

***The Electronic Stamp
Mail Server and Client
Project
Part 3: Software Design Specifications***

by



SDS Final
November 3, 2003

This Software Design Specification 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 part on the IEEE Standard 1016-1998, IEEE Recommended Practice for Software Design Descriptions.

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
Draft 1	Oct 20, 2003	R. G. Reed E Shen Stanislav Ustymenko Yunwei Wang	All Initial	Framework for Design Document
Draft 2	November 3, 2003	R. G. Reed E Shen Stanislav Ustymenko Yunwei Wang	Inserting each section as assigned	Final edit

Preface

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

Table of Contents

1. Introduction	
1.1. Purpose	
1.2. Scope	
1.3. Definitions and acronyms	
2. References	
3. Decomposition Description	
3.1. Module Decomposition	
3.1.1. Module 1 Description	
3.1.2. Module 2 Description	
3.2. Concurrent Process Decomposition	
3.2.1. Process 1 Description	
3.2.2. Process 2 Description	
3.3. Data Decomposition	
3.3.1. Data Entity 1 Description	
3.3.2. Data Entity 2 Description	
4. Dependency Description	
4.1. Intermodule Dependencies	
4.2 Interprocess Dependencies	
4.3 Data Dependencies	
5. Interface Description	
5.1. Module interface	
5.1.1. Module 1 Description	
5.1.2. Module 2 Description	
5.2 Process Interface	
5.2.1. Process 1 Description	
5.2.2. Process 2 Description	
6. Detailed Design	
6.1 Module Detail design	
6.1.1. Module 1 Detail	
6.1.2. Module 2 Detail	
6.2. Data Detail Design	
6.2.1. Data Entity 1 Detail	
6.2.2. Data Entity 2 Detail	
4.1 Title of Class.....	28
4.1.1 Structure	28
4.1.2 Function.....	29
4.1.3 Interfaces	29
4.1.4 Application Data.....	29
4.1.5 Detailed Design Description (Methods)	

Table of Contents

Change History	3
Preface.....	4
Table of Contents.....	6
Table of Figures and Tables.....	7
1. Introduction.....	8
1.1. Purpose	8
1.2. Scope	8
1.3. Definitions and acronyms	8
2. References.....	9
3. Decomposition Description	10
Feature 1. Configuration Interface.....	11
Feature 2. Send	12
Feature 3. Receive	13
Filter Rules	13
Feature 4. Mail Manipulation	15
Feature 5. Vendor	16
Feature 6. eStamp Management	17
Feature 7. eStamp Interface	18
4. Dependency Description.....	19
5. Interface Description.....	20
6. Detailed Design.....	21

Table of Figures and Tables

Table 1 : Design Features and Reference Documents for Requirements	8
Table 2 : Features accessible to user through the Configuration Interface.....	Error!
Bookmark not defined.	
Table 3 :Sending Functions Implementation Design Classes.....	Error! Bookmark not defined.
Table 4 : Receiving Functions Implementation Design Classes.....	Error! Bookmark not defined.
Table 5: Email Filter Rules and Actions to Implement eStamp Functionality	Error!
Bookmark not defined.	
Table 6: Mail Manipulating Function Implementation Design Classes	15
Table 7 : Vendor Requirements	16
Table 8 : eStamp Management Requirements	17
Table 9 : eStamp Interfaces.....	18

1. Introduction

1.1. Purpose

This document outlines the Software Design Specifications as part of the design plan and specifications for adding the eStamp functionality to an existing email client, mailPuccino.

This document expands the functionality described by the features in the Software Requirements Specifications (SRS) v3.0. Each feature discussed will describe the existing functionality of mailPuccino, and the additional classes, attributes and methods to be implemented.

1.2. Scope

This document takes the features as outlined in the SRS and expands each of the features to include the design issues. The features are described by the Use Case names as in the descriptions and diagrams in the SRS. Additional requirements implemented based on the internal requirements are given numbers preceded by an S. These features are listed in Table 1.

Table 1 : Design Features and Reference Documents for Requirements		
<i>Feature Number</i>	<i>Use Case Description</i>	<i>Reference</i>
1.	Configuration	SRS Appendix A
2.	Send	SRS Appendix B
3.	Receive	SRS Appendix C
4.	Mail Manipulation	SRS Appendix D
5.	Vendor	SRS Appendix E
6.	eStamp Management	This document
7.	eStamp Interface	This document

1.3. Definitions and acronyms

2. References

IEEE Recommended Practice for Software Design Descriptions, IEEE Std 1016-1998
(Revision of IEEE Std 1016-1987) Available at
<http://web.nps.navy.mil/~nschneid/is3020/PDF/1016-1998.pdf>

3. Decomposition Description

- 3. Decomposition Description
 - 3.1. Module Decomposition
 - 3.1.1. Module 1 Description
 - 3.1.2. Module 2 Description
 - 3.2. Concurrent Process Decomposition
 - 3.2.1. Process 1 Description
 - 3.2.2. Process 2 Description
 - 3.3. Data Decomposition
 - 3.3.1. Data Entity 1 Description
 - 3.3.2. Data Entity 2 Description

Feature 1. Configuration Interface

SRS Appendix A contains the requirements specifications for this feature.

Table 2 : Features accessible to user through the Configuration Interface			
Use case SRS	Set up routine purchase to buy stamps from a vender		
Use case SRS	Setup routine to Authenticate stamps		
R1.5	User may log an e-stamp for further use		eStamp Manager and eStamp, using a filterAction, display with eStampbookFrame
R1.6	User may invoke / revoke an e-stamp for / from further use.		Member of eStamp Class

Feature 2. Send

SRS Appendix B contains the requirements specifications for this feature.

