



## Tea Factory Management System

ITP2021\_S2\_WE\_G09

Registration No	Name
IT20146238	Jayathunga T.M.
IT20211950	Amarasekara M.G.V.C
IT20005108	Edirisooriya N.W.
IT20237240	Prabashwara U.K.A.L
IT20090562	Nagahawatta S.S.
IT20208462	Gunaratne B.S.
IT20012724	Ananda R.S.R.G.J.N
IT20272654	Rajapaksha R.P.S.V

<b>Group ID : ITP2021_S2_WE_G09</b>	
<b>Project title: Tea Factory management System</b>	
<b><u>Testing Function 1: Test Salary Management</u></b>	
<b>Test case ID : TU001</b>	<b>Test designed by:</b> <b>Reg. No- IT20005108</b> <b>Name- Edirisooriya N.W</b> <b>Reg. No- IT20129712</b> <b>Name- Godakanda P.G.S.</b> <b>Reg. No- IT20254520</b> <b>Name- Bhagya H.D.M.</b>
<b>Test priority (High/Medium/Low):</b>	High
<b>Test Description:</b> <ul style="list-style-type: none"> <li>• To calculate and add a salary for an employee for a specific month, first check if the employee exists.</li> <li>• If the employee exists, then check if a salary has already been calculated for that employee for the given month.</li> <li>• If there are none, then the salary will be calculated and added to the database.</li> </ul>	
<b>Preconditions (if there are any):</b> <ul style="list-style-type: none"> <li>• User has to login to the system</li> </ul>	
<b>Dependencies (if there are any):</b>	
<b>Test steps:</b>	
<ol style="list-style-type: none"> <li>1. Navigate to the calculate salary page.</li> <li>2. Enter the relevant information in the fields properly.</li> <li>3. Click on "Calculate Salary" button.</li> </ol>	

**Pass-condition:**

- An employee with the entered employee Id must exist in the database.
- A salary should not be already calculated for the given employee for the given month and year.

**Test executed Group ID: ITP2021\_S2\_WE\_10**

**Test Executed members details:**

**Reg. No- IT20129712**

**Name- Godakanda P.G.S.**

**Reg. No- IT20254520**

**Name- Bhagya H.D.M.**

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comments
TU001	Employee ID:01 Month : 05 Year : 2021 Attendance:20 Basic Salary:10000 OT hours : 8 Loans : 1000	"Employee Salary Added Successfully"	"Employee Salary Added Successfully"	Pass	All the entered data was valid.
TU001	Employee ID:02 Month : 02 Year : 2021 Attendance: 20 Basic Salary:10000 OT hours : 8 Loans : 1000	"This Employee Salary Already Calculated"	"This Employee Salary Already Calculated"	Pass	Test case passed because employee salary already available in database
TU001	Employee ID: 10 Month : 05 Year : 2021 Attendance:20 Basic Salary:10000 OT hours : 8 Loans : 1000	"Employee Id Invalid"	"Employee Id Invalid"	Pass	Test case passed because that employee id not a registered employee.

<b><u>Testing Function 1: Test Salary Management</u></b>					
<b>Logic in Interfaces</b>	<b>Validation and proper error messages</b>	<b>Connected to the database</b>	<b>Easy to use, User friendliness,</b>	<b>Easy navigation</b>	<b>Function is complete (%)</b>
1. based on good logic	<ol style="list-style-type: none"> <li>(Testing whether the Same Employee ID) <ul style="list-style-type: none"> <li>Error message pop up when "Employee Salary already calculated".</li> </ul> </li> <li>(Testing whether the Unregistered Employee ID) <ul style="list-style-type: none"> <li>Error message pop up when "Employee ID Invalid".</li> </ul> </li> </ol>	1.Database connection is successful	1. All the interfaces are user friendly.	1. User can easily navigate another page.	Overall function completed 100%

**Testing Function 2: Add New Buyer for Auction Stock**

**Test case ID : TU002**

**Test designed by:**

**Reg. No- IT20272654**

**Name- Rajapaksha R.P.S.V**

**Reg. No- IT20275860**

**Name- Nayanakelum S.H.R.M.**

**Reg. No- IT20275860**

**Name- Dabarera W.S.L.C.**

**Test priority (High/Medium/Low):**

High

**Test Description:**

- According to the system, in add buyer function, we must fill all details. (All are required).
- Add buyer function successfully execute with proper frontend validations.
- Then verify all input details and successfully upload data to the database

**Preconditions (if there are any):**

- User has to login to the system

**Dependencies (if there are any):**

**Test steps:**

- Navigate to the buyer details page.
- Then select 'Add Buyer' button.
- Input all the details in given input fields.
- Click 'Insert record' button

**Pass-condition:**

- User must input proper buyer details.

