

***The Electronic Stamp
Mail Server and Client
Project Plan
Part 6b: Software Test Results***

By

SEGY
Software Incorporation

STR Final
November 30, 2003

This Software Test Results Summary was prepared and provided as a deliverable for Florida State University, Software Engineering Class, CEN 5035, for Fall Term 2003. This document is based on the types of tests in the Software Testing Plan based on the implemented prototype.

Gabrielle Reed,
Project Manager

E Shen,
Repository Expert

Stanislav Ustymenko,
Project Leader

Yunwei Wang,
Technical Leader

Change History

Revision	Date	Author	Section/Pages Affected	Remarks
Version1	November 30, 2003	R. G. Reed E Shen Stanislav Ustymenko Yunwei Wang	All	Initial

Preface

This document is prepared as part of the requirements for the Project Plan assignment of the Software Engineering class. The information contained within is based on information provided in the text books, the Software Engineering Class and the SWENG website. We would like to thank the PALS Learning Systems Institute at the College of Education for the use of equipment, meeting space and software provided under National Science Foundation Grant # IIS-0218692.

Table of Contents

Preface.....	4
Table of Contents.....	5
1. Introduction.....	6
2. Prototype Test Results	6
2.1. Testing Results at a Glance.....	7
3. Conclusion	11
Appendix A. Ad-Hoc Testing of Modules.....	12
Appendix B Unit Test Methods	13
Appendix B.1. Unit Test for Estamp	13
Appendix B.2. Unit Test for Estamp Manager	15
Appendix C. Unit Test Driver for Estamp Vendor System	18
Test Description.....	18
Java Code for Unit Test	Error! Bookmark not defined.
Appendix D.1. GUI Test of the Configuration / Estamp Book	18
Appendix D.2. GUI Test of the Compose / Send	18
Appendix D.3. GUI Test of the Receive/Filter Panel	19
Appendix E. Integration Test of Estamp Purchase of client and Estamp Vendor	20
Appendix F. Security Tests for Estamp Integrity	21

1. Introduction

The purpose of this document is to provide the results of the Software Testing Plan (STP) for the prototype. The types of tests for each requirement are outlined in Table 1 of the STP. The tests were performed by comparing performance of each feature to the client requirements outlined in the “Whitepaper”.

2. Prototype Test Results

A comprehensive testing plan of the prototype was outlined in the Software Testing Plan (STP). The purpose of this was to provide a test plan that would demonstrate compliance with the SRS and the SDS.

The tests were performed by comparing the actual performance with the functional requirement for each of the features. The types of tests that were performed are defined in the Table 1 below. Table 2 provides a comprehensive list of the SRS functional requirements, a description of test and the status of the tests at a glance.

Table 3 provides the location of the detail results of the tests in the appendices by type of tests from Table 1. These are Appendices A – E, which include a description of the test, the test criteria for acceptable performance and actual test results.

Table 1: Description of the types of tests	
Test Type	Description
1. Unit level test	Scope to including the new classes. The class itself will be tested using a test driver to test the methods and members of the classes. 1.1 Unit testing on vendor system (encrypt and decrypt), 1.2 Unit testing on Estamp manager and Estamp functionality,
2. GUI interface test	Tests are performed to make sure that the features are implemented in a reasonable, usable way and that they meet the functional requirement. 2.1 GUI Configuration of the Estamp and the Estamp Book, 2.2 GUI Compose/ Send 2.3 GUI Receive/ Configure Filter panel,
3. System integration	Tests are performed to assure accurate coordination between the vendor for the Estamp purchase and the email client.
4. Security testing	Testing the potential hacks or impact to the vendor

2.1. Testing Results at a Glance

The following section reviews the tests performed on the prototype to meet as a minimum the requirements outlined in the SRS Document in Section 2.2 and provided the status of the testing as a result.

