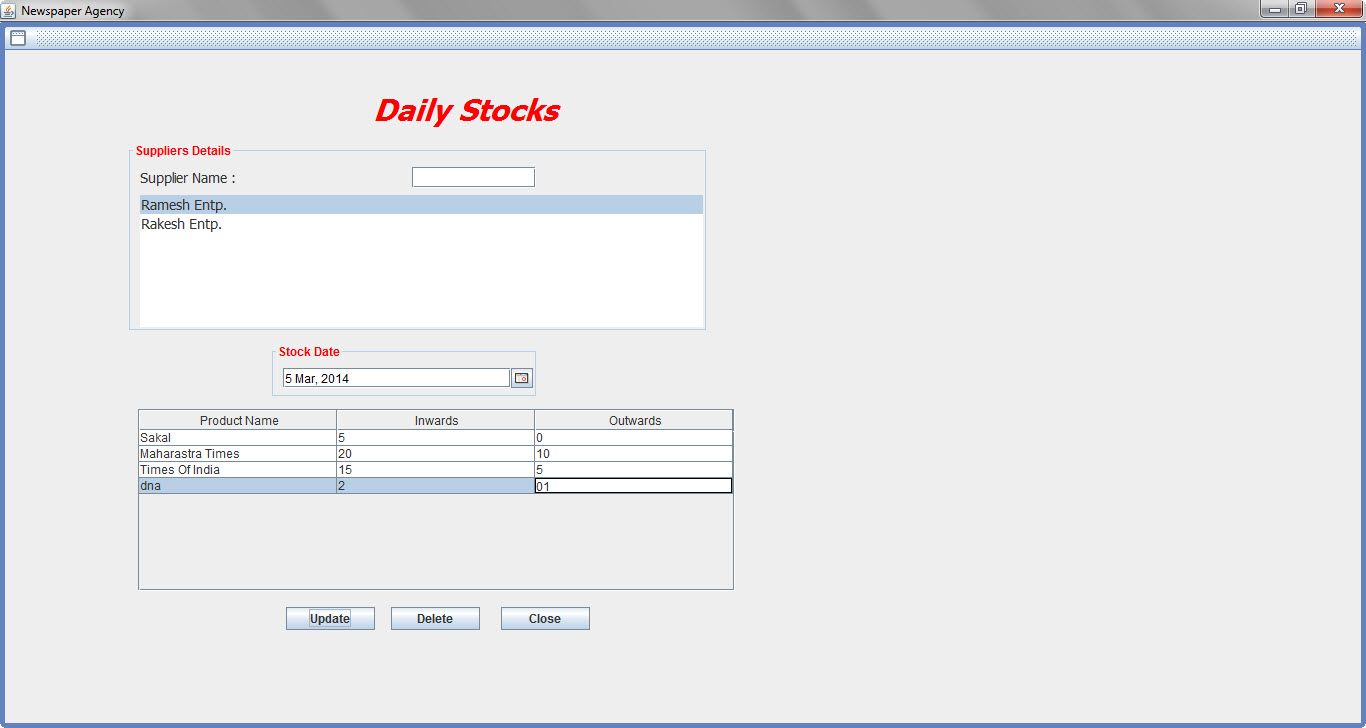
**Customer Report:**



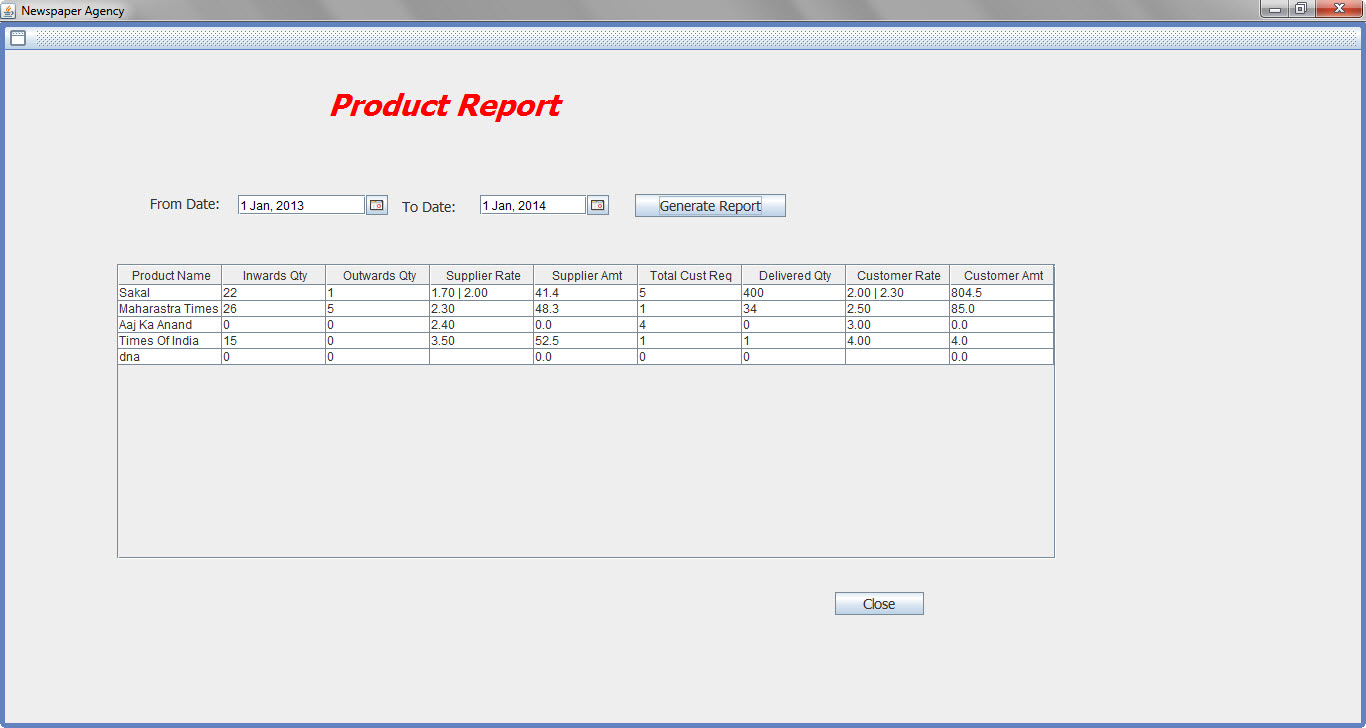
**Daily Customer Update:**



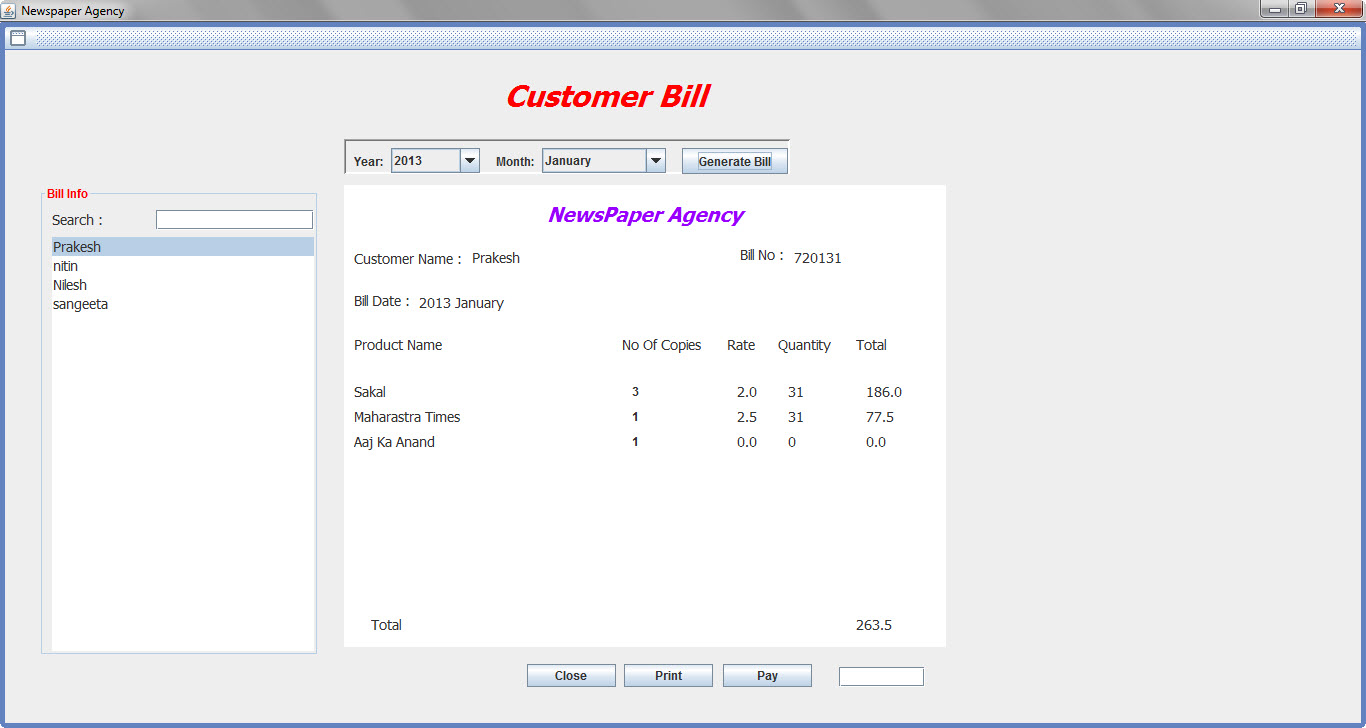
**Daily Stock Update:**



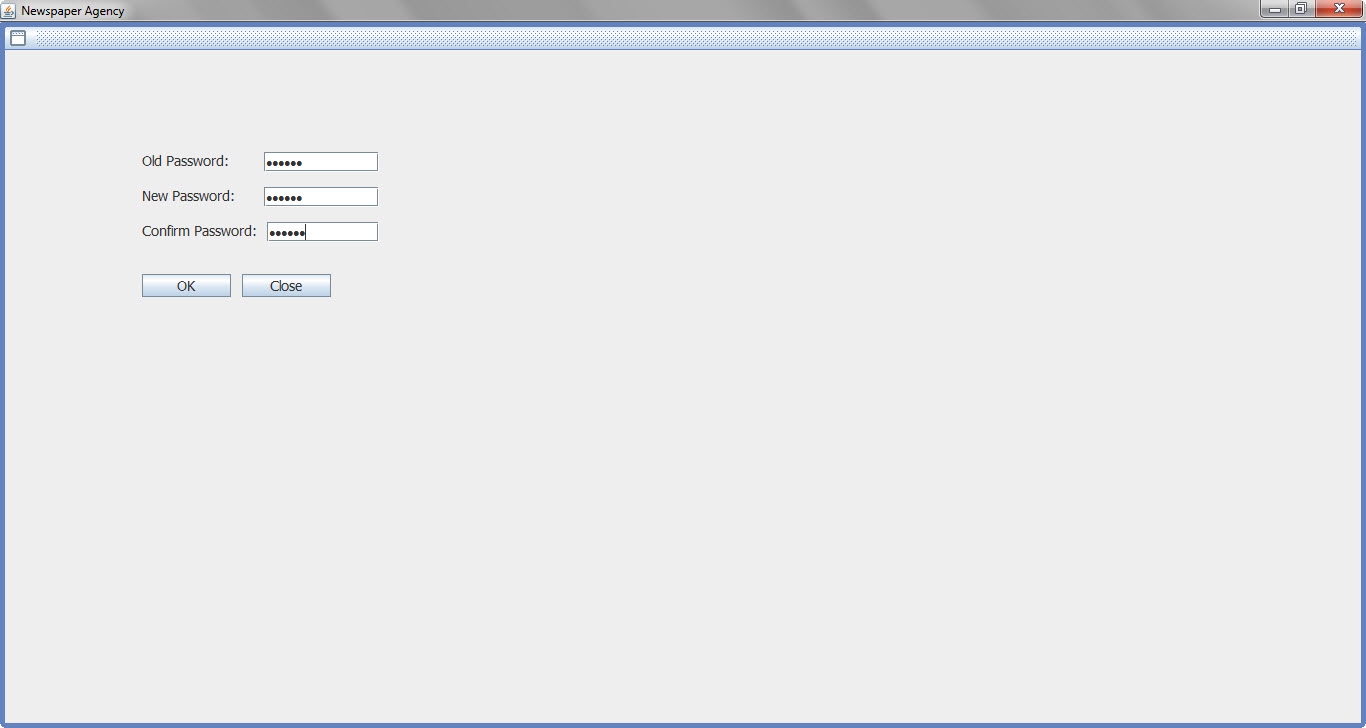
**Product Report:**



**Customer Bill:**



**Change Password:**



# **TESTING AND DOCUMENTATION**

# **TEST PLAN**

# **6.1.1 INTRODUCTION**

A Software Test Plan is a document describing the testing scope and activities. It is the basis for formally testing any software/product in a project. A document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process.

This is a Test Plan Document for Newspaper Management System (NPMS). The document outlines the test strategies, responsibilities and metrics. It also contains the current library of test cases that will be used for testing purposes and to keep development teams on track.

Potential users should focus on the user interface testing to be able to judge the usability.

