Use-Case Model

The use-case model used in this document to describe (mail)^n is accomplished using the 'fully dressed' format. Given the nature of the system, it is empirical to enumerate as many details as possible about the system for completeness.

Before presenting the use-case scenarios, we must illustrate the relations between actors and their goals. Namely:

Actors	**Goals**
Web-server	Inter-median between the database and the mail server and the user. It also provides an interface for the user.
Mail-server	This is where the emails are stored at first. They are held there until they are read and processed into the DB
Database	Stores mail messages, contact information, and task information. Also returns information when queried.
User	Must be able to send, receive, read, and store emails. Must also be able to manage contact groups and tasks.
Administrator	To create, or delete an account within the system.

Having identified the set of 'actors', and their respective goals, we continue to map out the first use case scenario, in which the primary actor is a user.

Use Case UC1: Processing Messages

Primary Actor: User
Stakeholders and Interests:

- User: Wants to navigate through the email system easily and be able to send, receive and store emails. In addition, the user also wants the ability to manage contact groups, and edit tasks.
- Administrator: Wants to be able to create, modify or delete accounts.
- Company: (given that (mail)^n is used within a corporation, although it could be used by anyone.) Wants to ensure that all of the employees using the system can easily do so without the need of someone to assist the employees. Also, the system has to be virus resistant, because employees will rely on having the ability to store information. The information must be guarded for safe retrieval.

Preconditions: User is connected online, and is using a modern web browser.
Success Guarantee (Post conditions): Emails are sent, received and stored. Tasks are correctly managed, and contact information is correctly updated. Administrators have full control over the user accounts.

Main Success Scenario (or Basic Flow):

1. User connects to the Internet.
2. User goes to the (mail)^n web site.
3. User enters his/her user name and password.
4. The system verifies that both user name and password match with the contents in the database.
5. The user is now logged into the system.
6. The first screen that the user sees is the 'inbox' screen.
7. From that point, the user may click on a particular unread message to view its content, or choose to perform any other available option.
8. Assuming that the user reads an unread message, and wishes to reply, either to the sender, or to the sender and the CC's, the user clicks on the "Reply" button.
9. A new screen appears in which the user may enter the recipient information, a subject line, and the body of the email.
10. After writing the email, and filling out the "To:" field, the message is ready to be sent.
11. The user clicks on the "Send" button.
12. The user is then presented with a "Message successfully sent" message upon completion of the sent action.
13. The user is redirected to the inbox screen, where he/she may continue to browse emails.
14. The user chooses to terminate his/her session.
15. The user successfully logs out.

Extensions (or Alternative Flows):

At any time, System fails:

To support recovery and correct storing of emails, ensure that all messages are stored into the database once the messages are read by the user. Emails will not be lost due to a local machine failure.

1. User restarts machine, connects to the internet, logs in, and emails are still present.
2. System keeps messages in external machines, not available to the user, so the messages are kept safe.
3. Invalid identifier:
    1. System does not match the string of username, and/or password with existing usernames and/or passwords in the database.
    2. System rejects entry, and redirects user to the initial screen where the user may attempt to log in once again.
4. Internet connection is lost:
    1. When reconnected, the user must enter his/her username and password again.
    2. Loosing Internet connection is the equivalent to logging out.
5. System detects failure to connect to mail server:
    1. An error message will display, stating that the mail server is temporarily out of service.

6a. The user does not fill out required fields of a message:

1. At least one valid recipient's address must be provided to send a message.

6b. The user enters an incorrect email address in the recipient’s box:

   1. The system will email the user shortly after the message has been sent, notifying the user that the email address does not exist.

7. The user wants to send an email attachment larger than 10MB:
   1. The system will display an error message stating that the attachment exceeds 10MB (the limit size of attachment per email).
8. The administrator wants to enter a new user’s information:
   1. The new username must not exceed 12 alphanumeric characters.
   2. The new password must not exceed 8 alphanumeric characters.
9. The user wants to add a new task:
   1. The user may not add a new task that takes place earlier than the present time at which the user wants to add the new task.
   2. An error message is presented stating that the task must be set to a future date.

Special Requirements:

- The front end of (mail)^n requires a modern web browser that understands Dynamic HTML and Cascading Style Sheets.
- Users are required to have an Internet connection.
- User-friendly front end.

Frequency of Occurrence: The system could be used as frequently as needed.

Open Issues:

- What can be customized for using the system within corporations versus using it publicly?

This particular fully dressed Use Case pinpoints the various ‘paths’ that a given user may take when using (mail)^n. It illustrates the ‘happy path’ as well as all of the areas in which the system may display an error message due to an improper use. It is explicit in pointing out the system’s limitations, as well as the user’s limitations. This information is essential that is needed to have a formal set of system constraints.

