

# Project Based Learning Report

On

## **“Testing of Canteen Management System”**

Submitted in the partial fulfillment of the requirements.

Software Testing

in

Electronics & Communication Engineering

By

|               |                 |
|---------------|-----------------|
| AASHISH KUMAR | PRN-2014111065  |
| SAURAV KUMAR  | PRN-2014111071  |
| NISHAL KUMAR  | PRN- 2014111072 |

Under the guidance of Course In-charge

Prof. P.A Chougale

Department of Electronics & Communication Engineering

Bharati Vidyapeeth  
(Deemed to be University)  
College of Engineering,  
Pune – 4110043

Academic Year: 2023-24

## **CERTIFICATE**

This is to be Certified that the Project Based Learning report entitled, “Testing of Canteen Management System” Work is done by

|               |                 |
|---------------|-----------------|
| AASHISH KUMAR | PRN-2014111065  |
| SAURAV KUMAR  | PRN-2014111071  |
| NISHAL KUMAR  | PRN- 2014111072 |

In partial fulfillment of the requirements for the award of credits for Project Based Learning (PBL) in Software Testing Bachelor of Technology Semester VIII,  
Electronics and Communication Engineering

**Date: 01-04-2024**

**Prof. P.A Chougale**  
**Course In-charge**

**Dr. Arundhati A. Shinde**  
**Professor & Head**

# **ABSTRACT**

The Canteen Management System is a web-based program that can be used by an administrator to streamline and simplify the operations of a canteen or cafeteria. It offers a comprehensive solution for handling menu planning, order placing, inventory management, billing, reporting, and more. The admin can quickly navigate through the various features and functionalities of the web application thanks to its user-friendly design. The admin can also design and oversee promotions, discounts, and package deals. Customers can place orders easily thanks to an easy online ordering process.

The system has effective inventory management, which enables the administrator to monitor stock levels, trigger low stock warnings, and produce reports on stock utilization. Additionally, the system has strong billing and reporting capabilities that let the administrator create invoices, receipts, and sales, revenue, and inventory reports. The system is suitable for small to large food service enterprises because it is expandable and can support several canteen locations.

# Table of Contents

|  |    |
|--|----|
| 1) Problem Statement .....                   | 6  |
| 2) Objective and scope of the software ..... | 8  |
| 3) Software Testing... ..                    | 9  |
| 4) Software Testing and Lifecycle... ..      | 13 |
| 5) Required Specifications .....             | 19 |
| 6) Test planning... ..                       | 32 |
| 7) Conclusion... ..                          | 41 |

# 1. Problem Statement

- **What is Canteen Management System?**

The traditional canteen management system in many institutions and organizations often relies on manual processes, which can be time-consuming, inefficient, and error-prone. These manual processes include taking orders, managing inventory, tracking payments, and generating reports, among others. This can result in delays, inaccuracies, and lack of transparency in the canteen operations, leading to customer dissatisfaction and financial losses.

## **Description of Topic:**

A canteen management system web application is designed to streamline and automate the operations of a canteen or cafeteria in an institution or organization. The web application provides a user-friendly interface for canteen staff to manage various tasks, such as taking orders, tracking inventory, managing payments, generating reports, and maintaining customer records. The system can also provide an interface for customers to place orders online, view menu items, make payments, and track their orders.

The canteen management system web application can be accessed from any device with internet connectivity, making it convenient for canteen staff and customers to use. It can also be integrated with other systems, such as the institution's accounting or inventory management system, for seamless data exchange and improved efficiency.

Some key features of a canteen management system web application may include menu management, order management, inventory management, payment management, reporting and analytics, and customer management. These features can help automate processes, reduce errors, improve customer experience, and provide real-time insights into the canteen operations.

Overall, a canteen management system web application aims to modernize and optimize canteen operations, improve customer service, and enhance financial management for institutions and organizations with canteen facilities.

- System Used: Canteen Management System :

Canteen Management system is a web application developed by Mayuri K. is an application that is operated by the administrator to maintain record of canteen essentials. It includes features like adding customer to generating invoices of the order done, it contains a food module to decide the dish simultaneously it has foodcategory to decide the type of food that you want to take.

- Modules taken for the testing-

- Login / Signup
- Customer

- Task to be Performed by Administrator–

- Login
- Selecting the required food items using food module
- Customer adding in the database using customer module
- Invoice generation using invoice module
- Getting average day sale using pie chart
- Logout

## **2. Objective and Scope of the Software:**

The objective of a canteen management web application operated by an administrator could be to improve the efficiency and effectiveness of managing a canteen. The scope of such a web application could include:

1. Menu management: The web application can allow the administrator to manage the canteen's menu, including adding new items, removing existing ones, and updating prices.
2. Order management: The web application can enable the administrator to manage orders placed by customers, including order tracking, order cancellation, and refunds.
3. Inventory management: The web application can help the administrator to keep track of the inventory, including the stock levels of food items and supplies, and manage re-ordering.
4. Payment management: The web application can allow the administrator to manage payments, including processing payments and generating invoices.
5. User management: The web application can help the administrator to manage users, including customer registration, login, and authentication, and employee management.
6. Reporting and analytics: The web application can provide real-time data on the canteen's sales, customer behavior, and other key performance indicators (KPIs) to help the administrator make data-driven decisions and improve the overall efficiency and profitability of the canteen.

## Software Testing :

*Software testing* can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases.

The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy, and usability. It mainly aims at measuring the specification, functionality, and performance of a software program or application.

*Software testing can be divided into two steps:*

1. **Verification:** it refers to the set of tasks that ensure that the software correctly implements a specific function.

2. **Validation:** it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

Verification: “Are we building the product right?”

Validation: “Are we building the right product?”



***Software Testing can be broadly classified into two types:***

**1. Manual Testing:** Manual testing includes testing software manually, i.e., without using any automation tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.

