TEST PLAN INSTRUCTIONS DOCUMENT

8/23/2013

Graduate Capstone

RECEIPT REWARDS

Table of Contents

1 Introduction 3
1.1 Intended Audience 3
1.2 References 3
1.3 Revision History 3
2 Test Methodology 4
2.1 Elements of Testing 4
2.2 Items To Be Tested 4
2.3 Items Not To Be Tested 4
3 Types of Testing 5
3.1 Functional Testing 5
3.2 Data Testing 6
3.3 User Testing (User Acceptance Testing) 7
3.4 Non-Functional Testing 7
3.5 Performance Testing 8
3.6 Integration Testing 9
3.6.1 Bottom Up Approach 9
3.6.2 Integration Plan 10
3.7 Unit Testing 11
Appendix 12
Appendix A: Functional Test Log 13
Appendix B: User Acceptance Testing Results ..... **Error! Bookmark not defined.**
Appendix C: Performance Test Results ..... **Error! Bookmark not defined.**
Appendix D: Integration Test Result Template 14

1 Introduction

The purpose of this document is to create a clear description about the approach to testing that will be done on the system. Testing is a crucial part to the development process, and must be understood and followed to ensure quality of the code and validation of the requirements.

1.1 Intended Audience

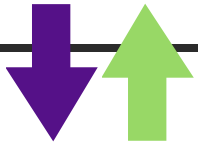
This document in intended for users of a moderate technical background. It is ideal for users to be very familiar with the system. In general, this document should be used by testers as a reference of all the testing involved in the system.

1.2 References

- http://www.dir.texas.gov/SiteCollectionDocuments/IT%20Leadership/Framework/Framework%20Extensions/SDLC/SDLC_testPlan_instructions.pdf
- http://www.engr.sjsu.edu/gaojerry/course/287/TestPlan.doc
- http://www.jiludwig.com/templates/ATTemplate.doc
- http://ltodi.est.ips.pt/es/index_files/pdf/TestPlanOutline.doc
- http://www.smithmusic123.com/application/views/download/download_file/Testing.pdf
- http://sucs.org/~tobeon/testing.pdf

1.3 Revision History

Name	Date	Reason For Change	Version
Andy Bottom	06/24/2013	Created the formatting of the document and added references	0.1
Andy Bottom	08/16/2013	Updated Testing Descriptions and added Integration Test Ret	0.2
Andy Bottom	08/22/2013	Finished Test Plan document.	1.0



2 Test Methodology

@TODO: Explain the coding standards.

2.1 Elements of Testing

@TODO

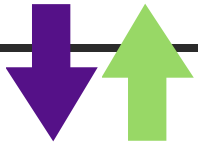
2.2 Items to Be Tested

- Integration Testing

2.3 Items Not To Be Tested

Due to time and scope constraints, formal Data Testing will not be done during the Capstone. Informal checks of the database may be done to ensure that data quality is at a passable level as to not cause critical bugs.

For more information about items that will not be developed, please refer to the **Vision and Scope Document**.



3 Types of Testing

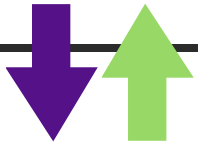
3.1 Functional Testing

Functional testing will be used to test all the functional requirements of the system. This type of test will ensure that the product is meeting the criteria of what the system was designed to do.

All the functional requirements can be found in the **Software Requirement Specification** document. Along with a list of the functional requirements, there are FIT criteria of which the tests are supposed to meet in order to pass.

If a functional requirement does not pass, and development related to those requirements has been completed, then a bug must be created to resolve the problem. Also, at this point, it may ultimately be decided that the requirement had changed from when it was last updated, and a new change to the requirement may be needed to be made to allow for the change. This change may then cause the functional pass to now pass and a bug may not need to be filed.

A log of the functional testing must be made in order to identify when in the testing process it was passed. Once a functional requirement was verified and passed functional testing, further testing of that requirement may not be needed to be done anymore. In this case, a passed requirement will be only tested in a larger smoke test of the entire system. The log can be found in this document under Appendix A.

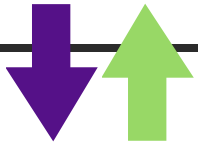


3.2 Data Testing

Data testing are tests that are done on the data in the database. The purpose of Data Testing is to ensure data quality of the system.

To determine the success criteria of a system, the data quality requirements will be found as a non-functional requirement of the database and system.

The types of quality standards that can be done with data testing may include but not limited to, relations in the database between entries. Ensuring no data is or become orphaned in the database; that all primary ids are unique and not duplicated. Other types of data testing may involve proper formatting of certain types of data. Such as all Date Fields will be formatted YYYY-MM-DD, or something similar.