**Test executed Group ID: ITP2021\_S2\_WE\_10**

**Test Executed members details:**

**Reg. No- IT20275860**

**Name- Nayanakelum S.H.R.M.**

**Reg. No- IT20275860**

**Name- Dabarera W.S.L.C.**

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comments
TU002	Company Name: SEP Register No: 0520 Address: Colombo Phone No: 077-236-5589	"Record Saved Successfully...!"	"Record Saved Successfully...!"	Pass	All the entered data was valid.
TU002	Company Name: SEP Register No: 0520568 Address: Colombo Phone No: 07775268	"Value Must Be less the or Equal to 9999"  "Please Match the requested Format"	"Value Must Be less the or Equal to 9999"  "Please Match the requested Format"	Pass	Test case passed because maximum registration number in not valid.
TU002	Company Name: SEP Register No: 052 Address: Colombo Phone No:	"Value Must Be Greater than or equal to 1000"	"Value Must Be Greater than or equal to 1000"	Pass	Test case passed because minimum registration number is not valid.

<b>Testing Function 2: Add new buyer for auction stock</b>					
<b>Logic in Interfaces</b>	<b>Validation and proper error messages</b>	<b>Connected to the database</b>	<b>Easy to use, User friendliness,</b>	<b>Easy navigation</b>	<b>Function is complete (%)</b>
1. based on good logic	3. (Testing whether the value greater than 4 as a register numbers) <ul style="list-style-type: none"> <li>Error message pop up when "Value must be less than or equal to 9999".</li> </ul> 4. (Testing whether the value less than 4 as a register numbers ) <ul style="list-style-type: none"> <li>Error message pop up when "Value Must be greater than or equal to 1000".</li> </ul>	1.Database connection is successful	1. All the interfaces are user friendly.	1. User can easily navigate another page.	Overall function completed 100%

### Testing Function 3: Add items to the cart

<b>Test case ID : TU003</b>	<b>Test designed by:</b> <b>Reg. No- IT20208462</b> <b>Name- Gunaratne B.S</b> <b>Reg. No- IT20222154</b> <b>Name- Jayasinghe W.A.P.K.V.</b> <b>Reg. No- IT19051758</b> <b>Name- Jayasinghe T.S.</b>
<b>Test priority (High/Medium/Low):</b>	Medium
<b>Test Description:</b> <ul style="list-style-type: none"> <li>• When adding an item to the cart, check that it has stock.</li> <li>• If stock is available, items will be added to the cart. Otherwise, it shows there are no stock available.</li> <li>• If there are not enough quantity, it will show the available stocks.</li> </ul>	

#### **Preconditions (if there are any):**

- User has to login to the system

#### **Test steps:**

1. Navigate to the 'New Bill' page
2. Select the product from the list that you want to add to the cart
3. Enter quantity
4. Click 'Add Item' Button

#### **Pass-condition:**

- The selected product needs to have sufficient stock.
- 'Quantity' must be a positive integer.
- 'Quantity' must be less than or equal to the available stocks.



**Test executed Group ID: ITP2021\_S2\_WE\_10**

**Test Executed members details:**

**Reg. No- IT20222154**

**Name- Jayasinghe W.A.P.K.V.**

**Reg. No- IT19051758**

**Name- Jayasinghe T.S.**

Test ID	Test Inputs	Expected Output	Actual Output	Result (Pass/Fail)	Comments
TU00	Product name: BOP 01/100g Quantity: 1	"Item Successfully Added to the list"	"Item Successfully Added to the list"	Pass	All the entered data was valid.
TU00	Product name: BOP 02/200g Quantity: 1000	"Available stock:398"	"Available stock:398"	Pass	Test case passed because it shows available all data in database.
TU003	Product name: OPA 05/500g Quantity: 1	"No stock Available"	"No stock Available"	Pass	Test case passed, because 500g product is not available in database.

<b><u>Testing Function 3: Add items to the cart</u></b>					
<b>Logic in Interfaces</b>	<b>Validation and proper error messages</b>	<b>Connected to the database</b>	<b>Easy to use, User friendliness,</b>	<b>Easy navigation</b>	<b>Function is complete (%)</b>
<p>1. Buttons are easy to access with their placement</p> <p>2. Responsive web design</p>	<p>1. Proper error messages shown with different validations</p> <p>2. Error message shows on the top of the page and can clearly see.</p>	<p>1.Database connection is successful</p>	<p>1. All the interfaces are user friendly.</p> <p>2. When 'Quantity' field keeps empty, it gets the default value as '1'</p>	<p>1. User can easily navigate another page with the side navigation bar.</p>	<p>Overall function is completed 100% with no issue.</p>