Table 2 The Status of Tests on the Functionality of the Prototype				
Feature	Ref #	Function	Type of test	Status of Tests
CE	R1.5	User may log an e-stamp for further use.	Testing of the function of the filter rule and the configuration function interface	Passed
CE	R1.6	User may invoke / revoke an e-stamp for / from further use.	Testing of the function of configuration of Estamp and Estamp Manager	Passed
CE	R2.5	User shall log the e-stamp for return use by click one checkbox in email client.	Test of reply button in compose and reuse of Estamp	Passed
CE	R3.3 .1	Email client shall check its logged e-stamps.	Test of Estamp Book GUI interface	Passed
CM	R1.1	User shall create a new email message in email client.	Already incorporated in MailPuccino	Passed
CM	R2.1 .2	User may select an existing e-stamp from logged e-stamps.	Testing of the compose GUI interface	Passed
CM	R2.1 .3	Email client shall request e-stamp from e-stamp vendor if email doesn't include an e-stamp as described in R2.1.1 and R2.1.2	System integration testing	Passed preliminary test with simplified vendor
CM	R2.3	Email client shall attach e-stamp to the header of email.	Testing of compose GUI, recipient GUI, and raw mail message	Passed
MM	R1.3	User shall reply and/or forward an email message saved in email client.	Already incorporated in MailPuccino	Passed

Table 2 The Status of Tests on the Functionality of the Prototype				
Feature	Ref #	Function	Type of test	Status of Tests
MM	R1.4	User shall delete an email message saved in email client.	Already incorporated in MailPuccino	Passed
MM	R2.4	User shall send email to the email server by click one button in email client.	Already in MailPuccino	Passed
RM	R1.2	User shall read a received email message in email client.	Already incorporated in MailPuccino	Passed
RM	R2.1 .1	User may use the existing e-stamp in an email header when it is a reply email.	Testing of the compose GUI interface	Passed
RM	R3.1	User shall begin receiving emails from email server by click one button in email client or when the email client is launched.	Already in MailPuccino	Passed
RM	R3.2	Email client shall extract e-stamp from the header of email before putting the email into folders.	Test of filter rules of Estamp value	Passed
RM	R3.3	Email client shall check the validity of e-stamp.	Test of the Estamp validity feature test	Superficial Demonstration for function of the filter rules
RM	R3.3 .2.2	Email client shall check the recipient of the e-stamp.	Test of the Estamp Manager/Estamp unit function	Not implemented
RM	R3.3 .2.3	Email client shall check the time frame of the e-stamp.	Test of the Estamp Manager/Estamp unit function	Not implemented

Table 2 The Status of Tests on the Functionality of the Prototype				
Feature	Ref #	Function	Type of test	Status of Tests
RM	R3.4	Email client shall separate emails with valid e-stamp from those without e-stamp or without valid e-stamp and put into different folders.	Test of filter rules Integration test	Passed
RM	R3.5	Email client may generate auto-reply message for emails without e-stamp or without valid e-stamp.	Test of Configuration GUI interface	Passed
VS	R2.2	E-stamp vendor shall generate and return e-stamp to email client.	System integration testing	Pass preliminary test with simplified vendor
VS	R3.3 .2.1	Email client shall request e-stamp vendor to authenticate the e-stamp.	Integration test of email client vendor	Not implemented, preliminary vendor does not provide this service. But, current RMI vendor server provides this service to vendor client.

Table 3: Types of Testing to be Performed on the Prototype			
Feature	Item	Description	Test Detail
All	1.1	Ad-Hoc Testing in coding.	Appendix A
All	1.2.	The unit test of the Estamp and Estamp Manager package.	Appendix B
VS	1.3	The unit test of the Estamp Vendor package.	Appendix C
CE	2.1.	GUI testing of the Configuration / Estamp book	Appendix D.1.
CM	2.2.	GUI testing of the compose / send	Appendix D.2.
RM	2.3.	GUI testing of the receive/filter panel	Appendix D.3
CM to VS	3.	Integration testing by checking the functional characteristics of features.	Appendix E
VS	4.	Security testing will be performed by a set of attacks on the integrity of Estamp.	Appendix F

3. Conclusion

From the results of the tests the implemented parts of the prototype have been tested and found to be free of error when using typical expected range values. Text fields accept any string. The data fields validate on digits only. Check boxes for true and false values only.