Vision

Revision History

Version	Date	Description	Author(s)
Inception draft	October 27, 2004	First draft. Will be refined during the elaboration phase.	Joseph Aulisi Elias Rosero Jwalant Dholakia

Introduction

Our team sees (mail)^n as an email program designed to house all the email received by an organization and its users in a form that is easily searchable, expansive, and permanent.

Positioning

Business Opportunity

Organizations today have a wide variety of email systems from which to choose, though none of them guarantee against data loss, and many are prohibitively expensive to employ and maintain. Although many systems offer users the ability to view their email via a client program installed on the user's machine, or via a web-based client, email itself is stored in proprietary data files, or plain text files, which can grow quite large, thus becoming unwieldy, and can have an adverse effect on the speed at which mail is retrieved and read.

Problem Statement

Email systems today are not secure and can potentially lose the data they were designed to store. Many systems that store email messages in a proprietary format keep mail in a single file, which, over time, is at a high risk of corruption. When this corruption occurs, users can expect to lose all of their messages. Some systems, due to their omnipresence, are often exploited by malicious virus attacks, which can compromise a client machine and propagate around the world in a matter of minutes, leaving a path of infected computers with no chance of recovering data.

Product Position Statement

(mail)^n is designed for use by small to mid-sized organizations that are looking for an email system that provides its users with the ability to view their email securely from any web browser. Each user's email is stored in a database, which offers a virtually unlimited mailbox size for each user, and offers the ability to quickly locate past emails. Apart from being a rock-solid email program, (mail)^n offers a convenient contact management tool, and a task database, which allows corporate users the ability to schedule inter-office meetings.

Alternatives and Competition

Currently, there are many alternatives to (mail)^n, including Microsoft Exchange, Lotus Notes, and even outsourced email services. One of the problems with outsourcing email is the mailbox size limit that vendors place on services. (mail)^{n} offers an unlimited mailbox size for all users on the system. Microsoft Exchange and Lotus Notes are pricey alternatives, requiring the purchase of hardware and software, but also the presence of someone with the skills to administer these services.

Stakeholder Descriptions

Stakeholder (Non-User) Summary

Major stakeholders will be business owners in a position to decide the best, most efficient and cost-effective way to manage mail. Those who decide on which mail system to roll out are the ones who stand to lose the most, should the system fail. (mail)^{n} offers a worry-free solution.

User Summary

Two types of users, administrators and employees, will be using (mail)^{n}. Employees are standard users with no administrative privileges. Administrators, on the other hand, have more privileges, and are responsible for adding employees to the system, among other tasks.

Key High-Level Goals and Problems of the Stakeholders

High-Level Goal	Priority	Problems and Concerns	Current Solution
World-wide accessibility	High	Will the system be available when users travel?	Most solutions offer a web access to email.
Unlimited storage	High	Will the system grow with the company?	Online services offer a service with a hard limit on disk quota.
Security	High	Will the system be resistant to viruses, worms, and prying eyes?	Bugs in current systems can allow for such attacks.
Ease of use	High	Will there be a need for training?	Most email systems are easy to use, but can become bloated with too many features.

User-Level Goals

–Employee: send, receive, and save emails, store contact information, set up tasks.
–Administrator: backup mail system, set up users easily.

User Environment

Employees and administrators will be able to access (mail)^{n} with any modern web browser, from any computer connected to the environment.

Product Overview

Product Perspective

(mail)^{n} will reside on a robust computer system housed within the walls of company, or within a co-location service. It will provide services to all the users of an organization, and will need to collaborate with an external mail service, and a database, as shown in figure Vision-1.0.