Testers use test plans, test cases, or test scenarios to test software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

**2. Automation Testing:** Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves the automation of a manual process. Automation Testing is used to re-run the test scenarios quickly and repeatedly, that were performed manually in manual testing.

Apart from regression testing, automation testing is also used to test the application from a load, performance, and stress point of view. It increases the test coverage, improves accuracy, and saves time and money when compared to manual testing.

***Software testing techniques can be majorly classified into two categories:***

**1. Black Box Testing:** The technique of testing in which the tester doesn't have access to the source code of the software and is conducted at the software interface without any concern with the internal logical structure of the software is known as black-box testing.

**2. White-Box Testing:** The technique of testing in which the tester is aware of the internal workings of the product, has access to its source code, and is conducted by making sure that all internal operations are performed according to the specifications is known as white box testing.

***What are different levels of software testing?***

| <b>Black Box Testing</b>                              | <b>White Box Testing</b>                                   |
|---|--|
| Internal workings of an application are not required. | Knowledge of the internal workings is a must.              |
| Also known as closed box/data-driven testing.         | Also known as clear box/structural testing.                |
| End users, testers, and developers.                   | Normally done by testers and developers.                   |
| This can only be done by a trial-and-error method.    | Data domains and internal boundaries can be better tested. |

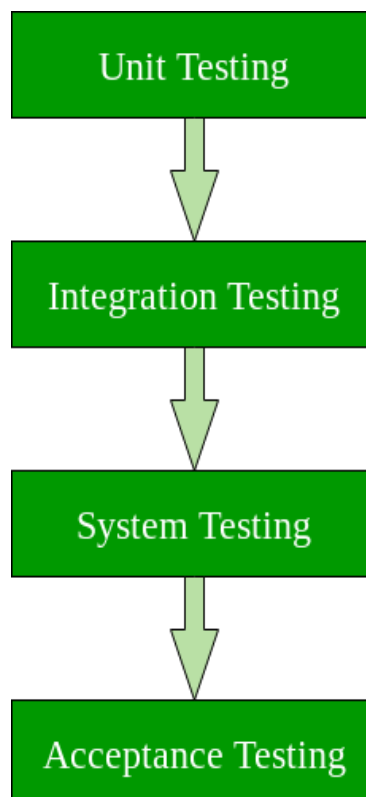
## Software level testing can be majorly classified into 4 levels:

**1. Unit Testing:** A level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.

**2. Integration Testing:** A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.

**3. System Testing:** A level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

**4. Acceptance Testing:** A level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.



## **Software Testing Lifecycle :**

The Software Testing Life Cycle (STLC) is a systematic approach to testing a software application to ensure that it meets the requirements and is free of defects. It is a process that follows a series of steps or phases, and each phase has specific objectives and deliverables. The STLC is used to ensure that the software is of high quality, reliable, and meets the needs of the end-users.

The main goal of the STLC is to identify and document any defects or issues in the software application as early as possible in the development process. This allows for issues to be addressed and resolved before the software is released to the public.

The stages of the STLC include Test Planning, Test Analysis, Test Design, Test Environment Setup, Test Execution, Test Closure, and Defect Retesting. Each of these stages includes specific activities and deliverables that help to ensure that the software is thoroughly tested and meets the requirements of the end users.

Overall, the STLC is an important process that helps to ensure the quality of software applications and provides a systematic approach to testing. It allows organizations to release high-quality software that meets the needs of their customers, ultimately leading to customer satisfaction and business success.

### **Characteristics of STLC :**

STLC is a fundamental part of the Software Development Life Cycle (SDLC) but STLC consists of only the testing phases.

STLC starts as soon as requirements are defined or software requirement document is shared by stakeholders.

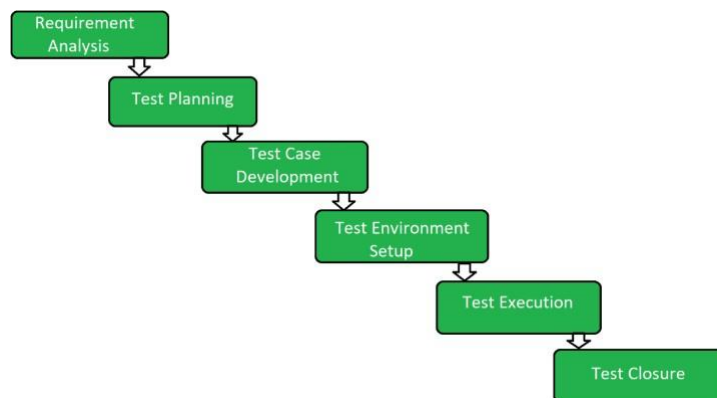
STLC yields a step-by-step process to ensure quality software.

## **Phases of STLC :**

1. **Requirement Analysis:** Requirement Analysis is the first step of the Software Testing Life Cycle (STLC). In this phase quality assurance team understands the requirements like what is to be tested. If anything is missing or not understandable then the quality assurance team meets with the stakeholders to better understand the detailed knowledge of requirements.

2. **Test Planning:** Test Planning is the most efficient phase of the software testing life cycle where all testing plans are defined. In this phase manager of the testing, team calculates the estimated effort and cost for the testing work. This phase gets started once the requirement-gathering phase is completed.

3. **Test Case Development:** The test case development phase gets started once the test planning phase is completed. In this phase testing team notes down the detailed test cases. The testing team also prepares the required test data for the testing. When the test cases are prepared then they are reviewed by the quality assurance team.



4. **Test Environment Setup:** Test environment setup is a vital part of the STLC. Basically, the test environment decides the conditions on which software is tested. This is independent activity and can be started along with test case development

5. **Test Execution:** After the test case development and test environment setup test execution phase gets started. In this phase testing team starts executing test cases based on prepared test cases in the earlier step.