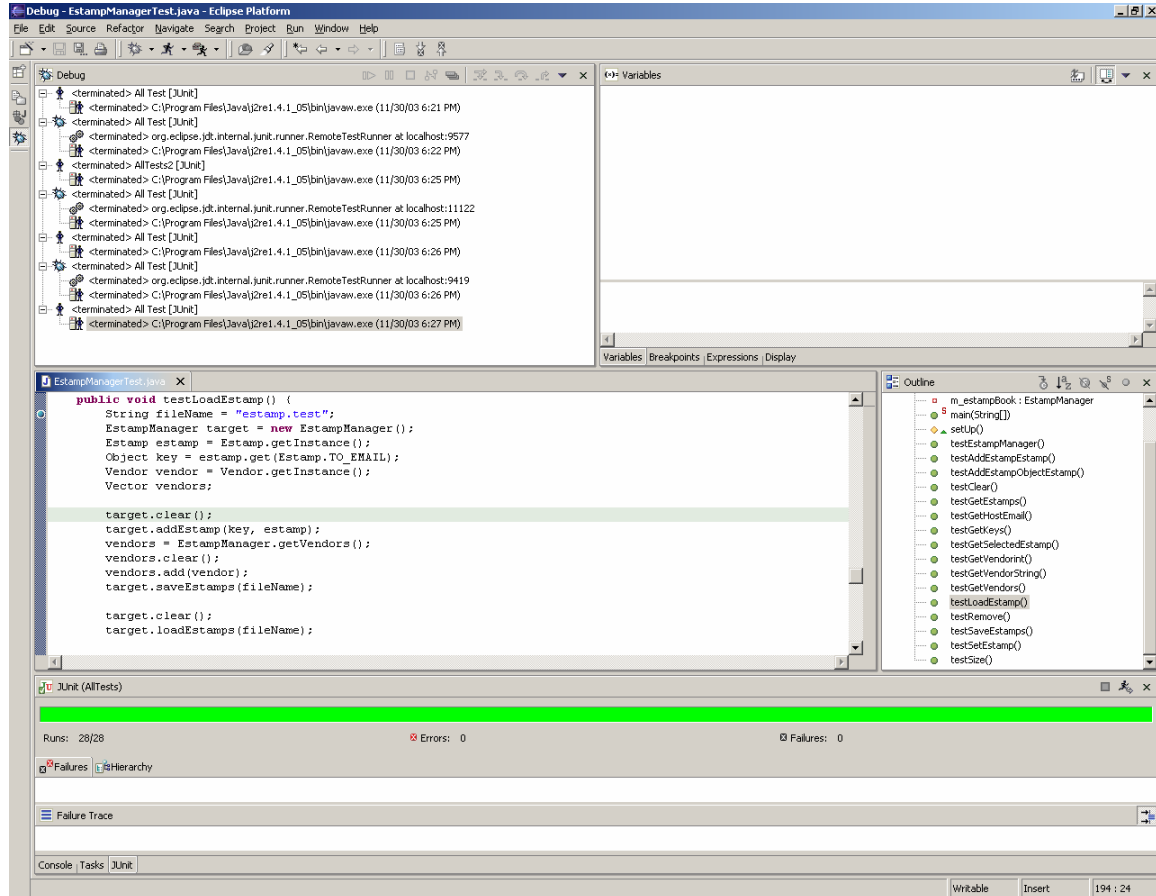
Appendix A. Ad-Hoc Testing of Modules

The programming practices of one programmer to another vary on the types of test that are performed on the module as it is being coded.