Table 3 :Sending Functions Implementation Design Classes			
<i>Ref #</i>	<i>Requirement Description</i>	<i>MailPuccino Functionality</i>	<i>Class Responsible for Additional Feature/Functionality</i>
R2.1.1	User may use the existing e-stamp in an email header when it is a reply to the email		eStamp.reusable
R2.1.2	User may select an existing e-stamp from logged e-stamps.		eStampManager
R2.3	Email client shall attach e-stamp to the header of email.		regularEmailComposer
R2.4	User shall send email to the email server by clicking one button in email client.	present	
R2.5	User shall log the e-stamp for return use by click one checkbox in email client.		eStampChoicePanel, eStamp

Feature 3. Receive

SRS Appendix C contains the requirements specifications for this feature.

Table 4 : Receiving Functions Implementation Design Classes			
<i>Ref #</i>	<i>Requirement Description</i>	<i>MailPuccino Functionality</i>	<i>Class Responsible for Additional Feature/Functionality</i>
R3.1	User shall begin receiving emails from email server by click one button in email client or when the email client is launched.	Automatic Retrieval	
R3.2	Email client shall extract e-stamp from the header of email before put the email into folders	By use of filter rules	Filter rule and eStampManager.getEStamp()
R1.2a	User shall read a received Regular email message in email client.	present	Filtered to specified LocalFolder
R1.2b	User shall read a received eStamp email message in email client.	Built in filter configuration	eStamp Manager using filterAction storing to specified LocalFolder
R3.3	Email client shall check the validity of e-stamp		eStamp.isValid()
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.	Built-in filter actions	Filter rule includes eStamp.isValid() and location folders for Valid, Invalid or Missing eStamps
R3.5	Email client may generate auto-reply message for emails without e-stamp or without valid e-stamp.		Filter action created to generate auto-reply message for Invalid or Missing eStamps

Filter Rules

MailPuccino already contains options to configure email filter rules. It will execute the action portion of the rule format when the condition that is set up in the rule returns a True state. Class members and class methods may be used as conditions for these rules. This flexibility allows us to check for the validity of a stamp with the execution of public methods of an eStamp class.

Table 2 contains examples of the filter rules to be implemented in the existing email client.

Table 5: Email Filter Rules and Actions to Implement eStamp Functionality			
<i>Rule Number</i>	<i>Condition</i>	<i>Methods to be Implemented</i>	<i>Action</i>
1.	Email without eStamp in header OR Invalid eStamp	getHeader(), eStamp.isValid()	Move to specified LocalFolder such as Trash, autoReply()
2.	Valid eStamp and reusable	getHeader(), eStamp.isValid(), eStamp.getReusable()	Move to specified LocalFolder such as Inbox, saveEStamp()
3.	Valid eStamp and not reusable	getHeader(), eStamp.isValid(), eStamp.getReusable()	Move to specified local folder such as Inbox.

Feature 4. Mail Manipulation

See SRS Appendix D contains the specifications for this feature.

Table 6: Mail Manipulating Function Implementation Design Classes			
<i>Ref #</i>	<i>Requirement Description</i>	<i>MailPuccino Functionality</i>	<i>Class Responsible for Additional Feature/Functionality</i>
R1.1	User shall compose a new email message in email client.	present	eStamp is provided by Vendor.purchase() if stamp is not available
R1.3	User shall reply and/or forward an email message saved in email client.	present	Reply can reuse stamp, forward is like a new compose.
R1.4	User shall delete an email message saved in email client.	Present localFolder	

Feature 5. Vendor

SRS Appendix E contains the specifications for this feature.

Table 7 : Vendor Requirements			
<i>Ref #</i>	<i>Requirement Description</i>	<i>MailPuccino Functionality</i>	<i>Class Responsible for Additional Feature/Functionality</i>
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		Vendor.getEStamp()
R2.2	E-stamp vendor shall generate and return e-stamp to email client.		Vendor.createEStamp()
S6	Transaction records should be kept		Vendor.Transaction
S7	eStamps are to be constructed using client provided information and encrypted and provided to the Client for writing to Header.		Vendor.createEStamp (clientinformation)

Feature 6. eStamp Management

This document contains the specifications for this feature.

Table 8 : eStamp Management Requirements		
S1	eStamps are stored in an easily retrievable format	Class eStampManager
S2	eStamp contains information and methods as needed	Class eStamp

Feature 7. eStamp Interface

This document contains the specifications for this feature.

Table 9 : eStamp Interfaces			
R3.3.1	Email client shall check its logged e-stamps.	eStampBookFrame.display()	
S4	The eStamps are to be easily used	eStampChoicePanel	
R3.3.2.1	Email client shall request e-stamp vendor to authenticate the e-stamp. (this is revised to be encrypted within the email client)		eStamp.isValid() using Vendor.pubKey to decode the eStamp values
R3.3.2.2	Email client shall check the recipient of the e-stamp.		eStamp.toEmailAddress, eStamp.fromEmailAddress
R3.3.2.3	Email client shall check the time frame of the e-stamp.		eStamp.issueDate, eStamp.expireDate
S5	The eStamps can be listed	eStampPanel	

4. Dependency Description

4.1. Intermodule Dependencies

4.2 Interprocess Dependencies

4.3 Data Dependencies

5. Interface Description

5.1. Module interface

5.1.1. Module 1 Description

5.1.2. Module 2 Description

5.2 Process Interface

5.2.1. Process 1 Description

5.2.2. Process 2 Description

6. Detailed Design

6.1 Module Detail design

6.1.1. Module 1 Detail

6.1.2. Module 2 Detail

6.2. Data Detail Design

6.2.1. Data Entity 1 Detail

6.2.2. Data Entity 2 Detail