*The activities that take place during the test execution stage of the Software Testing Life Cycle (STLC) include:*

- **Test execution:** The test cases and scripts created in the test design stage are run against the software application to identify any defects or issues.
- **Defect logging:** Any defects or issues that are found during test execution are logged in a defect tracking system, along with details such as the severity, priority, and description of the issue.
- **Test data preparation:** Test data is prepared and loaded into the system for test execution
- **Test environment setup:** The necessary hardware, software, and network configurations are set up for test execution
  - **Test execution:** The test cases and scripts are run, and the results are collected and analyzed.
  - **Test result analysis:** The results of the test execution are analyzed to determine the software's performance and identify any defects or issues.
  - **Defect retesting:** Any defects that are identified during test execution are retested to ensure that they have been fixed correctly.
  - **Test Reporting:** Test results are documented and reported to the relevant stakeholders.

6. **Test Closure:** Test closure is the final stage of the Software Testing Life Cycle (STLC) where all testing-related activities are completed and documented. The main objective of the test closure stage is to ensure that all testing-related activities have been completed and that the software is ready for release.

At the end of the test closure stage, the testing team should have a clear understanding of the software's quality and reliability, and any defects or issues that were identified during testing should be resolved.

### **3. Required Specifications for Testing :**

*Software required for the Testing purpose:*

#### **➤ Selenium IDE-**

Selenium IDE (Integrated Development Environment) is primarily a record/run tool that a test case developer uses to develop Selenium Test cases. Selenium IDE is an easy-to-use tool from the Selenium Test Suite and can even be used by someone new to developing automated test cases for their web applications. One does not require any special setup to get started with Selenium IDE. You just need to add the extension of your specific browser. Selenium IDE provides you with a GUI (Graphical User Interface) for easily recording your interactions with the website or the software.

#### **Features of Selenium IDE**

There are several features provided in the IDE under the toolbar, using which one can control the execution of test cases:

1. **Speed Control** – Helps control the speed of test cases
2. **Run All** – Allows execution of the entire Test Suite
3. **Run** – Runs the currently selected test
4. **Pause/Resume** – Allows a user to pause and resume a particular test case
5. **Step** – Helps step into each specific command in the test script
6. **Rollup** – Helps group all the Selenese Commands together and make them execute as a single operation. The features keep on getting eliminated or added depending on the usage of different versions of Selenium IDE extensions.
7. **Record and play** - Selenium IDE (Integrated Development Environment) provides a "record and play" feature that allows testers to create automated test scripts by recording their interactions with a web application and then playing back those interactions as a test script.

### ***Benefits of Using Selenium IDE :***

- Provides you the capability of automatically recording your test cases based upon the interactions with the browser
- Gives developers greater flexibility in executing the test cases. Either the test developer can run the entire test suite consisting of multiple test cases or execute a single test case
- Operates on the basis of the rich set of Selenese commands, which helps the IDE understand what needs to be done
- Allows the test developers to set breakpoints for the purpose of debugging particular test cases
- Test cases can be re-used using the run command. (e.g. allowing you to re-use the logic of login or reload on multiple places in the entire suite)
- Use of multiple-locators for each element in the IDE ensures successful execution.

### ***Limitations of Selenium IDE:***

- Not suitable for testing extensive data
- Connections with the database cannot be tested
- Cannot handle the dynamic part of web-based applications
- Does not support capturing of screenshots on test failures
- No feature available for generating result reports

### **Other Tools of selenium**

1. **Selenium WebDriver:** This is the most commonly used component of Selenium and provides a programming interface for interacting with web browsers. WebDriver allows developers to write code in their preferred programming language (such as Java, C#, Python, or Ruby) to automate browser actions, such as clicking buttons, filling forms, navigating between pages, and extracting data from web pages. WebDriver supports multiple .



2. **Selenium Grid:** Selenium Grid is a distributed testing framework that allows tests to be executed on multiple browsers, operating systems, and machines in parallel. It enables scalable and efficient test execution across different environments, making it suitable for large-scale web testing or cross-browser testing. Selenium Grid allows testers to create a test configuration specifying the desired combination of browsers, OSs, and machines, and then distribute the tests across these configurations for faster and more comprehensive testing.
3. **Selenium Remote Control (RC):** Selenium RC is the older version of Selenium that has been deprecated and is no longer actively maintained. It provides a server-based architecture for remote browser automation and allows tests to be written in multiple programming languages. However, Selenium RC has been largely replaced by WebDriver due to its limitations in terms of speed, stability, and ease of use.

## **4. Test Planning :**

In software testing, documentation is very important. Testing should be documented to provide efficient resource control monitoring. For successful testing, a test plan plays a very important role.

### **1. Test Plan**

A test plan is a document that consists of all future testing-related activities. It is prepared at the project level and in general, it defines work products to be tested, how they will be tested, and test type distribution among the testers. Before starting testing there will be a test manager who will be preparing a test plan. In any company whenever a new project is taken up before the tester involves in the testing the test manager of the team would prepare a test Plan.

### **2. Importance of Test Plan**

The following are some of the key benefits of making a test plan:

- It acts as a quick guide for the testing process.
- It helps to avoid out-of-scope functionalities.
- It determines the time, cost, and effort.

### **3. Test Plan Guidelines**

- Avoid Overlapping and repetition.
- Avoid Lengthy Paragraph.
- Use lists and tables.
- Update plan.



### ***A. Test Strategy :***

Example: In an application A, B, C, D features have to be developed, but the B feature has already been designed by other companies. So the development team will purchase B from that company and perform only integrated testing with A, B, C.

***B. Testing Methodology:*** The methods that are going to be used for testing depend on application to application.

***C. Approach:*** The approach of testing different software is different. It deals with the flow of applications for future references.

