Test Results for Rub BBQ RMS

Version 1.0
March 5th 2011

Prepared for:
Jared Leonard, Owner
RUB BBQ
2407 W Lunt Ave.
Chicago, IL, 60645

Prepared by:
Jerod Hodgkin
Mike Schenk
Jessie Floyd
Scott Leonard
Weber State University
Ogden, UT 84408

Summary

Unit Tests: Each developer is responsible for writing unit tests for the modules they are developing. The results of this effort will be described in the "Unit Test Summary Report" section.
There are currently 23 tests with 22 passing. These numbers are expected to increase over the next few weeks.

Component Tests: This is a client-server application and as such we are using a number of message passing interfaces. These are being tested using automation along side our unit tests. The results of these test can be seen in the "Unit Test Summary Report" and the "Test Summary Report" sections.
The BBQRMS client is currently 28% covered by Unit and Component tests.
The BBQRMS services are currently 26% covered by Unit and Component tests.

System Tests: The results of our manual system testing efforts can be seen in the "Issue Summary Report" section. We are using Jira to track issues found during system testing.
The number of issues reported spiked in the beginning of February and is expected to climb over the next few weeks.
The number of open issues is expected to descend rapidly over the next several weeks.
Issues opened: 57
Issues resolved: 24

Summary: For a complete list of which features have been tested, are currently being test, or are not ready to test, see the "Feature Test Results" section.
Total components = 44
Testing Completed = 8
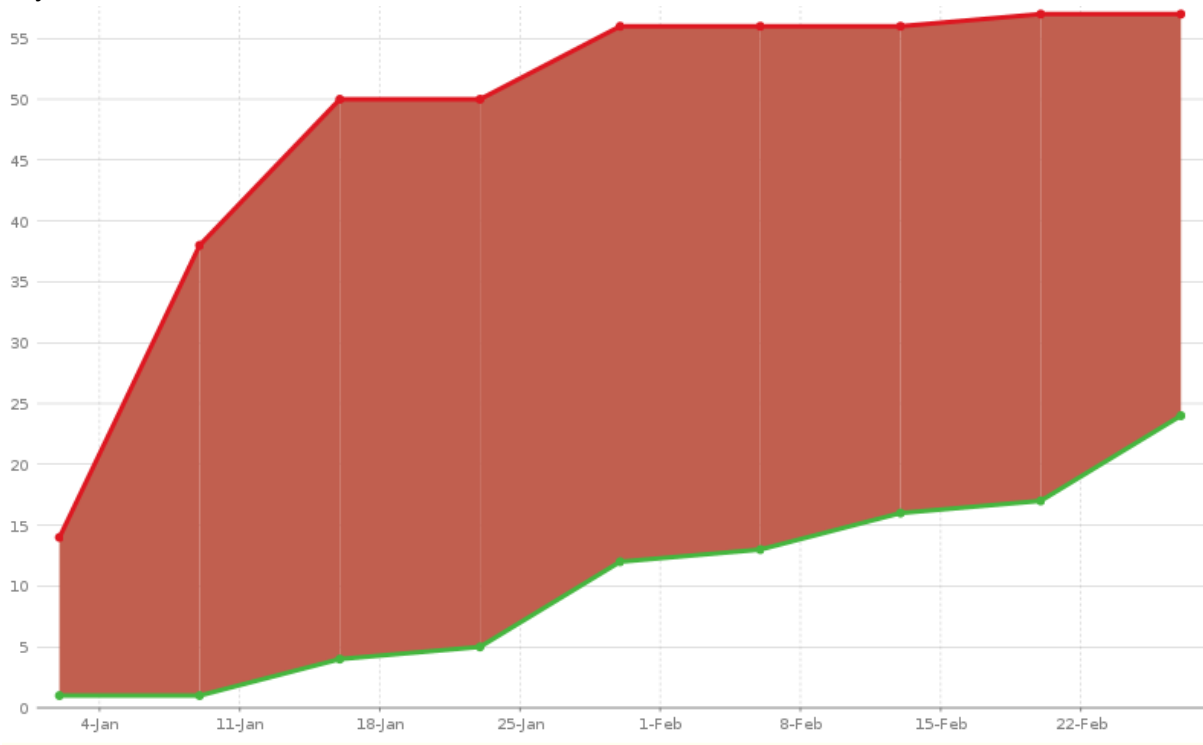Not ready for Testing = 13
Currently being tested = 23

Issue Summary Report

Report: <u>Created vs. Resolved Issues Report</u>

Project: <u>BBQRMS</u>

Chart

This chart shows the number of issues **created** vs. the number of issues **resolved** in the last **60** days.

Data Table

Period	Created	Resolved
Week 2, 2011	14	1
Week 3, 2011	24	0
Week 4, 2011	12	3
Week 5, 2011	0	1
Week 6, 2011	6	7
Week 7, 2011	0	1
Week 8, 2011	0	3
Week 9, 2011	1	1
Week 10, 2011	0	7

