

# Travelling and Daily Allowances Management System

Software Test Report



Version <1.0>  
<2/4/2018>

*Submitted by:*

*Nikhil Srivastava (B16CS020)*

*Zaid Khan (B16CS040)*

## **VERSION HISTORY**

<b>Version #</b>	<b>Implemented By</b>	<b>Revision Date</b>	<b>Brief Description</b>
1.0	Nikhil Srivastava and Zaid Khan	02/04/2018	<i>Unit testing and system integration testing of Software for Travelling and Daily allowances.</i>
1.1	Kuldeep Singh and Kavish Gambhir	10/04/2018	Cross Testing for System as whole and independent functions.

---

<sup>1</sup> CS-223 Software Test Report

# Table of Contents

<b>1.0 Introduction</b>	
• Purpose	4
• Application Overview	4
• Testing Scope	4
<b>2.0 Test Plan</b>	<b>5</b>
<b>3.0 Details of unit testing</b>	<b>5</b>
• Registration process	5
• Login process	8
• Authenticate login details	10
• Authenticate change password response	11
• Apply for reimbursement	13
• Send notification for approval	15
• Fetch notification for user	16
<b>4.0 Details of system testing</b>	<b>17</b>

# **1 Introduction**

## **1.1 Purpose**

The purpose of making this test report is to explain the various activities as a part of testing travelling and daily allowance application. Through testing we are going to remove faults and errors to avoid user-end failure of the software.

## **1.2 Application Overview**

Travelling and daily allowance application is a command line software for applying reimbursement for students, professor and staff members for their expenses and they can also check status of their applied approvals. Basically this software deals with the management of the travel allowances borne by the students, professor and staff members. By means of this application, the user would be able to apply for reimbursement of travel allowance, before and after college funded trips. This application will create an interface, where user can register/login using respective credentials and can apply for reimbursement. Further these requests will be sent to admin for further approval.

There are several modules like RegistrationManager, User, Student, Professor, Staff, Admin, AuthenticationManager, Application, NotificationManager etc.

## **1.3 Testing Scope**

- In Scope
- Out Of Scope
- Items not tested

a) In scope- Functional testing is applied for the following modules that are in scope of Testing are as follows-

1. RegistrationManager for registration of user (Professor, Staff and Student)
  2. LoginManager for logging in of User and Admin
  3. Application Class
  4. AuthenticationManager
  5. NotificationManager
- b) Out of scope- Performance related tests, scalability related test and space related tests has not been performed.
- c) Items not tested- User, Admin, Student, Professor, Staff, Notification, Credentials are not tested as they are model class according to our design and they need not be tested as separate classes are there for functions that are performed by the objects. Like RegistrationManager class is responsible for registration of User. Login Manager is responsible for logging in of user and admin.

## **2 Test Plan**

1. Unit testing is performed for all modules described above in which the functionalities are not dependent on other modules.
2. For unit testing all modules described above in scope, Control flow testing and Data flow testing is applied which includes predicate coverage and complete branch coverage.
3. As all the modules are not independent and they have a shared interface and have a interdependency So attempts are carried out to perform appropriate system integration testing by using bottom up approach. All these are done appropriately to minimize errors.

### **3 Details of unit testing**

#### **3.1 Register in RegistrationManager which is defined in SignUp Manually class**

##### **3.1.1 Test Items**

The unit to be tested here is function registration() which is there in RegistrationManager and implemented in SignUpManually. This function registers user i.e. Student, Professor and Staff.

##### **3.1.2 Features to be tested**

- Valid input is being provided i.e input must be in accordance with the parameters.
- Validity of registration process i.e whether a user is pre existed or is a new user.

##### **3.1.3 Item pass/fail criteria**

- If the input is not in accordance with the parameters then it need to ask again for valid inputs from the user which is trying to get registered. Otherwise proceed further.
- No two user can have same username so when this problem arises application need to notify that username is already taken or simply is pre-existed so you need to change username before registering. Or simply go for forgot password if you don't remember your password.
- If the systems passes all those criteria then the test is said to be passed.