***D. Assumptions:*** In this phase, certain assumptions will be made. Example:

- The testing team will get proper support from the development team.
- The tester will get proper knowledge transfer from the development team.

***E. Risk:*** All the risks that can happen if the assumption is breaking. For Example, in the case of wrong budget estimation, the cost may overrun.

***F. Backup/Mitigation Plan-*** If any risk is involved then the company must have a backup plan, the purpose is to avoid errors.

### ***G. Roles and Responsibilities:***

Example:

- Test Manager: Manages the project, takes an appropriate resource and gives project direction.
- Tester: Identify the testing technique, verify the test approach, and save project cost.

***H. Scheduling:*** For Example, writing test case date and ending test case date.

**I. Defect Tracking:** There are the following methods for the process of defect tracking:

- Information Capture: In this, we take basic information to begin the process.
- Prioritize: The task is prioritized based on severity and importance.
- Communicate: Communication between the identifier of bug and fixer of bug.
- Environment: Test the application based on hardware and software.

Example: The bug can be identified using bug tracking tools such as Jira, Mantis, Trac.

**J. Test Environment-**

Example:

- Software configuration on different operating systems, such as Windows, Linux, Mac, etc.
- Hardware Configuration depends on RAM, ROM, etc.

**K. Entry and Exit Criteria:** The set of conditions that should be met in order to start any new type of testing or to end any kind of testing.

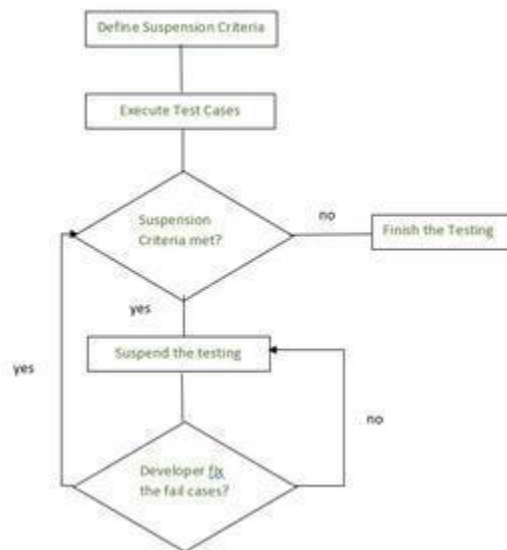
Entry Condition:

- Necessary resources must be ready.
- The application must be prepared.
- Test data should be ready. Exit

Condition:

- There should not be any major bug.
- Most test cases should be passed.
- When all test cases are executed.

Example: If the team member report 45% of the test cases failed, then testing willbe suspended until the developer team fixes all defects.



*Flow chart showing entry and exit condition*

**L. Test Automation:** It consists of the features that are to be automated and which features are not to be automated.

- If the feature has lots of bugs then it is categorized as Manual Testing.
- If the feature is frequently tested then it can be automated.

**M. Deliverables-** It is the outcome from the testing team and that is to be given to the customers at the end of the project.

Before testing phase:

- Test plan document.
- Test case document.
- Test design specification.

During testing phase:

- Test scripts.
- Test data.
- Error logs.

After testing phase:

- Test Reports.
- Defect Report.
- Installation Report.

*N. Templated:* It is followed by every kind of report that is going to be prepared by the testing team.

## **5. IMPLEMENTATIONS :**

Using Selenium IDE

## **SOFTWARE: Youthappam**

### **TEST PLAN for Login Module**

#### **1.TEST PLAN IDENTIFIER-**

Test\_login

#### **2.REFERENCES:**

SRS, Use Case Document, Test Strategy, Project Plan, Project Guidelines etc.

#### **3.INTRODUCTION:**

The objective of this test plan is to check the functionality of the Canteen Management System. Here all the functions such as ordering of meal, payment for the order etc test plan we have made.

#### **4. TEST ITEM –**

Login Module

#### **5. SOFTWARE RISK ISSUES –**

- Regular testing and evaluation of the system should be conducted to ensure its accuracy.
- Data Privacy and Security, use appropriate encryption techniques to protect user data.
- Performance and Scalability, the recommendation system should be designed to handle large volumes of data and user traffic.

#### **6. FEATURES TO BE TESTED-**

- Login Functionality

#### **5. FEATURES NOT TO BE TESTED-**

- Database
- Excel tools



## **6.TEST APPROACH-**

The test approach will involve a combination of manual and automated testing techniques

## **7.TEST PASS FAIL CRITERIA-**

Pass/Fail criteria is the predetermined conditions that must be met for a test case to be considered successful (pass) or unsuccessful (fail). These criteria are used to determine the overall quality and readiness of the application for deployment.

- Pass: Login functionality of the web application are working as expected, and there are no critical defects.
- Fail: Login functionality are not working as expected, or there are critical defects that affect the core functionalities of the web application.

## **8.SUSPENSION CRITERIA –**

These criteria are typically based on critical issues or risks, such as data corruption, or system failures, that require immediate attention before testing can continue.

It should suspend if the login page is not working even after entering correct credentials and getting reloaded again and again.

## **9.DELIVERABLES –**

including test reports, defect reports, test scripts and other documentation.

## **10.ENVIRONMENTALREQUIREMENTS-**

Google, Active Internet, Selenium IDE

## **11.SCHEDULE A TSET-**

Testing is schedule on 21 April,2023.

## LOGIN

Login: All credentials are correct.