3.3 User Testing (User Acceptance Testing)

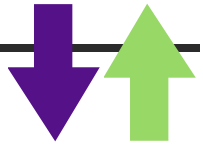
User Acceptance Testing will be a large aspect of the testing for this project. User Acceptance testing will be done to determine whether or not the product or prototype is on the correct path and meets the needs of individual users of the system. The importance of this type of testing cannot be understated. Because users will ultimately be the ones to use the product, it is vital that the application pass the tests from User Testing.

In addition to testing Non-Functional Requirements, you can also obtain very valuable information from User Acceptance Tests. Some of the data may determine if the product is meting the user needs, if the interface is helpful or inhibiting, if the overall experience of the application.

Whenever User Acceptance Testing is performed, the results will be logged so that the data can be used and acted upon the findings. The results of the User Acceptance Testing can be found in this document under Appendix B.

3.4 Non-Functional Testing

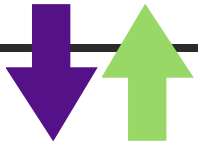
The non-functional tests in this project are spread out between the other forms of testing. Performance sorts of tests will be done in the Performance Testing. Other generic non-functional requirements will be tested in the Functional Tests. See section 3.1



3.5 Performance Testing

Performance Testing is done in respects to time and duration. The goal of performance testing is to test efficiency of the system. Performance Tests typically will come from Non-Functional Requirements. The success of the requirements will come from the FIT Criteria.

As mentioned before, performance testing can test thing such as the amount of wait time a call to the web server takes, or the size of the json file that is returned from the web services to the phone. The performance tests goal is to make the application faster for the user not to have to wait, and that the size of data is efficient to prevent sluggishness, memory leaks or database timeouts.



3.6 Integration Testing

@TODO: kind of look this over and reword awkward parts.

Integration testing is performed to ensure parts, modules or code chunks were together with other code sets, when they are being integrated together, hence the name. The importance of these tests is to provide a set of tests during integration as a check that pieces are fitting together and unexpected bugs that occur are caught immediately.

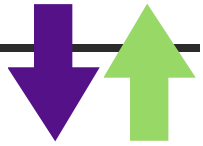
Due to the nature of my project, having 4 large standalone modules, the integration testing will be the most vital of the tests for this system. By doing integration tests, it will be the most efficient at identifying bugs during this stage.

Also, integration tests are great in doing black box testing which allows for very broad tests to be done to catch other inconsistences and get these fix.

With the iterative type of software development, the integration test is also perfect fit. At the end of iterations, an integration test will be performed to ensure all changes work well with the existing product and functional prototype is resulted.