##### **3.1.4 Test Cases\_\_**

Test Id	Test Input	Expected Output	Actual Output	Result Pass/ Fail	Comments (if any)
3.1.4.1	choice=1 Implicit Variables: username="Zaid" name="Zaid Raza" password="Hello" security Question="What is your project?" security Answer="Application for Ta/Da settlement" mobile Number="96108120 70" roll Number="B16CS040 " year="2nd year" ugPg Type='u'	Permission Granted! Congratulations! You have been registered Your Username is Zaid	Permission Granted! Congratulations! You have been registered Your Username is Zaid	Pass	As all details are valid and is in accordance with the required input parameters.
3.1.4.3	choice=5	Enter a valid choice	Please enter valid entry	Pass	As options are from 1-3
3.1.4.4	choice=2 Implicit Variables: username="Zaid" name="Zaid Raza" password="Hello" security Question="What is your project?" security Answer="Application for Ta/Da settlement" mobile Number="96108120 70" position="HOD"	Permission Granted! Congratulations! You have been registered Your Username is Zaid	Permission Granted! Congratulations! You have been registered Your Username is Zaid	Pass	-----

<b>3.1.4.5</b>	choice=1 Implicit Variables: username="Zaid"	Username already exists, enter a new username	Username already exists Enter a new username	Pass	As username pre-exists
----------------	--	---	---	------	---------------------------

As for all invalid inputs it will ask for that input again till it gets a valid input and basically when it gets a valid entry it proceed further.

### 3.1.5 Test Result

As for all valid inputs as well as invalid inputs all the expected results are same as actual results. This means that this function passes the first test criteria. And also if the user is pre-existed then it will ask for to register again as username already exists. So second criteria is also passed.

## 3.2 Login functionality in LoginManager which is implemented in SimpleLogger

### 3.2.1 Test Items

The unit to be tested here login() which is there in LoginManager and it is implemented in SimpleLogger as LoginManager is just a interface. This function allows user (Student, Professor and Staff members) and admin to login in the application.

### 3.2.2 Features to be tested

- Valid input must be provided i.e. username must exist to proceed further and password must also match with the password with which account was created.

### 3.2.3 Item pass/fail criteria

- If the credential details i.e. username and password matches with that of the saved data then the user/admin can be logged in. Basically will print successfully logging in.



- After 3 wrong attempts application moves out from login window and moved to homepage.
- If the functionality passes all the criteria then the test is said to be passed.

### 3.2.4 Test Cases

Test Id	Test inputs	Expected Output	Actual Output	Result (pass /fail)	Comments (If any)
3.2.4.1	Implicit inputs: username=""Zaid" password="Hello"	Successfully logging in	Successfully logging in	Pass	User account is already created. And input is matched with the stored
3.2.4.2	Implicit input: username= "zid" password="Hello"	Wrong username or password!, you have 2 attempts remaining	Wrong username or password! You have 2 attempts remaining	Pass	As User account that was created has username as Zaid with the same password
3.2.4.3	Implicit input: username= "zaid" password="Hllo"	Wrong username or password!, you have 2 attempts remaining	Wrong username or password!, you have 2 attempts remaining	Pass	As Password is not matched with the stored input
3.2.4.4	Implicit input: username= "aid" password="Hello"	Max attempts reached, login failed!	Max attempts reached, login failed!	Pass	As number of attempts exceeded the counter i.e. 3 here

### 3.2.5 Test result

As for all valid inputs expected output is matching with the actual output. This means that this functionality passes the test criteria.

### 3.3 Authenticate login Details which is in AuthenticationManager class

#### 3.3.1 Test Items

The unit to be tested here is authenticateLoginDetails(Credentials loginDetails) and it is implemented in AuthenticationManager. This functions authenticate that the password entered matches with the correct password and returns true when authenticated and false for the other condition.

#### 3.3.2 Features to be tested

- Validate whether the user entered correct password. (It is previously checked that the username exists or not, this function only checks for matching password)

#### 3.3.3 Item test pass/fail criteria

- If the password matches with the original password corresponding to the username then this function return true or 1 otherwise it must return 0 or false.
- If the systems passes all those criteria then the test is said to be passed.

#### 3.3.4 Test Cases

Test Id	Test inputs	Expected Output	Actual Output	Result (pass /fail)	Comments (If any)
3.3.4.1	Implicit inputs: username="zaid" password="Hello"	Return true	Return true	Pass	As password for username Zaid is Hello
3.3.4.2	Implicit inputs: username="zaid" password="Hklllo"	Return false	Return false	Pass	As password for username Zaid is Hello and provided input is Hklllo

#### 3.3.5 Test Results

For all the inputs, expected output is same as the actual output. As the test criteria is passed, this means that this functionality works properly.

### **3.4 Authenticate Change Password Response which is implemented in AuthenticationManager**

#### **3.4.1 Test Items**

The unit to be tested here is function `authenticateChangePasswordResponse(User currentUser, string response)` which is implemented in `AuthenticationManager`. This function allows user i.e. Student, Professor and Staff to change password.

