Requirements Based Test Cases for
DRUMRS

Revision 0.1
November 1, 2017

Kah Hin Lai
Ross Faber
Chun F Chak
Amiel Hassan
Group 12

Document Revision History

Rev	Date	Author	Change Description
0.1	**11/1**	Everyone	adding test cases

This document is originating from Neil Bitzenhofer and DataCard Corporation.

Table of Contents

1. Test Requirements

1.1 Objective

The purpose of the Test Requirements section is to list ALL hardware and software test requirements, whether explicitly determined from any relevant documents or implicitly determined from experience and product knowledge. For most projects, the documents referred to may be the Product Definition Document, Software/Hardware Requirements Specification and perhaps the Software/Hardware Design Specification. A Test Case Matrix is provided that simply lists all the test cases by title or description, and includes a method of tracking when the test case was run and whether it passed or not.

1.2 Definitions and Acronyms

List any technical terms or acronyms used in the document, along with their meanings.
Examples for this document:

SRS	Software Requirements Specification
TM	Traceability Matrix
DRUMRS	Dynamic Response to Urgent Maintenance Request Systems

1.3 Traceability Matrix

Requirement \ Test Case	TestCaseID1	TestCaseID2	TestCaseID3	TestCaseID4	TestCaseID5	TestCaseID6	TestCaseID7	TestCaseID8	TestCaseID9
1.1	X								
1.2		X							
1.3			X						
1.4				X					
1.5					X				
1.6						X			
1.7							X		
1.8								X	
1.9									X

Requirement \ Test Case	TestCaseID10	TestCaseID11	TestCaseID12	TestCaseID13	TestCaseID14	TestCaseID15	TestCaseID16	TestCaseID17	TestCaseID18	TestCaseID19	TestCaseID20	TestCaseID21	TestCaseID22	TestCaseID23	TestCaseID24
2.1	X														
2.2		X													
2.3			X												
2.4				X											
3.1					X										
3.2						X									
3.3							X								
3.3.1								X							
3.4									X						
3.5										X					
4.1											X				
5.1												X			
5.2													X		
5.3														X	
5.3.1															X

2. Test Cases

Test Case 1	Emergency Request
Description:	User will be able to communicate an emergency request to maintenance staff
Test Inputs:	Emergency request to DRUMRS system
Expected Results:	Maintenance staff receives emergency maintenance request.
Dependencies:	None
Test Steps:	1. CCM requests emergency service 2. Request will go through DRUMRS system 3. Respective Maintenance staff will receive request.

Test Case 2	Normal Request
Description:	User will be able to communicate a low priority to maintenance staff
Test Inputs:	low priority request to DRUMRS system
Expected Results:	Maintenance staff receives low priority maintenance request.
Dependencies:	None
Test Steps:	1. CCM requests low priority service 2. Request will go through DRUMRS system 3. Respective Maintenance staff will receive request.

Test Case 3	One task per Request
Description:	User will be able to send multiple requests with only one request going through
Test Inputs:	9 request to DRUMRS system within 30 minutes, 9 requests after 30 minutes
Expected Results:	Maintenance staff receives 2 requests. One right away, and then one after 30 minutes.
Dependencies:	None
Test Steps:	1. CCM requests service once. 2. Confirm maintenance staff receives request. 3. CCM sends 8 more requests within half an hour of original request. 4. Confirm maintenance staff does not receive more requests. 5. Repeat steps 1 - 4. 6. Confirm that the interval is working correctly.

Test Case 4	Assign Priority
Description:	Managers will be able to assign a priority to a work order created by a request
Test Inputs:	Low-priority request
Expected Results:	A work order with “routine” priority
Dependencies:	None
Test Steps:	1. CCM will create a low- priority request 2. Manager will receive request 3. Manager will create work order for request 4. Manager will assign priority “routine” 5. Staff will be able to view priority. 6. Repeat steps 1-5 for each priority.