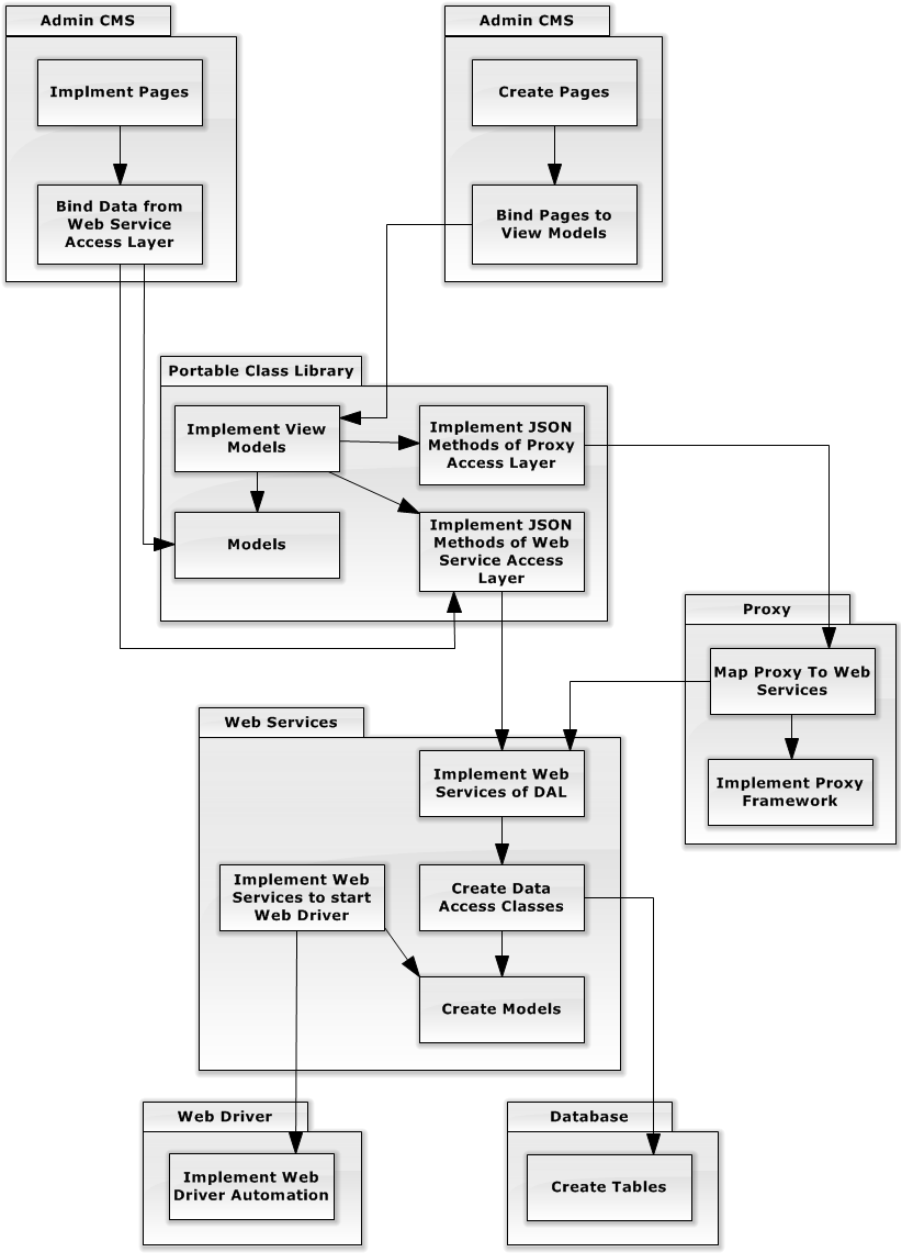
3.6.1 Bottom Up Approach

As specified in the **Iteration Document**, I will be developing with a Bottom-Up Approach. As with all things, everything has its benefits and disadvantages. With the bottom-up approach, the advantage is that the backend functionality will be tested first and built upon one another. This is advantageous because testing on database, web services and admin can all be done early. This will ensure the quality of the back-end functionality. However, the disadvantage is that the phone application will be implemented and integrated last. This is problematic because functioning prototype won't be created to the very end, thus testing and quality of the phone app will be rushed thus opening the possibility of more bugs.



3.6.2 Integration Plan

The following is a diagram representing all the high level modules and sub-modules of the project. The lines represent the dependencies of each component.

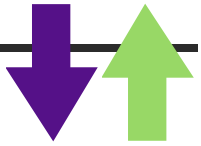


3.7 Unit Testing

Unit testing is essentially the use of automation to perform testing. Often times, a test case(s) should be created with a piece of functionality is created. This test case will be added to the test suite for later use. Often times the Unit test can be ran to ensure that no bugs are created. This is the advantage to catch a wide variety of bugs so that they don't occur and are prevented.

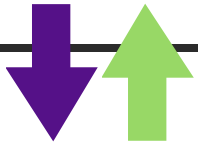
The disadvantage of unit testing is the development time will be needed to create the test cases, but this pays off in the long run as they provide solid testing down the line and no additional resource is needed to test since it is automation.

Even though the benefit of Unit Testing is evident, due to the limited amount of developer resources and time, Unit Testing will not be done for the scope of this project as valuable time will be needed to develop the entire system.



Appendix

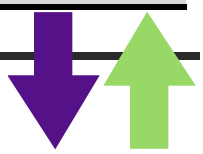
@TODO: Make a cooler Title or something. Maybe a mini-table of contents or something.



Appendix A: Functional Test Log Example

The functional testing will be referencing the Functional and Non-Functional Requirements (that can be found in the Software Requirement Specification.) All success criteria will ensure that it passes the FIT criteria.

Req ID	Test Case Description	Date Tested	Pass / Fail	Test Result Description
REQ-01	Provide a brief description of the functionality the case will test	mm/dd/yy	Pass	
REQ-02	Add more lines as needed and remove blue text prior to use			
REQ-03				
REQ-04				
REQ-05				
REQ-06				
REQ-07				
REQ-08				
REQ-09				
REQ-10				
REQ-11				
REQ-12				
REQ-13				
REQ-14				
REQ-15				