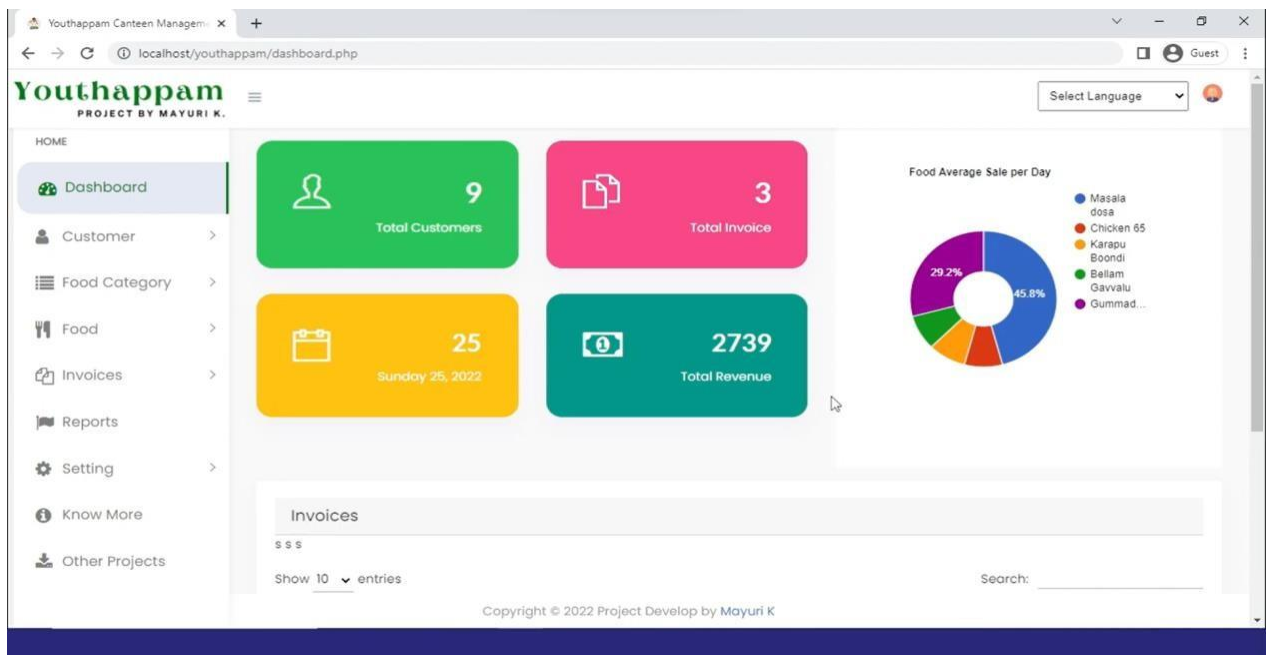
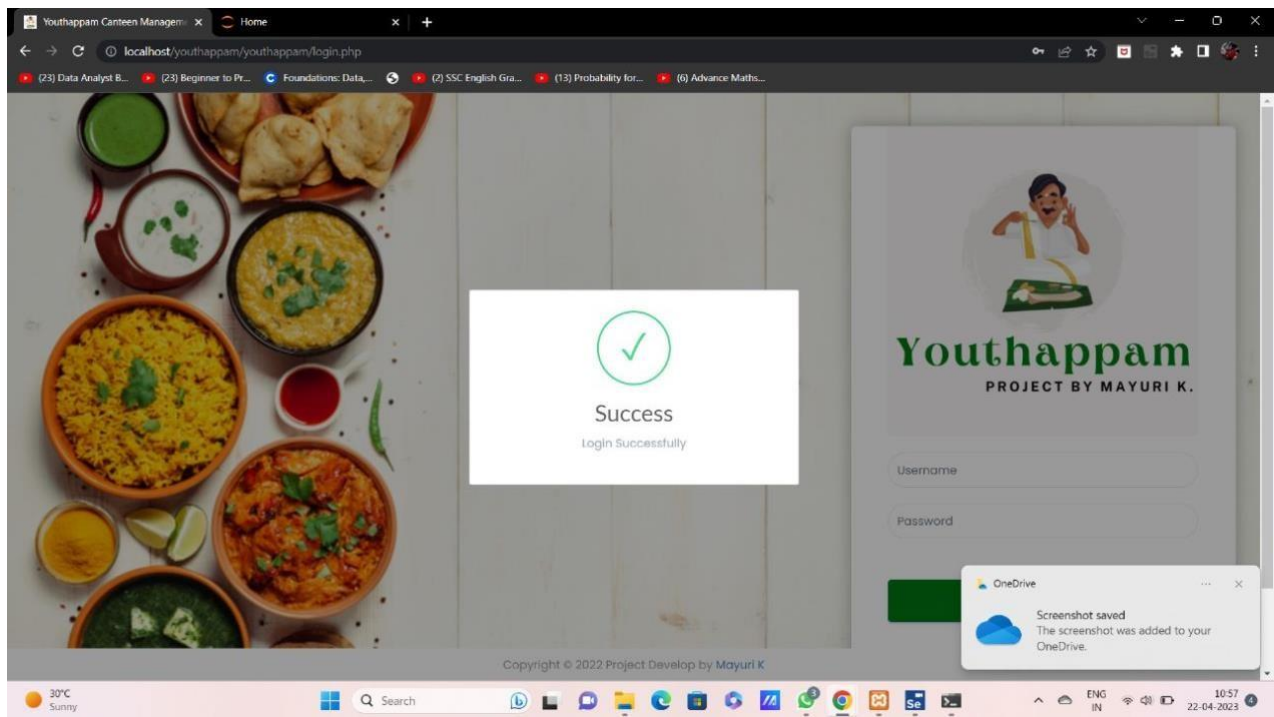
```
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestLogin():
    def setup_method(self, method):
        self.driver = webdriver.Chrome()
        self.vars = { }

    def teardown_method(self, method):
        self.driver.quit()

    def test_login(self):
        # Test name: login
        # Step # | name | target | value
        # 1 | open | http://localhost/youthappam/youthappam/login.php |
        self.driver.get("http://localhost/youthappam/youthappam/login.php")
        # 2 | setWindowSize | 786x816 |
        self.driver.set_window_size(786, 816)
        # 3 | click | css=img |
        self.driver.find_element(By.CSS_SELECTOR, "img").click()
        # 4 | click | id=username |
        self.driver.find_element(By.ID, "username").click()
        # 5 | type | id=username | mayuri.infospace@gmail.com
        self.driver.find_element(By.ID, "username").send_keys("mayuri.infospace@gmail.com")
        # 6 | click | id=password |
        self.driver.find_element(By.ID, "password").click()
        # 7 | type | id=password | rootadmin
        self.driver.find_element(By.ID, "password").send_keys("rootadmin")
        # 8 | sendKeys | id=password | ${KEY_ENTER}
        self.driver.find_element(By.ID, "password").send_keys(Keys.ENTER)
        # 9 | runScript | window.scrollTo(0,44) |
        self.driver.execute_script("window.scrollTo(0,44)")
```