Test Case 5	Provide additional information
Description:	DRUMRS system will ask for additional information after request
Test Inputs:	Low-priority request
Expected Results:	Query for additional information
Dependencies:	None
Test Steps:	1. CCM will create a low- priority request

	2. DRUMRS interface will ask CCM for more information including the problem, contact information, and the type of problem
Test Case 6	Templates for general maintenance
Description:	The DRUMRS system will have templates for common requests.
Test Inputs:	Low-priority request
Expected Results:	A list of available templates will display
Dependencies:	None
Test Steps:	1. CCM will create a low- priority request 2. DRUMRS system will show multiple templates for common requests, including but not limited to toilet paper, markers, paper towels, lights, cleaning.
Test Case 7	Request for Custom Maintenance
Description:	User will be able to not use the template for request and enter custom information for the request.
Test Inputs:	None
Expected Results:	Custom request form is displayed
Dependencies:	None
Test Steps:	1. CCM request for a form of maintenance request. 2. CCM choose the optional custom request form. 3. Custom request form is displayed. 4. CCM enter information of custom request. 5. Verify that system receive accurate request information.
Test Case 8	Call center staff shall be able to create a request in call center.
Description:	Call center staff create request
Test Inputs:	None
Expected Results:	Request is made
Dependencies:	None
Initialization:	Request point is installed in call center
Test Steps:	1. CCM calls in to the call center to make maintenance request. 2 Call center staff record maintenance request information from CCM. 3. Call center staff create a request. 4. Verify that system receive the correct request.
Test Case 9	CCM are able to create request in all rooms that are maintained by custodians and maintenance workers.
Description:	All request points that are installed is publicly accessible
Initialization:	Request points are installed in all selected room
Test Steps:	1. Go to the one of the selected room 2.Verify that requests points in that room are accessible 3. Repeat step 1 and 2 for all the other selected room.
Test Case 10	Sending Notification when work order is created from request.
Description:	Check if the notification is sent
Test Inputs:	None
Expected Results:	a SMS message should be received on the phone
Dependencies:	Test
Initialization:	Phone numbers of the assigned staff and CCM who initiated the request are in the system.
Test Steps:	1. Created a work order from a request and assign it to a maintainer. 2. SMS notifications are created by system and sent to the maintainer and CCM who initiated the request. 3. Verify that notification is sent to the correct phone number.
Exception Path:	If the maintainer has switch off the notification, maintainer will not receive SMS notification in step 2.

Test Case 11	A staff member will be able to turn on/off notifications
Description:	turn off notification
Test Inputs:	None
Expected Results:	a notification is not received by the maintainer.
Dependencies:	None
Test Steps:	1. Maintainer turns of notification in setting 2. Create a system SMS notification manually and send it to the maintainer. 3. Verify that notification is not received by the maintainer.

Test Case 12	Manager will receive a phone call while an emergency request is made.
Description:	emergency notification
Test Inputs:	None
Expected Results:	A phone call for emergency notification
Dependencies:	None
Initialization:	Phone numbers of selected maintainer staff are in the system
Test Steps:	1. Make an emergency request in the system. 2. Automated voice calls are created by system and it calls selected maintainer staff. 3. Verify that system is calling the correct phone number

Test Case 13	Sending Notification to others
Description:	Checks that if a notification is sent
Test Inputs:	None.
Expected Results:	a SMS message should be received on the phone.
Dependencies:	None
Initialization:	A phone number that corresponds to a staff is set in the system.
Test Steps:	1. A notification is requested by System Administrators or Maintenance Staff. 2. Verify that the numbers of people that need to send notifications. 3. Send out notice to specific phone numbers 4. Verify that notification is sent.

Test Case 14	Assigning tasks
Description:	Checks that if a maintainer can be able to assign task to themselves
Test Inputs:	None.
Expected Results:	A task is assigned to a maintainer.
Dependencies:	None
Test Steps:	1. Loaded the time schedule of a maintainer and unassigned task chart. 2. match if any suitable time slots for maintainer to finish the task 3. Verify the assigning task with maintainer. 4. Verify if the System have record of Assigning task to a specific maintainers

Test Case 15	Updating tasks' status
Description:	Checks if a staff member should change the status of a task.
Test Inputs:	The information of a task, such as Name, Date, maintainers who respond to the task's ID and passwords.
Expected Results:	the status of a task is changed by a staff member.
Dependencies:	None
Test Steps:	1. Loaded the task checklists 2. A maintainer logged into the System and Change the status of the task 3. Verify if the System recognize the changes occurred. 4. Go to step 2 to check all the status functionable.