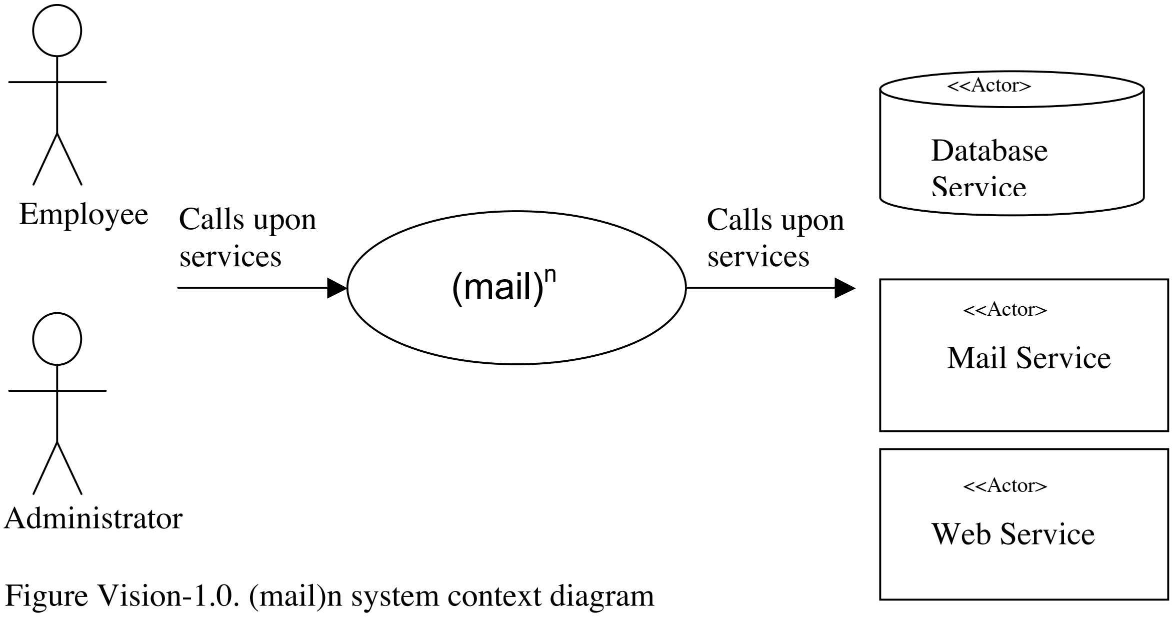

Figure Vision-1.0. (mail)n system context diagram

Summary of Benefits
Because (mail)^n is web-based, the system is accessible from anywhere in the world. Users are able to login and access all the information they need. An employee will be able to save every correspondence they receive and send while working at an organization. Since the system is regularly backed up, there is no need to fear loss of emails. (mail)^n is not prone to virus attacks, so the data remains safe. People using (mail)^n will already be familiar with point navigating web sites, so there is no need to train them to use this system.

Summary of System Features

- all data is stored in a database, which allows for fast searches, and eliminates the possibility of email file corruption
- there is no need to delete emails
- users can store all contact information in the system
- users can set up meetings by sending an email to fellow employees. If fellow employees agree to meet at the specified date and time, the tasks for the agreeing employees are updated to reflect this appointment
- the contact module easily interfaces with mail module, which makes addressing emails easy

Other Requirements and Constraints

For an in-depth look at the design constraints, functionality, usability, reliability and performance, see the Supplementary Specification and Use Case models.

Supplementary Specification

Revision History

Version	Date	Description	Author(s)
Inception draft	October 27, 2004	First draft. Will be refined during the elaboration phase.	Joseph Aulisi Elias Rosero Jwalant Dholakia

Introduction

This section documents various (mail)^n requirements and specifics not captured in the use cases.

Functionality

Logging and Error Handling
Make a log of all identified errors for future review.

Pluggable Business Rules
Depending upon the requirement of client Company some minor modifications in the look and feel of the system (like changing menu system, color, adding company logo etc) can be added.

Security
(mail)^n is a web-based e-mail system. So, in order to ensure that only company employees who have the right to use the system are using it, and no one else, user authentication is of prime importance.

Usability

Human Factors

- (mail)^n is going to be used in an office environment by employees and company administrators. The employees might end up spending a good deal of time using (mail)^n in any given day. So, it is extremely important that (mail)^n is user friendly.
- The text size should be large enough so that it does not get difficult and cumbersome to read messages after an entire day in front of the computer screen.
- The employee should be given an audible signal when a new mail arrives because he might not be constantly monitoring the (mail)^n browser to check for new mails (Proposed).
- It is also important that the new incoming messages do not take too much time to open up as this might increase the frustration and stress on part of the employees using (mail)^n.

Reliability

Recoverability

If the server or internet connection is down, an employee might not be able to access his mails on (mail)^n. This might be a bottleneck for the system but it does not seem to be a critical issue. No further decisions are made about it as of now. However, if the system goes down because of some internal error (like database server crash, invalid query etc) the system should be set up in such a way that it recovers as much unsaved data as possible and restarts itself.

Performance

A consistent and efficient performance of (mail)^n is important. But after all, (mail)^n is a web-based application involving a Wide Area Network. And sometimes this network might go down and bring the application to a halt. Factors like these create a bottleneck on the performance of (mail)^n. Further performance issues will be identified once the system is up and running and in production use.

Supportability

Adaptability

(mail)^n can be set up to be used by an administrator or an employee. Depending upon the user type certain privileges (like adding new users, deleting users etc) are allowed.