1. The programmers are encouraged to implement small sections of the code at a time.
2. The programmer is to use stubs with skeleton code which had been completed to a minimum structure for operability of the particular aspect whether it is a method or member of the object.
3. Frequent compiles of the code assures quick assessment of the cause of any problem that is evident.
4. Documentation of the code is stressed, and should be well constructed sentences that express the code in proper grammar.
5. The programmers are encouraged to use the Data Dictionary, or update the entries as the design or implementation dictates a change.
6. Every member is coded with a set, get and display functionality included in the methods of the object.
7. Every method has either a return value, or a status value returned.
8. The Test Driver of the method is to exercise each method and alter the state of each member.

Appendix B Unit Test Methods

The unit testing for the Estamp and EstampManager are based on the JUnit testing toolkit. The JUnit provides a platform for Java unit testing.



Appendix B.1. Unit Test for Estamp

Table 4: Details of the Estamp Functionality Testing			
Id		Anticipated Results	Results
ESa	testEstamp()	1. The estamp object created should not be null.	Passed
	Test default constructor: Estamp()	2. The fields in the estamp object should not be null.	Passed
ESb	testEstampObjectArray() Test constructor: Estamp(Object[])	1. The estamp object created by this constructor should not be null.	Passed
		2. The estamp object should contain the same fields as the parameters.	Passed
		3. The estamp object should not equal to the estamp object created by the default constructor.	Passed

Table 4: Details of the Estamp Functionality Testing			
Id	Description	Anticipated Results	Results
		4. The estamp object should not equal to another estamp object created by the same constructor with different fields.	Passed
ESc	testEstampVector() Test constructor: Estamp(Vector)	1. The estamp object created by this constructor should not be null.	Passed
		2. The estamp object should contain the same fields as the parameters.	Passed
		3. The estamp object should not equal to the estamp object created by the default constructor.	Passed
		4. The estamp object should not equal to another estamp object created by the same constructor with different fields.	Passed
ESd	testGet() Test get/set method: Object get(int) void set(int, Object)	1. The returned object of get method should equals the set method's parameter	Passed
ESe	testGetFields() Test getFields method: Object[] getFields()	1. The returned object array of getFields method should equals the parameters to create the estamp object.	Passed
		2. The returned object array of getFields method should not equals the parameters to create another different estamp object.	Passed
ESf	testSetFields() Test setFields method: void setFields(Object[])	1. The estamp object after calling setFields method should equal to the estamp object created by the constructor using the same parameter.	Passed
		2. The estamp object should not equal to the estamp object calling the setFields method with a different parameter.	Passed
		3. The estamp object should not equal to the estamp object created by default constructor.	Passed
ESg	testEqualEstamp() Test equals method: boolean equals(Estamp)	1. An estamp object should not equal to null.	Passed
		2. An estamp object should equal to itself.	Passed
		3. An estamp object should equal to the estamp object created by the same parameters.	Passed
		4. An estamp object should not equal to the estamp object created by different parameters.	Passed
ESh	testGetInstance(): Test getInstance method: Estamp getInstance()	1. The estamp object created by the getInstance method should not be null.	Passed
		2. The fields of the estamp object should not be null.	Passed
		3. The FROM_EMAIL field of the estamp object should equal to the host's email address.	Passed
		4. The ISSUE_DATE field of the estamp object should be today.	Passed
		5. The EXPIRE_DATE field of the estamp object should be tomorrow.	Passed
		6. The ACTIVE field of the estamp object should be true.	Passed
		7. The REUSABLE field of the estamp object should be true.	Passed
		8. The VENDOR_ID field of the estamp object should equal to the id of the first vendor in the vendor list.	Passed
ESi	testIsValid(): Test isValid method: Boolean isValid()		Skipped
ESj	testEncryptEstamp(): Test encrypt method:		Skipped

Table 4: Details of the Estamp Functionality Testing			
Id	Description	Anticipated Results	Results
	String encrypt(Estamp)		
ESk	testDecryptEstamp(): Test decrypt method: Estamp decrypt(String)		Skipped