#### **3.4.2 Features to be tested**

- Valid input is being provided i.e input must be in accordance with the parameters and if the security answer provided by the user is same as the security answer existing in the database then it must ask for new password.
- Is new password entered matches with that of the confirm new password i.e. New password is confirmed twice so that there will be no need to reduce redundancy.

#### **3.4.3 Item pass/fail criteria**

- If the security answer matches with the the provided response then it will allow user to change password. It will ask for the new password. Otherwise application it will ask for security answer again for the same user. If it matches then it will proceed further.
- If the new password matches with the confirm new password then the password will be changed to new password. Otherwise 3 attempts are provided to enter correct responses
- If the systems passes all those criteria then the test is said to be passed.

#### **3.4.4 Test Cases**

Test Id	Test inputs	Expected Output	Actual Output	Result (pass/fail)	Comments (If any)	Cross Testing
3.4.4.1	Implicit inputs: Security answer = "Application for Ta/Da settlement" newPassword="algorithm" confirmNewPassword="algorithm"	Password successfully updated	Password successfully updated	Pass	As all input matches with that of the saved securityAnswer and new password matches with that of the confirmNewPassword	Verified
3.4.4.2	Implicit inputs: Security answer="answer"	Wrong answer you have 2 attempts left	Wrong answer you have 2 attempts left	Pass	As response is not matching with the security answer	Verified
3.4.4.3	Implicit inputs: response="Application for Ta/Da settlement" newPassword="slgorithm" confirmNewPassword="algorithm"	The two password do not matches	The two password do not matches	Pass	As new password is not same as confirm new password	Verified
3.4.4.4	Implicit inputs: Security answer="Application for Ta/Da settlement"	Wrong attempts reached max allowed value, hence pass remains unchanged !	Wrong attempts reached max allowed value, hence pass remains unchanged !	Pass	As wrong attempts reached maximum allowed value So it will not ask further the security answer	Verified

### 3.4.5 Test Result

For all the inputs, expected output is same as the actual output. As the test criteria is passed, this means that this functionality works properly. And also it is checked that the user already exists or not. That is there in system testing of this functionality integrated with database access layer.

### 3.5 Apply for Reimbursement in Application class

**3.5.1** The unit to be tested here is **applyForReimbursement()** functionality implemented in application class. This function allows user to apply for there travel and daily allowance for a trip.

#### 3.5.2 Features to be tested

- Is reimbursement is applied for a logged in user .
- Validity of the reimbursement application
- If applying for the pre-existing request then whether it should change just the amount spent and status of the reimbursement notification.

#### 3.5.3 Item pass/fail criteria

- On applying for reimbursement admin and user gets a notification which is awaiting for approval.
- If it is a pre- existing request and he applied for post phase Then according to money spent user may have to return money.
- On applying for post reimbursement notification he just need to enter money spent during the journey and rest required details according to the request.
- At the time of applying how much money is remaining.
- If the systems passes all those criteria then the test is said to be passed.

#### 3.5.4 Test Cases

Test Id	Test inputs	Actual Output	Expected Output	Result (pass /fail)	Comments (If any)
---------	-------------	---------------	-----------------	---------------------	-------------------

3.5.4.1	Implicit inputs: Choice =1; Going from="Jodhpur" Going To="Delhi" Purpose="Presentation" Duration= 5days; Phase = "pre " Money Needed= 500; TripMentor="nikhil"	Reimbursement request is applied successfully. Notification is sent to admin for approval and is also stored in userData.	Same as actual output	Pass	As all the data is in accordance with the required one and 500 is less than the money remaining.
3.5.4.2	Implicit inputs: Choice =2; Money spent =1000	Reimbursement request is applied successfully. Notification is sent to admin for approval and is also stored in userData	Same as actual output	Pass	As all the data is in accordance with the required one and 500 is less than the money remaining.
3.5.4.3	Implicit inputs: Choice =1; Going from = "Jodhpur" GoingTo="Delhi" Purpose="Presentation" Duration= 5days; Phase = "pre " moneyNeeded= 50000; TripMentor="nikhil"	Money exceeded the remaining balance so reimbursement request is not registered	Same as actual output	Pass	As money spent crosses the amount left.

### 3.5.5 Test Result

For all the inputs, expected output is same as the actual output. As the test criteria is passed, this means that this functionality works properly.

## 3.6 Send notification to admin for approval and user to notify about the status

**3.6.1** The unit to be tested here is **sendNotification(string userId)** functionality implemented in NotificationManager class which returns void. This function let the application to send notification for approval and notify.

### 3.6.2 Features to be tested

- Whether the application is sending a notification on successful reimbursement request.