Configurability

(mail)^n has the functionality to classify incoming messages into various sub-folders depending upon desire of the system user. (mail)^n is definitely configurable in making e-mails organized and systematically classified.

Implementation Constraints

(mail)^n is developed using Java, and PostgreSQL as well as other components. The main reason for this was ease of use and ability to modify code efficiently in the future if needed.

Purchased Components

We were able to set up the server on a test basis using existing hardware. No purchased components are identified as of now.

Free Open Source Components

(mail)^n makes heavy use of open source components. PostgreSQL is an open source database. The pages of (mail)^n are served with Tomcat, an open source Java environment from the Apache Software Foundation. Java itself is free to use.

Interfaces

Noteworthy Hardware and Interfaces

- Server
- Internet Connection Hardware

Software Interfaces

- Web Browser

Domain Rules

ID	Rule	Changeability	Source
RULE 1	User Authentication required before showing (mail)^n mails page	Low	(mail)^n policy
RULE 2	Idle connections for more than 15 minutes automatically logged out of (mail)^n	Depending upon business rules of a company for which (mail)^n is set up, modification can be made into the system to disable this feature	Company policy
RULE 3	Once browser window is closed, the user is automatically logged out of the system. Re-authentication required to login again	To enhance security of (mail)^n we keep changeability of this rule to Low	(mail)^n policy

Legal issues

Making sure all terms and conditions specified in the open source software user-agreement are satisfied.
Strictly enforce (mail)^n user agreement upon client companies.

Information in Domains of Interest

Network connectivity

If the Local Area Network on the client company site goes down, all local machines on the company network might not be able to access mail using (mail)^n even if the Internet

Network and (mail)^{n} server are up and running. In this case, the company might be willing to invest in a stand-alone computer that is directly connected to an outside network to access important messages.

Glossary

Revision History

Version	Date	Description	Author(s)
Inception draft	October 27, 2004	First draft. Will be refined during the elaboration phase.	Joseph Aulisi Elias Rosero Jwalant Dholakia

Term	Definition and Information	Aliases
(mail)^{n}	Email system where all message information is saved to an object relational database, allowing unlimited mailbox size, and a secure, easy to use interface	
Message	An individual email sent or received by a user of the system	Mail, email
Inbox	A virtual repository of new, unsorted messages	
Sent folder	A virtual repository recording messages that have already been emailed from the user's account	
Deleted folder	A virtual repository of messages a user has removed from the inbox, but is not ready to permanently remove from the system	
Folder	A user-defined repository of filed messages	
Log in	The action of supplying a id and password in order to gain access to (mail)^{n}	
Log out	The action of leaving (mail)^{n} securely.	
Session	The time starting when the user logs in until the user logs out	
Module	A distinct feature of (mail)^{n}, including the mail module, the contacts module, and the tasks module	
Contact	Any entry into the contacts database that includes information about a person, such as name, street address, business, email address, and telephone number	
Task	Any entry into the tasks database that describes an event, or appointment with the attributes of date and time	Event, appointment, meeting
Search	An action of iterating through the database to find a particular piece of information	
Username	A unique identifier for each user of (mail)^{n}	Id, employee id, uid, e-id, eid
Employee	Any identifiable user of (mail)^{n}	
Administrator	An employee of an organization that has extended privileges on the system, such as user creation, password assignment, and user deletion	admin
Database	A central repository of all email messages, contacts, and tasks	

Term	Definition and Information	Aliases
To	The addressee of a new mail message	recipient
From	The original sender of a mail message	
CC	Additional recipients of a mail message	
BCC	Blind carbon copy, which conceals recipients of a message from other recipients	
Flag	An attribute of a mail message	
Attachment	A separate file that is part of a mail message	
Forward	The action of sending a received message to another party	
Reply	The action of answering a received message with a response the to the sender	
Reply All	The action of answering a received message with a response to the sender and all parties to which the message was addressed	
Flagged	An attribute of a message that has been distinctly marked, as to alert the user that this message needs additional attention	
Body	The part of an email that carries the actual semantic message	
Subject	The part of an email that contains a summary of the body of the message	
Timestamp	The date and time a message was sent or received	
Read	(pronounced reed) The action of opening an email message and reading its contents	
Read	(pronounced red) The state of a message after it has been opened and scrutinized	
Unread	The state of a message before it has been opened	
Mail-id	A unique identifier for each mail message	mailid
Group	A user-defined list of intended recipients of messages	
Invitation	A message sent to a group of employees asking them to attend a meeting	Envite, evite, invite, nvite
Public	A contact that is visible to all users of (mail)^{n}	
Private	A contact that is visible to only the user that created the contact	
Category	An attribute of the Contacts table that records if a contact is public or private	