Customer should have a thorough understanding of the document, as it will enable to have an even better understanding of the use cases that will be implemented and their black box working style.

# **6.1.2 OBJECTIVE**

The main objective of the Test Plan for the Newspaper Management System are as follows:

* To identify the feature of the system that will be tested.
* To identify and define all the activities necessary to prepare for and conduct the testing on the Newspaper Management System.
* To discuss the testing techniques being used to test the Newspaper Management System.

# **DEFINITIONS**

The following are some terms and definitions that are related to the test plan of the Newspaper Management System:

* **Pass/Fail Criteria:** Decision rules that are determine whether a software item passes or fails a test.
* **Test:** A collection of one or more test cases.
* **Test Item:** A software item that is an objective of testing.
* **Test Plan:** A document describing the scope, approach, resources, and schedule of the indented testing activities.
* **Test Summary Report:** A document summarizing the testing activity and results.
* **Testing:** The process of analyzing a software item to detect the differences between the existing and required conditions.

# **TEST ITEMS**

The section of the Test Plan lists the all items of the Newspaper Management System that will be tested:

* Registration
* Login
* Request for Newspaper
* Admin view/modify stock
* Change Password

# **APPROACH**

This section of the test plan describes the overall approach for testing the Newspaper Management System project. The approach followed for the testing the Newspaper Management System ensures that all the major feature of the project is adequately tested.

# **Unit Testing**

The Unit Testing is a test that tests each single module of the software to check for the errors. This is mainly done to discover errors in code of the Newspaper Management System. The main goal of the unit testing would be to isolate each part of the program and to check the correctness of the code. In the case of Newspaper Management System all the forms will be tested. There are many benefits for this unit testing:

* The Unit Testing facilitates change in the code.
* It allows testing to be done in a bottom up fashion.

At the same time, unit testing has some disadvantages such as it might not identify each and every error in the system.

# **Integration Testing:**

In Integration testing, the individual modules are combined and tested as a whole unit. The integration testing generally follows unit testing where each module is tested as a separate unit. The main purpose of the integration testing is to test the functional and performance requirements on the major items of the project. All the modules of the project developed individually would be combined together and tested as a whole system in the integration testing.

# **Regression Testing:**

The Regression testing is done whenever modifications are made to the source code of the project. The Regression testing can also be defined as a process of testing changes made to the computer program and also makes sure that the older programming still works with the new changes.