Exception Path:	If the status have been changed for twice in 30 minutes, notify maintenance manager for further assistance.
Test Case 16	Filter tasks
Description:	Check if the system will be able to filter tasks according to status, Date, Maintainers.
Test Inputs:	a list of tasks
Expected Results:	a list of tasks is sorted with a special order such as status.
Dependencies:	None
Test Steps:	1. Loaded the list of the tasks 2. execute the buttons of filtering task according to status 3. verify if the task is match.
Test Case 17	Task Attributes
Description:	Adding new information to an existed task
Test Inputs:	the item of the task
Expected Results:	an updated task is appeared in the task list.
Dependencies:	None
Test Steps:	1. Loaded the list of task. 2. verify that a staff member has accessible to update task information. 3. Check if the update task file is different with the original document. 4. Verify if the update task file is loaded into the database.
Test Case 18	Version Control
Description:	Be able to see who edited the task.
Test Inputs:	the item of the task
Expected Results:	a list of history according to the task
Test Steps:	1. Loaded the list of task. 2. Verify if the task has been updated. 3. check if the person on the history is the same as expected.
Test Case 19	Unmodifiable Work Order ID
Description:	Prevents work orders from being deleted
Test Inputs:	User modification of work order
Expected Results:	Work order and ID persisting
Dependencies:	None
Initialization:	Work order modification begins
Test Steps:	1. DRUMRS user begins modifying work order in system. 2. Verify work order ID cannot be modified
Test Case 20	Manager Assign Work Order
Description:	Allows managers to assign work orders to workers
Test Inputs:	Manager work order modification
Expected Results:	Work order Assigned To status reflects manager's worker choice
Dependencies:	None
Initialization:	Work order modification begins
Test Steps:	1. Manager selects work order 2. Manager begins modifying work order 3. Manager views list of subordinate workers 4. Manager selects worker and submits modification 5. Verify work order Assigned To displays name of chosen worker
Exception:	Subordinate worker is not listed under manager. Will need to be added.
Test Case 21	User Permissions

Description:	Allow and disallow DRUMRS functions for users
Test Inputs:	DRUMRS User Groups: Worker, manager, system admin, IT
Expected Results:	Error when user attempts to perform action not permitted to their user group. Success when permitted
Dependencies:	Permission groups set up within system
Initialization:	DRUMRS function is used
Test Steps:	1. For each user group, attempt all functions associated and not associated with the group (as outlined in other test cases) 2a. For all allowed functions, verify the operation was allowed and completed 2b. For all restricted functions, verify the operation was not carried out, and an error is displayed
Exception:	User lacks a user group or user successfully performs unpermitted operation.

Test Case 22	Manager Report Generation
Description:	Generate comprehensive manager subordinate work order report
Test Inputs:	Manager requests report
Expected Results:	Comprehensive list of all subordinate workers and work orders assigned to them
Dependencies:	Manager has subordinate workers within building
Initialization:	Manager requests work order report
Test Steps:	1. Manager initiates work order report request 2. Verify all workers managed by the manager are present 3. Verify all workers are associated with work orders that have been assigned to them

Test Case 23	System Admin Routine Check
Description:	System admin vital system functionality check
Test Inputs:	System admin requests system check
Expected Results:	Status of system components as outlined in specification 5.3
Dependencies:	None
Initialization:	System admin requests system check
Test Steps:	1. System admin initiates system check 2a. Verify that all system check criteria is checked 2b. Verify all request points are connected through pinging 2c. Verify call center is connected through pinging
Exception:	False positive/negative results from any component

Test Case 24	System Admin Routine Check Notify
Description:	IT notification as a result of system admin check
Test Inputs:	Failed system admin routine check
Expected Results:	SMS message sent to IT support team
Dependencies:	Faulty system component
Initialization:	Make each of the components fail
Test Steps:	1. For each component tested by routine check as outlined in specification 5.3, make fail 1b. Disconnect a request point and run routine check 1c. Disconnect call center and run routine check 2. Verify an SMS message was sent with associated issue message 3. Verify SMS message was received by IT support