## Requirement Traceability Matrix

| S.no. | Req.ID | Req. Des                 | TC_ID | TC Des.                         | Test Design | Test Exec. |         |          | Defects Status | Req. coverage status |
|-------|--------|--------------------------|-------|---------------------------------|-------------|------------|---------|----------|----------------|----------------------|
|       |        |                          |       |                                 |             | TEnv       | UAT_env | Prod_env |                |                      |
| 1     | test01 | Login to the application | Login | All credentials are correct     | Completed   | passed     | Passed  | No run   | Test OK        | Completed            |
|       |        |                          | TC2   | In email id "@" is missing      | Completed   | Passed     | No run  | No run   | NA             | Partial              |
|       |        |                          | TC3   | Username field is empty         | Completed   | Passed     | No run  | No run   | NA             | Partial              |
|       |        |                          | TC4   | Username & password field blank | Completed   | Passed     | No run  | No run   | NA             | Partial              |
|       |        |                          | TC5   | Password field empty            | Completed   | Passed     | No run  | No run   | NA             | Partial              |
|       |        |                          | TC6   | Password wrong                  | Completed   | passed     | No run  | No run   | NA             | Partial              |

Using Selenium IDE

**SOFTWARE: Youthappam**

**TEST PLAN for Customer Module**

**1.TEST PLAN IDENTIFIER-**

test\_customer

**2.REFERENCES:**

SRS, Use Case Document, Test Strategy, Project Plan, Project Guidelines etc.

**3.INTRODUCTION:**

The objective of this test plan is to check the functionality of the Canteen Management System. Here all the functions such as ordering of meal, payment for the order etc test plan we have made.

**4. TEST ITEM –**

Customer Module

**5. SOFTWARE RISK ISSUES –**

- Regular testing and evaluation of the system should be conducted to ensure its accuracy.
- Data Privacy and Security, use appropriate encryption techniques to protect user data.
- Performance and Scalability, the recommendation system should be designed to handle large volumes of data and user traffic.

**6. FEATURES TO BE TESTED-**

- Customer adding Functionality

**5. FEATURES NOT TO BE TESTED-**

- Database
- Excel tools

## **6.TEST APPROACH-**

The test approach will involve a combination of automated testing technique.

## **7.TEST PASS FAIL CRITERIA-**

Pass/Fail criteria is the predetermined conditions that must be met for a test case to be considered successful (pass) or unsuccessful (fail). These criteria are used to determine the overall quality and readiness of the application for deployment.

- Pass: Customer adding functionality of the web application are working as expected, and there are no critical defects.
- Fail: Customer adding functionality are not working as expected, or there are critical defects that affect the core functionalities of the web application.

## **8.SUSPENSION CRITERIA –**

These criteria are typically based on critical issues or risks, such as data corruption, or system failures, that require immediate attention before testing can continue.

It should suspend if the customer is not getting added in the list of customer

## **9.DELIVERABLES –**

Test reports, defect reports, test scripts and other documentation.

## **10.ENVIRONMENTALREQUIREMENTS-**

Google, Active Internet, Selenium IDE

## **11.SCHEDULE A TSET-**

Testing is schedule on 21 April,2023.



#After login page this will open for customer module

```
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities
```

```
class TestCustomer():
```

```
    def setup_method(self, method):
```

```
        self.driver = webdriver.Chrome()
```

```
        self.vars = { }
```

```
    def teardown_method(self, method):
```

```
        self.driver.quit()
```

```
    def test_customer(self):
```

```
        # Test name: customer
```

```
        # Step # | name | target | value
```

```
        # 1 | open | http://localhost/youthappam/youthappam/login.php |
```

```
        self.driver.get("http://localhost/youthappam/youthappam/login.php")
```

```
        # 2 | setWindowSize | 1552x832 |
```

```
        self.driver.set_window_size(1552, 832)
```

```
        # 3 | click | id=username |
```

```
        self.driver.find_element(By.ID, "username").click()
```

```
        # 4 | type | id=username | mayuri.infospace@gmail.com
```

```
        self.driver.find_element(By.ID, "username").send_keys("mayuri.infospace@gmail.com")
```

```
        # 5 | click | id=password |
```

```
        self.driver.find_element(By.ID, "password").click()
```

```
        # 6 | type | id=password | rootadmin
```

```
        self.driver.find_element(By.ID, "password").send_keys("rootadmin")
```

```
        # 7 | sendKeys | id=password | ${KEY_ENTER}
```

```
        self.driver.find_element(By.ID, "password").send_keys(Keys.ENTER)
```

```
        # 8 | click | linkText=Customer |
```

```
        self.driver.find_element(By.LINK_TEXT, "Customer").click()
```

```
        # 9 | click | linkText=Add Customer |
```

```
        self.driver.find_element(By.LINK_TEXT, "Add Customer").click()
```

```
        # 10 | click | id=name |
```

```
        self.driver.find_element(By.ID, "name").click()
```

```
        # 11 | type | id=name | parth
```

```
        self.driver.find_element(By.ID, "name").send_keys("parth")
```

```
        # 12 | click | id=brandName |
```

```
        self.driver.find_element(By.ID, "brandName").click()
```