So, before any new version of the software product is released, the old test cases for the project will be run against the software with the changes made, to make sure that the old functionalities of the project still work.

# **Acceptance Testing:**

By performing acceptance tests on an application the testing team will deduce how the application will perform in production. This test mainly qualifies the project and decides if it will be accepted by the users of the system. The users or the customers of the project are responsible for the test.

# **System Testing:**

The system testing is mainly done on the whole integrated system to make sure that the project that has been developed meets all requirements. The test cases for the system testing will be the combination of unit and integration tests.

# **TEST DESIGN/TEST CASE SPECIFICATIONS**

# **Registration Successful**

|  |  |
| --- | --- |
| **Test Case** | **Registration Successful** |
| Actor | User |
| Pre-conditions | Database created, Interface created |
| Detailed Description | Tests if the correct information is entered in the database and user enters appropriate information |
| Test Procedure | Enter required data into input fields |
| Expected Results | Username will be verified in the database with the information submitted through the input fields from the interface |
| Expected Interface Output | Registration Successful |
| Test Results | Passed |

# **Registration Failed**

|  |  |
| --- | --- |
| **Test Case** | **Registration Successful** |
| Actor | User |
| Pre-conditions | Database created, Interface created |
| Detailed Description | Tests if the correct information is entered in the database and user enters appropriate information |
| Test Procedure | Enter required data into input fields with blank or wrong detail |
| Expected Results | Displays Invalid details message against incorrect input field. |
| Expected Interface Output | Registration Failed. |
| Test Results | Passed |

# **Login Successful**

|  |  |
| --- | --- |
| **Test Case** | **Login Successful** |
| Actor | Admin/User |
| Pre-conditions | Database created, Interface created |
| Detailed Description | Tests if the UserId is created in the database when Admin enters appropriate information |
| Test Procedure | Enter required data into input fields which are username and password |
| Expected Results | Username will be verified in the database with the information submitted through the input fields from the interface |
| Expected Interface Output | Login Successful |
| Test Results | Passed |

# **Login Failed**

|  |  |
| --- | --- |
| **Test ID** | **Failed to Login** |
| Actor | Admin/User |
| Pre-conditions | Database created, Interface created |
| Detailed Description | Tests if the UserId is created in the database when Admin/User enters appropriate information |
| Test Procedure | Enter required data into input fields with blank or wrong username and password |
| Expected Results | Username will be verified in the database and displays the message Invalid Username/Password. |
| Expected Interface Output | Login Failed |
| Test Results | Passed |

# **Change Password Successful**

|  |  |
| --- | --- |
| **Test Case** | **Change Password Successful** |
| Actor | Admin/User |
| Pre-conditions | Database created, Interface created |
| Detailed Description | Tests if the UserId is created in the database when Admin/Owner enters appropriate information |
| Test Procedure | Enter required data into input fields which are username and password |
| Expected Results | Username will be verified in the database with the information submitted through the input fields from the interface |
| Expected Interface Output | Change Password Successful |
| Test Results | Passed |

# **Change Password Failed**

|  |  |
| --- | --- |
| **Test Case** | **Change Password Failed** |
| Actor | Admin/User |
| Pre-conditions | Database created, Interface created |
| Detailed Description | Tests if the UserId is created in the database when Admin/Owner enters appropriate information |
| Test Procedure | Enter required data into input fields which are username and password |
| Expected Results | Password does not changed |
| Expected Interface Output | Change Password Failed |
| Test Results | Passed |

# **Add Customer requirements**

|  |  |
| --- | --- |
| **Test Case** | **Add Customer Requirements Successful** |
| Actor | Admin/User |
| Pre-conditions | Database created, Interface created, request details created already in database |
| Detailed Description | Tests if the stock available and generate token number once users post request with newspaper type and no of newspapers. |
| Test Procedure | Select required data from input fields which are newspaper type, language and number of papers. |
| Expected Results | Request is accepted and token number is generated |
| Expected Interface Output | Newspaper request Successful |
| Test Results | Passed |