### 3.6.3 Item pass/fail criteria

- On applying for reimbursement admin and user gets a notification which is awaiting for approval.
- If it is a pre- existing request then the previous notification is updated with status and phase.
- If the systems passes all those criteria then the test is said to be passed.

### 3.6.4 Test Cases

Test Id	Test inputs	Actual Output	Expected Output	Result (pass /fail)	Comments (If any)
3.6.4.1	Implicit inputs: Choice =1; Going from="Jodhpur" Going To="Delhi" Purpose="Presentation" Duration= 5 days; Phase = "pre " Money Needed= 500; TripMentor="nikhil" Creating a notificationObject from given details and passing it to the	Reimbursement Notification sent to Admin You will be notified of the final decision	Same as actual output	Pass	-----

	function Explicit input :Notification				
<b>3.6.4.2</b>	Implicit inputs: Choice =2; Money spent =1000 Creating a notificationObject from previous details and passing it to the function Explicit input :Notification	Reimbursement Notification sent to Admin You will be notified of the final decision	Same as actual output	Pass	-----

### 3.6.5 Test Result

For all the inputs, expected output is same as the actual output. As the test criteria is passed, this means that this functionality works properly.

## 3.7 Fetch notification for user notifications implemented in DataBaseAccessLayer

**3.7.1** The unit to be tested here is **fetchNotificationsForUser(string userId)** functionality implemented in DataBaseAccessLayer class which returns vector of notifications. This function let the application to fetch list of notifications from the database.

### 3.6.2 Features to be tested

- Whether it is returning list of notifications on passing it with the userId of the user



### 3.6.3 Item pass/fail criteria

- On calling this function this must retrieve the vector of notification. And if no notifications exists then it will return null.
- If the systems passes all those criteria then the test is said to be passed.

### 3.6.4 Test Cases

Test Id	Test inputs	Actual Output	Expected Output	Result (pass /fail)	Comments (If any)
3.6.4.1	Explicit input: username="Zaid"	Vector of notifications is returned	Same as actual output	Pass	To track of applied reimbursement requests

### 3.6.5 Test Result

For all the inputs, expected output is same as the actual output. As the test criteria is passed, this means that this functionality works properly.

## 4 Details of the system testing

System testing is done to check the behaviour of a complete and fully integrated software product coded according to the requirements specification. The main focus of this testing is to check validity in accordance with Business, Functional and End-user requirements.

For this test report Top down approach and end to end testing is done.

### Entry Criteria for system testing

1. Unit and integration testing is performed for all modules.
2. Everything is as per System Requirement Specification Report.
3. User is pre registered before logging in.
4. Provided inputs are in accordance with the parameters.{ As all the modules are tested in unit and integration testing }

## System Testing

Step	Test Step	Intended Result	Actual Result	Pass/Fail	Comments
1.	Login As User-> Apply for reimbursement { Notification is sent to both admin and the user }	Successfully Logged in and Request for reimbursement is registered	Same as intended result	Pass	
2.	Login As User-> Check Notifications { Notifications is retrieved from the database class }	Successfully Logged. List of Notifications is printed	Same as intended result	Pass	
3.	Login As User->Apply for Reimbursement (Previous request)	Successfully Logged in. Request for reimbursement is registered with updated phase and status. And a	Same as intended result	Pass	

		Notification is sent to Both admin and user			
4.	Register as a student	Successfully registered As a student	Same as intended result	Pass	
5.	Register as a Professor	Successfully registered as a Professor	Same as intended result	Pass	
6.	Register as a staff	Successfully registered as Staff Member	Same as intended result	Pass	
7.	Login As Admin->Show all requests->Approve a particular request	Successfully registered as admin. List of all notifications is printed. A Particular request is approved(Status is changed to true)	Same as intended result	Pass	
8.	Forgot Password->Login as a user(Can be student/professor/staff)	Password is updated. Successfully logged in as user.	Same as intended result	Pass	
9.	Login as User->Change details	Logged in successfully. Updated details successfully	Same as intended result	Pass	

Test Id	Test inputs	Actual Output	Expected Output	Result (pass /fail)	Comments (If any)	Post-Cross-Testing Comment
10	Login As User -> Apply for reimbursement -> 3	Error should be thrown as notification number 3 does not exist. Range is {1,2}	Segmentation fault. Program crashes.	Fail	Range exception should be handled.	Resolved
11	Login As User -> Update reimbursement -> 1 ->1	User Dashboard should be appeared	Admin Dashboard is appeared	Fail	Wrongly redirected to dashboard of Admin	Resolved
12	Login As User -> View notifications	No Notifications should be printed	Large number of empty notifications are printed	Fail	It should have thrown error that no requested reimbursement exists.	Resolved