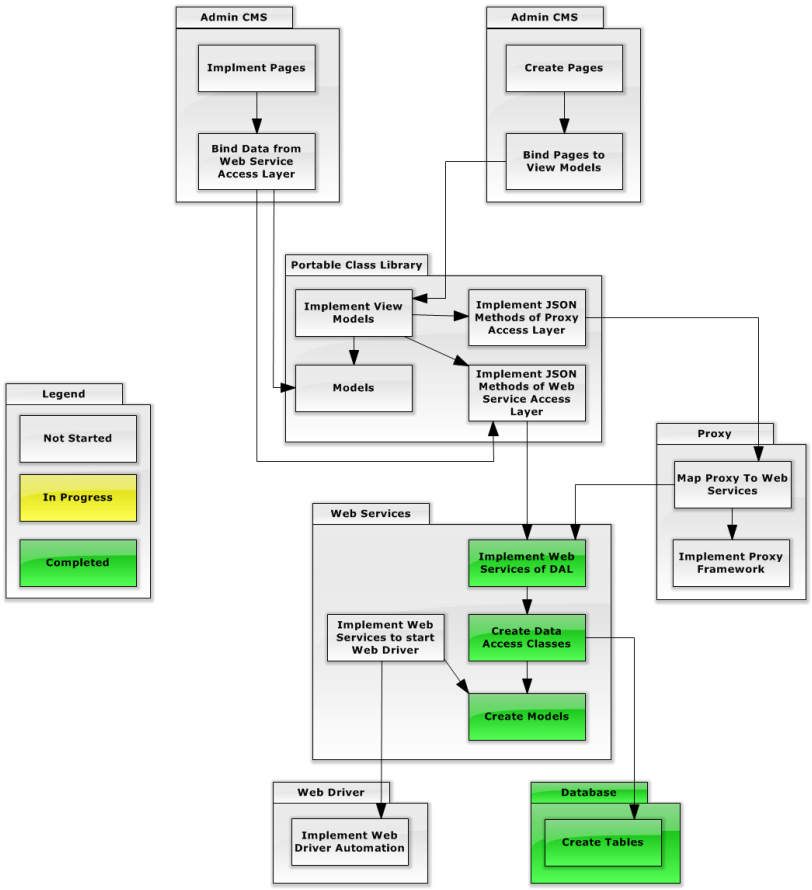


Appendix B: Integration Test Result Template

The following is an example of the form used to log the results of the Integration Tests.

Integration ID		Test Author	
Test Name		Date Tested	
Modules Involved			

Test Summary



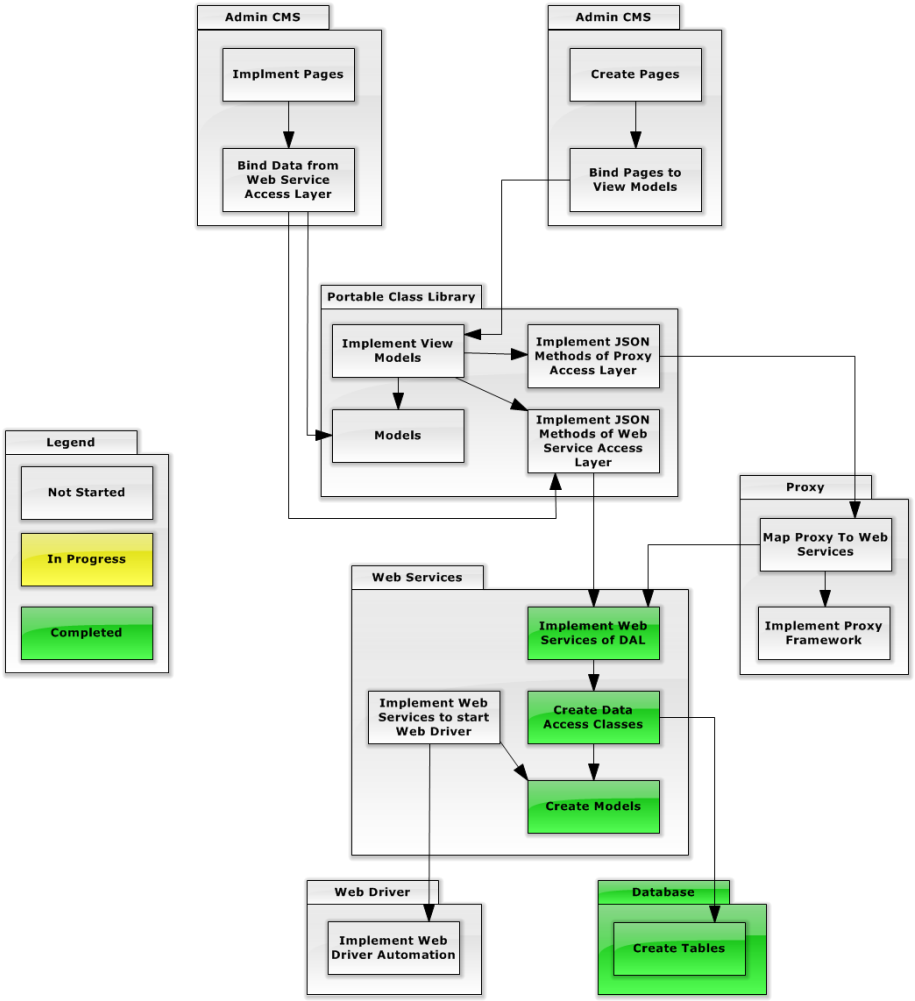
Appendix C: Integration Test Results

INT-01

Integration ID	INT-01	**Test Author**	Andy Bottom
Test Name	DAL Test	**Date Tested**	@TODO
Modules Involved	Web Services; Data Access Layer; Database;		