# **Maintain Stock**

|  |  |
| --- | --- |
| **Test Case** | **Add/View Stock successful** |
| Actor | Admin |
| Pre-conditions | Database created, Interface created |
| Detailed Description | Tests if the Admin is able to add/view stock into the database. |
| Test Procedure | Add/View no of cylinder depending on type of cylinder |
| Expected Results | Database now contains added stock |
| Expected Interface Output | Stock addition successful |
| Test Results | Passed |

# **User Interface Testing:**

|  |  |
| --- | --- |
| **Test Case** | User Interface Creation |
| Actor | Admin/Owner |
| Pre-conditions | Database created |
| Detailed Description | Web view is correctly created |
| Test Procedure | Log in as an administrator.  Navigate to all the tabs and screens |
| Expected Results | The Java Swing pages should be all correctly served |
| Expected Interface Output | All forms are correctly displayed |
| Test Results | Not Tested |

|  |  |
| --- | --- |
| **Test Case** | Icon Rendering |
| Actor | Admin/Owner |
| Pre-conditions | Database created |
| Detailed Description | All icons in the web forms are correctly displayed |
| Test Procedure | Log in as an administrator.  Navigate to all the tabs and screens that have icons.  Verify that all icons are shown and that the right icons are shown in the right locations |
| Expected Results | The Java Swing pages should have all the right icons shown in the correct places |
| Expected Interface Output | All forms are correctly displayed |
| Test Results | Not Tested |

# **Pass or Fail Criteria**

The test cases executed on the Newspaper Management System will pass if they meet the specific requirements mention in the document. A Test cases said to be fail, if the desired functionality is not satisfied by the system.

# **TEST REPORTS**

# **Introduction**

The Test Results presents the results obtained by the Newspaper Management System.

# **Test Case Result Summary**

The summary of the Test case result has been depicted in the table shown below:

These are the test cases Admin page. The major of the testing has been concentrated on the Admin pages.

|  |  |  |
| --- | --- | --- |
| **Test Case #** | **Description** | **Results/Comments** |
| TC#1 | Registration | Passed |
| TC #2 | Admin Login | Passed |
| TC #3 | User request newspaper | Passed |
| TC #4 | User check status | Passed |
| TC #5 | Admin view/add stock | Passed |

# **Detailed Test Results**

# **Manual Testing**

To start with, I have performed manual testing on the Newspaper Management System. Manual Testing is one of the oldest and rigorous methods of software testing. This testing strategy gives the best opportunity to check every page thoroughly and make sure it works as expected manner. Due to the complexity of the various automation tools and time available for testing the entire application, I have preferred to use Manual Testing based on the fact that is one of the best methods of testing suggested for a beginner. All the test cases mentioned in the test plan will be tested here. The result of the manual testing are represented in the following tables:

# **TC # 1 – Registration**

|  |  |  |
| --- | --- | --- |
| **Test Unit** | **Test Case** | **Result** |
| **Registration Button** | Wrong format of information entered into the data fields of the Registration page | The system generates an error message to the user indicating that the wrong format of data is entered and to re-enter the data. |
| **Registration Button** | A valid input field entered by the user | The users gets registered successfully once the admin verifies the details. |

# **TC # 2 – Login**

|  |  |  |
| --- | --- | --- |
| **Log In Button** | An invalid username or password entered by the user | The system generates a message “invalid user” or “invalid password” |
| **Log In Button** | A valid username or password entered by the user | The system logs on the user and transfers him to the home page. |

# **TC # 5 – Admin Add/View Stock.**

|  |  |  |
| --- | --- | --- |
| **Test Unit** | **Test Case** | **Result** |
| **Admin Add Stock** | Selected valid cylinder type and cylinder no | The system generates a message: “Inventory Updated” and added to database. |
| **Admin View Stock** | Admin click on View Stock link | Admin can view the stock details. |