Appendix B.2. Unit Test for Estamp Manager

Table 5: Estamp Manager Functionality Tests			
ID	Test and Description	Anticipated Result	Result
EMa	testEstampManager() Test the default constructor EstampManager()	1. The estampManager object created by the constructor should not be null. 2. The key set of the estampManager object should not be null. 3. The vendors list of the estampManager object should not be null.	Passed Passed Passed
EMb	testAddEstampEstamp() Test addEstamp method: void addEstamp(Estamp)	1. The estampManager object after calling addEstamp method should contain the estamp in the parameters	Passed
EMc	testAddEstampObjectEstamp() Test addEstamp method: void addEstamp(Object, Estamp)	1. The estampManager object after calling addEstamp method should contain the estamp in the parameters mapped by the key in the parameters	Passed
EMd	testGetEstamps() Test getEstamps method: Vector getEstamps(Object key)	1. The returned vector should not be null. 2. The returned vector should contain the estamp in the parameters of addEstamp method.	Passed Passed
EMe	testGetSelectedEstamp() Test getSelectedEstamp method: Estamp getSelectedEstamps(Object, int)	1. After calling addEstamp method, 1.1. The estamp object added to one index should equal to the returned estamp object from the same index by getSelectedEstamp. 1.2. The estamp object added to one index should not equal to the returned estamp object from any other indices by getSelectedEstamp. 2. The method should return null for empty or null key. 3. The method should return null for invalid index. 4. The method should handle the out of bound index.	Passed Passed Passed Passed Passed
EMf	testSetEstamp() Test setEstamps method: void setEstamps(Object, Vector)	1. The estampManager object, after calling setEstamp method, should contain the same estamps vector as in the parameters of the setEstamp method mapped by the same key as in the parameters of the setEstamp method. 2. The estampManager object, after calling	Passed Passed

Table 5: Estamp Manager Functionality Tests			
ID	Test and Description	Anticipated Result	Result
		setEstamp method, should not contain the estamps vector replaced by the estamps vector in the parameters of the setEstamp method.	
EMg	testGetKeys() Test getKeys method: Object[] getKeys()	1. After adding a estamp object to the estampManager object, the returned object array of getKeys method should contain the key in the parameters of the addEstamp method.	Passed
		2. After calling clear method of the estampManager object, 2.1 The returned object array of getKeys method should not be null.	Passed
		2.2 The returned object array of getKeys method should have no items.	Passed
EMh	testGetVendorint() Test getVendor method: Vendor getVendor(int)	1. After add a vendor object into the vendors vector, 1.1. The vendor object added to one index should equal to the returned vendor object from the same index by getSelectedEstamp.	Passed
		1.2. The vendor object added to one index should not equal to the returned vendor object from any other indices by getSelectedEstamp.	Passed
		2. The method should return null for invalid index.	Passed
		3. The method should handle the out of bound index.	Passed
EMi	testGetVendorString() Test getVendor method: Vendor getVendor(String)	1. After add a vendor object into the vendors vector, the vendor object added to one index should equal to the returned vendor object by getSelectedEstamp with the same vendor id as parameter	Passed
EMj	testClear() Test clear method: void clear()	1. After calling the clear method to estampManager object with non-empty estamps hash table, 1.1. The estampManager object should not contain a null estamps hash table.	Passed
		1.2. The estampManager object should contain an empty estamps hash table.	Passed
		2. After calling the clear method to estampManager object with empty estamps hash table, 2.1. The estampManager object should not contain a null estamps hash table.	Passed
		2.2. The estampManager object should contain an empty estamps hash table.	Passed
EMk	testLoadEstamps() Test loadEstamps/saveEstamps method: void loadEstamps(String) void saveEstamps(String)	1. After calling saveEstamps method and then the loadEstamps method, the estampManager object should contain the same estamps hash table.	Passed

Table 5: Estamp Manager Functionality Tests			
ID	Test and Description	Anticipated Result	Result
EMl	testRemove() Test remove method: void remove(String)	1. After removes the only one key from the estampManager object, the estampManager should contains a null estamps hash table.	Passed
EMm	testSize() Test size method int size()	1. The returned value of size method for a estampManager with only one estamp should be 1.	Passed
		2. The returned value of size method for a estampManager with no estamp should be 0.	Passed

Appendix C. Unit Test Driver for Estamp Vendor System

Test Description

The Estamp vendor system was test with a simplified vendor client and vendor server. The java RMI method was implemented successfully. We used the test code and test outlined in Software testing plan Appendix C.