Test Summary

@TODO

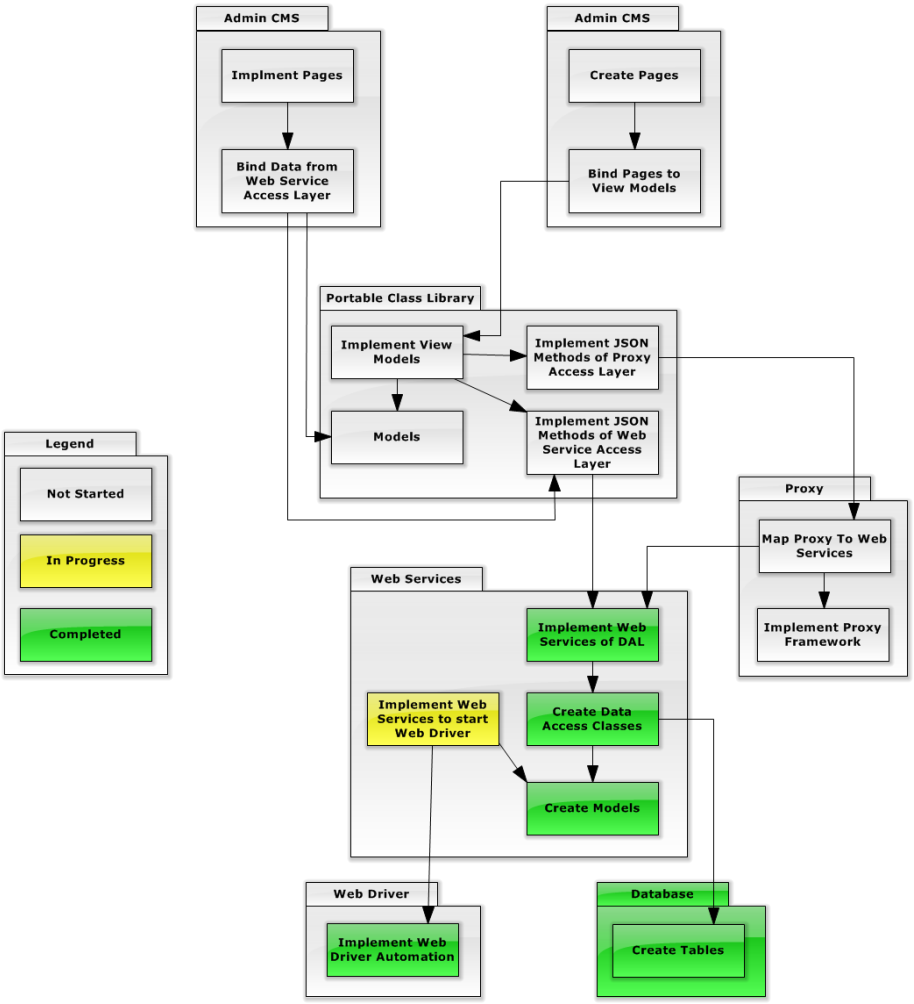


INT-02

Integration ID	INT-02	**Test Author**	Andy Bottom
Test Name	Web Driver Test	**Date Tested**	@TODO
Modules Involved	Web Services; Web Service Automation Layer;		