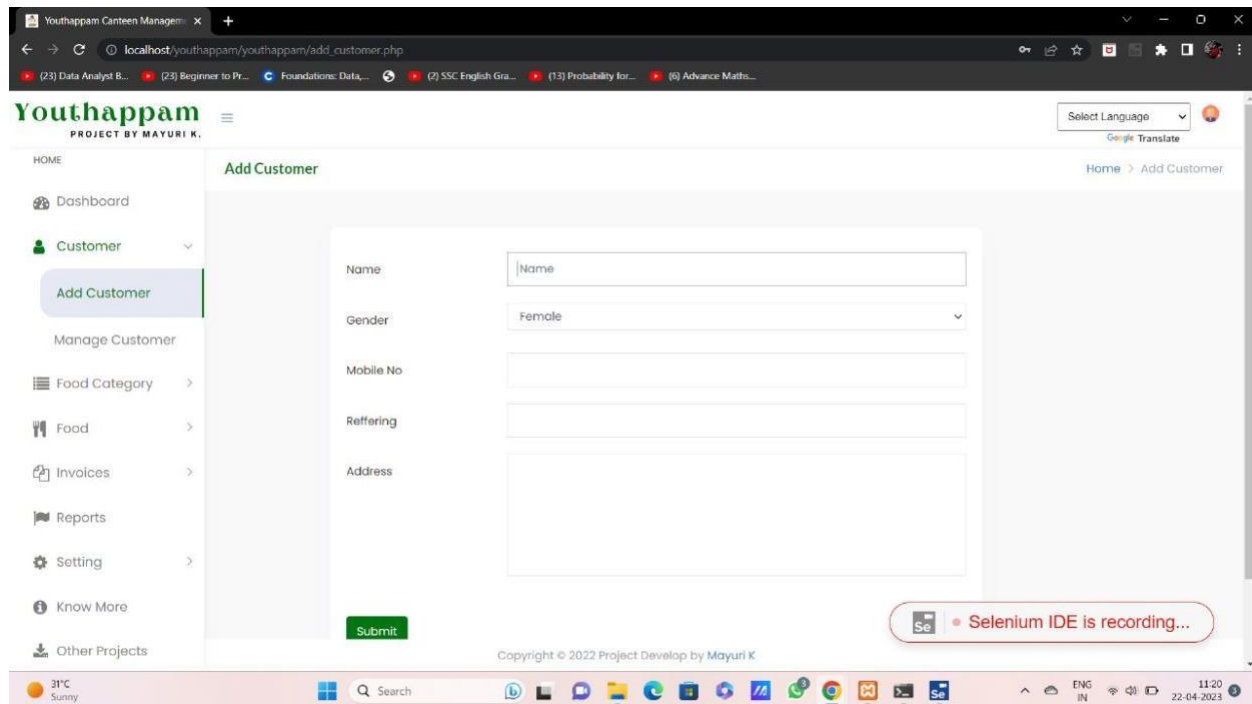
```
        # 13 | select | id=brandName | label=Male
```



```

dropdown = self.driver.find_element(By.ID, "brandName")
dropdown.find_element(By.XPATH, "//*[@option[. = 'Male']]").click()
# 14 | click | name=mob_no |
self.driver.find_element(By.NAME, "mob_no").click()
# 15 | type | name=mob_no | 9898989898
self.driver.find_element(By.NAME, "mob_no").send_keys("9898989898")
# 16 | click | name=reffering |
self.driver.find_element(By.NAME, "reffering").click()
# 17 | type | name=reffering | himanshu
self.driver.find_element(By.NAME, "reffering").send_keys("himanshu")
# 18 | click | name=address |
self.driver.find_element(By.NAME, "address").click()
# 19 | type | name=address | vishrantwadi
self.driver.find_element(By.NAME, "address").send_keys("vishrantwadi")
# 20 | click | id=createProductBtn |
self.driver.find_element(By.ID, "createProductBtn").click()
# 21 | runScript | window.scrollTo(0,16.799999237060547) |
self.driver.execute_script("window.scrollTo(0,16.799999237060547)")
# 22 | click | css=.container-fluid |
self.driver.find_element(By.CSS_SELECTOR, ".container-fluid").click()

```



Youthappam Canteen Manager x +

localhost/youthappam/youthappam/add\_customer.php

(23) Data Analyst R... (23) Beginner to Pr... Foundations: Data... (2) SSC English Gra... (13) Probability for... (6) Advance Maths...

# Youthappam

PROJECT BY MAYURI K.

This Project is developed for Academic Purposes

Select Language

Google Translate

HOME

- Dashboard
- Customer
  - Add Customer
  - Manage Customer
- Food Category
- Food
- Invoices
- Reports
- Setting
- Know More
- Other Projects

## Add Customer

Home > Add Customer

Name: parth

Gender: Male

Mobile No: 9898989898

Referring: himanshu

Address: vishrantwadi

Submit

Copyright © 2022 Project Develop by Mayuri K

Selenium IDE is recording...

31°C Sunny

Search

ENG IN

11:21 22-04-2023

Youthappam Canteen Manager x +

localhost/youthappam/youthappam/customer.php

(23) Data Analyst R... (23) Beginner to Pr... Foundations: Data... (2) SSC English Gra... (13) Probability for... (6) Advance Maths...

# Youthappam

PROJECT BY MAYURI K.