Unit test Summary Report

	Result	Test Name	Project
☐	Passed	PaymentTest	BBQRMS.Tests
☐	Passed	OrderTest	BBQRMS.Tests
☐	Passed	MenusTest	BBQRMS.Tests
☐	Passed	CookTest	BBQRMS.Tests
☐	Passed	CashierTest	BBQRMS.Tests
☐	Passed	CancelTest	BBQRMS.Tests
☐	Passed	AddToOrderTest	BBQRMS.Tests
☐	Passed	AddPaymentTest	BBQRMS.Tests
☐	Passed	PlaceOrderTest	BBQRMS.Tests
☐	Passed	NewPaymentTest	BBQRMS.Tests
☐	Passed	CustomerOrderScreenViewModelConstructorTest	BBQRMS.Tests
☐	Passed	WhenScreenIsOpen_ThenDailySalesReportCanBeRunAndDefaultsToLastWeek	BBQRMS.Tests
☐	Passed	WhenDailySalesReportIsRun_ThenNoExceptionsAreThrown	BBQRMS.Tests
☐	Passed	WhenUserClocksOutAndConfirms_ThenClockOutTimeIsRecordedAndUserIsLoggedOut	BBQRMS.Tests
☐	Passed	WhenLogoutIsClicked_ThenScreenRevertsToLoginScreenAndCurrentUserIsCleared	BBQRMS.Tests
☐	Passed	WhenApplicationStarts_ThenLoginViewModelIsCurrentFullScreenContent	BBQRMS.Tests
☐	Passed	WhenUserLogsInWithValidPIN_ThenUserLoggedInMessageIsPublished	BBQRMS.Tests
☐	Passed	WhenUserLogsInFirstTime_ThenEmployeeTimeClockIsCreated	BBQRMS.Tests
☐	Passed	WhenUserLogsInMultipleTimes_ThenEmployeeTimeClockRecordIsReused	BBQRMS.Tests
☑	Inconclusive	WhenUserForgetsToClockOut_ThenNewTimeClockRecordIsCreatedOnNextLogin	BBQRMS.Tests
☐	Passed	WhenUserLoggedInMessageIsReceived_ThenFullScreenContentSwitchesToPostLoginViewModel	BBQRMS.Tests
☐	Passed	WhenUserEntersWrongPINFormat_ThenLoginViewModelIsStillCurrentAndErrorIsDisplayed	BBQRMS.Tests
☐	Passed	WhenUserEntersUnrecognizedPIN_ThenLoginViewModelIsStillCurrentAndErrorIsDisplayed	BBQRMS.Tests

This is a list of the current set of Unit tests for the BBQRMS. This list is expected to grow rapidly over the next few weeks as the application starts to stabilize. We intentionally did not to adhere to a test driven development practices due to time constraints and other concerns. Now that we have established some solid application patterns we expect to see greater test coverage as we approach a beta release.

Test Summary Report

The following is a snapshot of the code covered by the current suite of unit tests.

Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
HodgkinJT@JEROD-E6500 2011-03-05 22:28:32	3689	73.15 %	1354	26.85 %
BBQRMS.WCFServices.dll	1166	77.48 %	339	22.52 %
{} BBQRMS.WCFServices	1166	77.48 %	339	22.52 %
BBQRMSSolution.exe	2523	71.31 %	1015	28.69 %
{} BBQRMSSolution	16	100.00 %	0	0.00 %
{} BBQRMSSolution.BusinessLogic	13	43.33 %	17	56.67 %
{} BBQRMSSolution.Messages	8	40.00 %	12	60.00 %
{} BBQRMSSolution.Models	40	86.96 %	6	13.04 %
{} BBQRMSSolution.Properties	5	100.00 %	0	0.00 %
{} BBQRMSSolution.ServerProxy	816	69.51 %	358	30.49 %
{} BBQRMSSolution.ViewModels	1208	71.65 %	478	28.35 %
{} BBQRMSSolution.ViewModels.Reports	0	0.00 %	144	100.00 %
{} BBQRMSSolution.Views	274	100.00 %	0	0.00 %
{} BBQRMSSolution.Views.ViewParts	86	100.00 %	0	0.00 %
{} RUBPrototypes.SampleData	57	100.00 %	0	0.00 %

Feature Test Results

KEY
T = Tested.
N = Not Ready for testing.
C = Currently being tested.

TOTALS T = 8, N = 13, C = 23

Customer Orders

C - Enter an order.
C - Save an order.
C - Recall a saved order for modification.
C - Take partial payment on a order and save for later payment.
C - Take payments on an order.
N - Read payment card information from hardware MSR.
N - Refund order on cancelled order with partial payment.
N - Discount an order by percentage.
N - Discount order by dollar amount.
C - Send an order to the cook for completion.
N - Cook's screen ability to mark orders completed.
C - Credit card integration sends correctly formatted information to the API.
C - System receives payment authorization from credit card processor.
C - Cancel a customer order.

Inventory

C - Enter a new inventory supplier.
C - Change existing supplier details.
C - Delete supplier.
C - Enter new inventory items.
C - Receive inventory (quantity updates properly.)
C - Record used inventory (quantity updates properly.)
C - Delete inventory item.
C - Quantities update properly on inventory reconciliation.

Employee Administration

T - Create a new system user (employee.)
T - Change employee details from admin screen.
C - Change employee PIN from admin screen.
C - Employee can change their own pin.
N - Delete an employee.
T - Add a role to an employee.
T - Remove a role from an employee.

T - Add and remove privileges from a role.

T - Create a new role.

C - Delete a role.

Menu Management

N - Add new menu items to the database.

T - Build a menu from menu items already in the database.

N - Delete menus from the database.

N - Delete menu items from the database.

Reporting

C - Reports calculate correctly.

C - Reports generate accurate information.

N - Inactivated data will show up in reports where required.

Security

T - Users can log in to the system.

C - Users see interface items that appropriately relate to their roles.

N - Manager approval is required before applying discount.

N - Deleted (inactivated) users cannot log in.

Performance

N - Taking customer orders must occur within the time frames specified in the respective use case specifications.