Test Summary

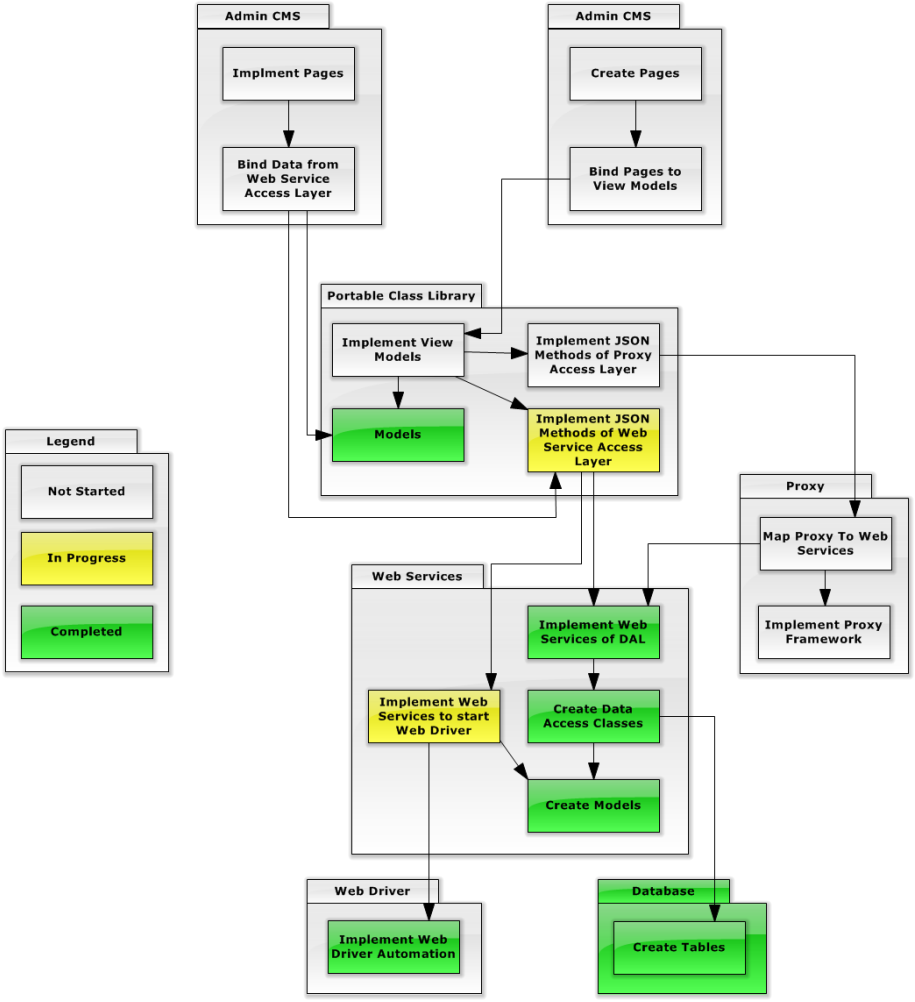
@TODO



INT-03

Integration ID	INT-03	**Test Author**	Andy Bottom
Test Name	Automation Test	**Date Tested**	@TODO
Modules Involved	Web Service Automation Access Layer; PCL;		

Test Summary
@TODO

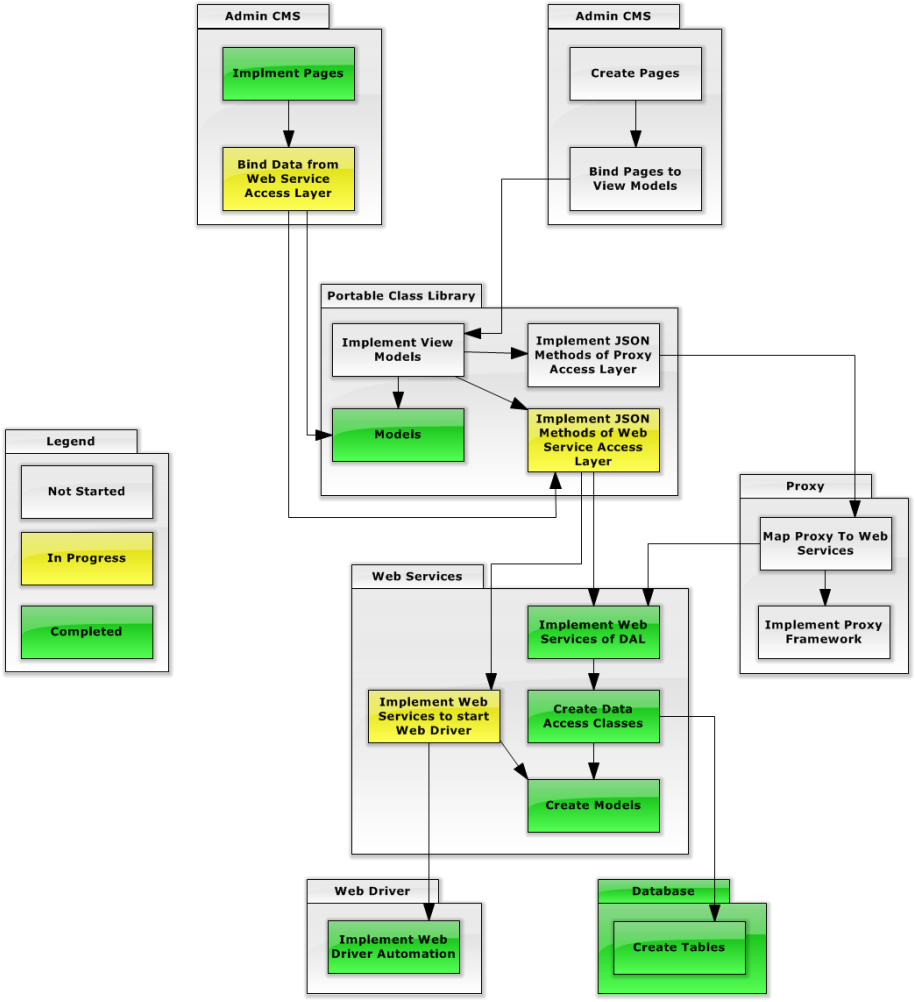


INT-04

Integration ID	INT-04	**Test Author**	Andy Bottom
Test Name	CMS Binding Test	**Date Tested**	@TODO
Modules Involved	Web Admin CMS; PCL		

Test Summary

@TODO



INT-05

Integration ID	INT-05	Test Author	Andy Bottom
Test Name	Complete Web Service Test	Date Tested	@TODO
Modules Involved	Admin CMS; PCL; Web Services; Web Driver;		