Select Language

Google Translate

















HOME

- Dashboard
- Customer
  - Add Customer
  - Manage Customer
- Food Category
- Food
- Invoices
- Reports
- Setting
- Know More
- Other Projects

## Manage Customer

Show 10 entries

Search:

| # | Client Name        | Gender | Mobile NO   | Referring | Address  | Action  |
|---|--------------------|--------|-------------|-----------|--|---|
| 1 | Rammolli Kallathli | Male   | 8090809080  | Subodh    | Old Kanakapura Rd, Basavanagudi, Bengaluru, Karnataka 560004 |   |
| 2 | Sachin Rajan       | Male   | 8070809805  | Akash     | Plot No.8, Sahajan Colon, Kolkata                            |   |
| 3 | Himanshu           | Male   | 91985463821 | No one    | viman nagar  |   |
| 4 | saksham            | Male   | 8888999999  | aakriti   | dhankowadi,katraj  |   |
| 5 | saksham            | Male   | 8888999999  | aakriti   | dhankowadi,katraj  |   |
| 6 | saksham            | Male   | 8888999999  | aakriti   | dhankowadi,katraj  |   |
| 7 | saksham            | Male   | 8888999999  | aakriti   | dhankowadi,katraj  |   |
| 8 | himanshu           | Male   | 9999900000  | saksham   | tingre nagar,pune  |   |

Showing 1 to 8 of 8 entries

Previous 1 Next

Copyright © 2022 Project Develop by Mayuri K

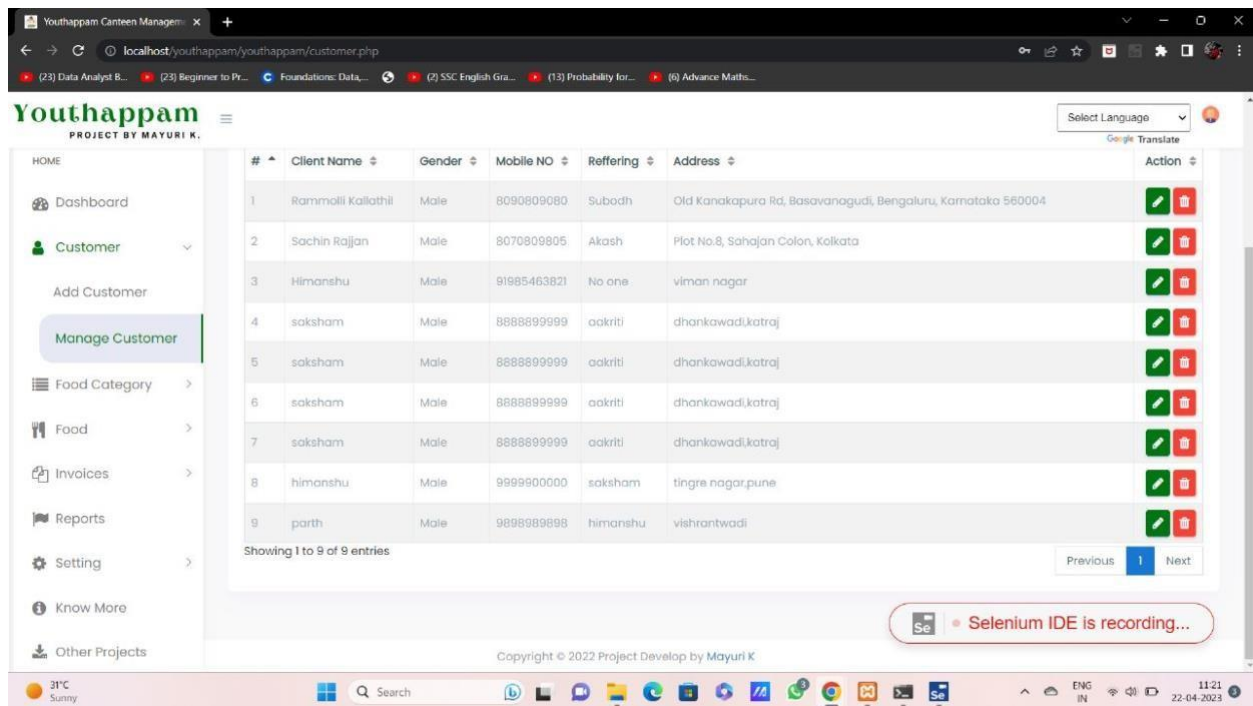
Selenium IDE is recording...

31°C Sunny

Search

ENG IN

11:16 22-04-2023



## Requirement Traceability Matrix

| S.no | Req.ID | Req. Des                    | TC_ID    | TC Des.                | Test Design | Test Exec. |         |          | Defects Status | Req. coverage status |
|------|--------|-----------------------------|----------|------------------------|-------------|------------|---------|----------|----------------|----------------------|
|      |        |                             |          |                        |             | TEnv       | UAT_env | Prod_env |                |                      |
| 1    | test   | Adding customer to the list | customer | All fields are entered | Completed   | Passed     | No run  | No run   | Test OK        | Completed            |
|      |        |                             | TC2      | Any field is empty     | Completed   | Passed     | No run  | No run   | NA             | Not Completed        |

## **CONCLUSION**

In conclusion, the canteen management web application system has proven to be a highly efficient and effective solution for streamlining canteen operations. The system provides seamless online ordering and payment options, real-time inventory tracking, automated menu updates, and user-friendly interfaces for canteen staff. The system has significantly reduced manual tasks, minimized errors, and improved overall operational efficiency. Furthermore, the system has received positive feedback from users for its ease of use and convenience. Overall, the canteen management web application system has been a successful implementation, enhancing the canteen experience for staff.

## **COURSE OUTCOME**

C01: Perceive importance of testing techniques in software quality management and assurance

C03: Apply different testing methodologies used in industries for software testing