Appendix D.1. GUI Test of the Configuration / Estamp Book

This next section outlines the results of the tests of the GUI interface that were performed on the Estamp Configuration feature:

Table 6: Results of the GUI interface Tests on the Configuration Panel.		
Item	Description	Results
1	“Config Estamp” panel should be shown with default values.	Passed
2	The “From Name”, “From Email Address” and “Issued Date” should be not editable.	Passed
3	The “Active” and “Reusable” should contain “Yes” and “No” in the drop down list.	Passed
4	The “Vendor” should contain all vendor ids in the drop down list.	Passed
5	Click “Add” button. Fill in a new vendor id and click “OK”. New one appears in the list	Passed
6	If click “Cancel” button. The vendor id in the “Config Estamp” panel should not be changed.	Passed

Appendix D.2. GUI Test of the Compose / Send

This next section outlines the results of the tests of the GUI interface that were performed on the Sending feature:

Table 7: Results of the GUI interface Tests on the Configuration Panel.		
Item	Description	Results
1	When we select the new message option from the write email icon on the panel, the compose email panel opens.	Passed
2	The Panel will be tested to see that if we use an Estamp it appears in the header.	Passed
3	If no Estamp is selected, that an Estamp does not appear in its header.	Passed
4	If the Estamp is selected, then the list of available Estamps to the appropriate person is displayed in the list.	Passed
5	Create Estamp information to be sent to the Estamp Vendor for use in creating an Estamp.	Passed

6	Confirm that the information entered into the Estamp appears in the Estamp list.	Passed
7	If the user clicks yes, check that Estamp saved will be listed in the EstampBook.	Passed
8	If the user clicks no, check to make sure there is no Estamp in the EstampBook.	Passed
9	When the email is sent, check and see that the complete email is received with the Estamp attached.	Passed

Appendix D.3. GUI Test of the Receive/Filter Panel

This next section outlines the results of the tests of the GUI interface that were performed on the Receive/Filter feature:

Table 8: Results of the GUI interface Tests on the Configuration Panel.		
Item	Description	Results
1	Selecting Config Filters from toolbox, results in the configuration filters panel opening.	Passed
2	When configuration Filter panel is open, several rule sets can be saved and recovered.	Passed
3	When a rule set is selected, the respective members appear in the rule set list.	Passed
4	When a particular rule is selected, it appears on the rules tab panel.	Passed
5	When selecting a rule, an action panel is accessible to construct the action.	Passed
6	The Actions selected will be to move the message to a folder.	Passed
7	The actions selected will be to log the Estamp.	Passed
8	Turn log on.	Passed
9	After the download of a selection of emails of varied conditions of Estamp quality such as valid, reusable, non-reusable and no Estamp.	Passed
10	Look at the EstampBook and confirm that the actions have taken place properly.	Passed
11	Turn Log off and recheck the EstampBook after subsequent emails.	Pending
12	Turn off the valid rule option.	Passed

Appendix E. Integration Test of Estamp Purchase of client and Estamp Vendor

The preliminary design to integrate the new components into the existing system required modification of the email client in order to implement the functionality needed

The importance of the integration testing is to assure the proper function of the vendor in response to an actual request from the email client for an Estamp. This testing was performed from the client side to initiate a purchase request in the email client. The vendor creates estamp and returns it. To confirm that it has functioned properly, we have examine the code and determined that hard coded stamps are not presented in the email client. Any estamp displayed is from the simplified vendor server.

Appendix F. Security Tests for Estamp Integrity

Implementation of the Vendor that used encryptions came late in the semester and we did not have time to do any security testing.