Test Summary

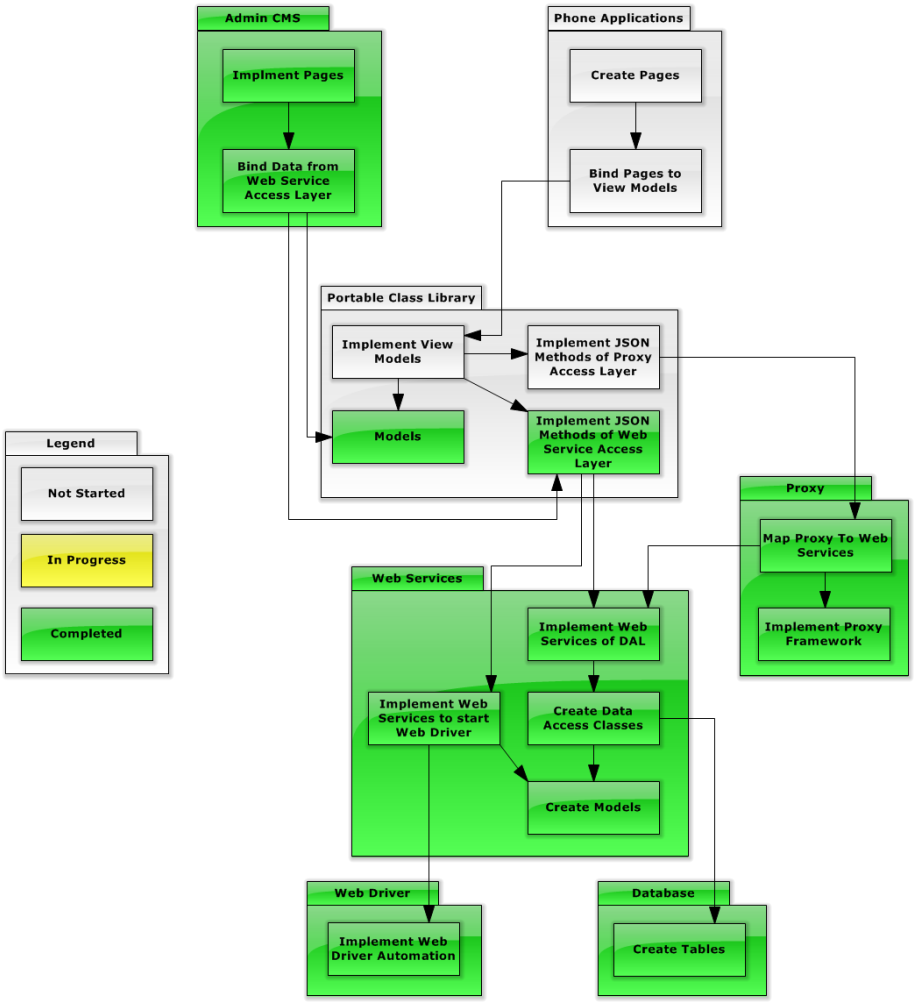
@TODO

INT-06

Integration ID	INT-06	**Test Author**	Andy Bottom
Test Name	Proxy Test	**Date Tested**	@TODO
Modules Involved	Proxy; Web Services;		

Test Summary

@TODO

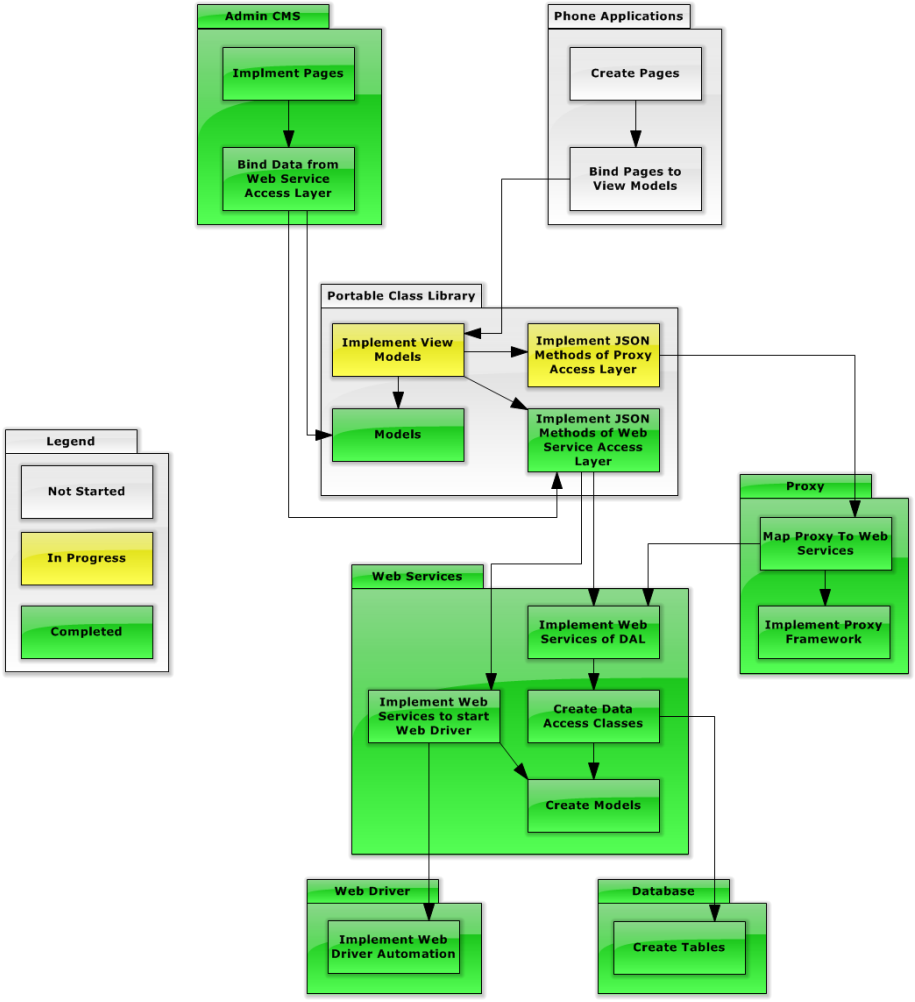


INT-07

Integration ID	INT-07	**Test Author**	Andy Bottom
Test Name	Initial View-Model Test	**Date Tested**	@TODO
Modules Involved	PCL;		

Test Summary

@TODO

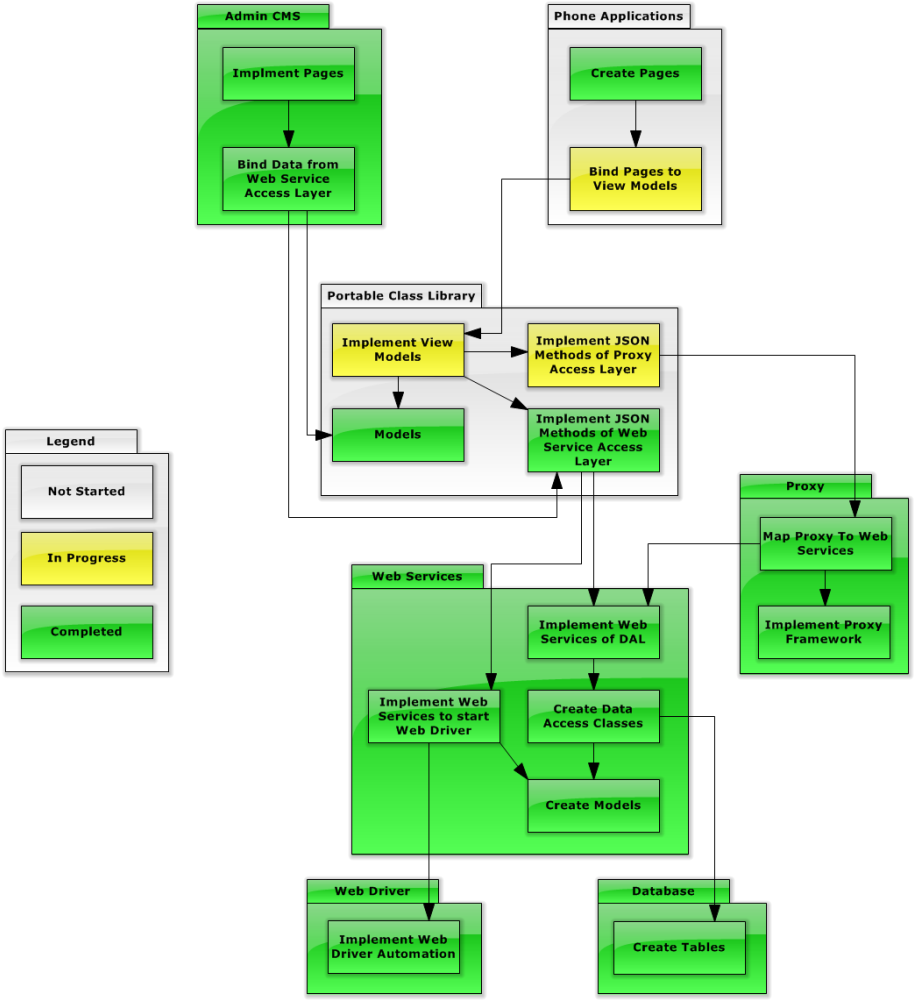


INT-08

Integration ID	INT-08	**Test Author**	Andy Bottom
Test Name	Phone Binding Test	**Date Tested**	@TODO
Modules Involved	Phone Apps; PCL;		

Test Summary

@TODO



INT-09

Integration ID	INT-09	Test Author	Andy Bottom
Test Name	Complete Integration	Date Tested	@TODO
Modules Involved	Phone Application; PCL; Proxy; Web Services;		

Test Summary

@TODO