2009

# Share world: A web-based reuse managment system

Raymol Philomina Rappai

# SHARE WORLD: A WEB-BASED REUSE MANAGEMENT SYSTEM

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

by

Raymol Philomina Rappai

March 2009

SHARE WORLD: A WEB-BASED REUSE MANAGEMENT SYSTEM

A Project

Presented to the

Faculty of

California State University,

San Bernardino

by

Raymol Philomina Rappai

March 2009

Approved by:

[redacted]

Dr. Ernesto Gomez, Chair, Computer Science and Engineering

3/11/2009
Date

[redacted]

Dr. Tong Lai Yu

[redacted]

Dr. Kerstin Voigt

# ABSTRACT

In a community there is always a need to share and get items. This project enables the sharing of items in a web-friendly environment. It is a web-based application that allows managing communities or groups, users and their items. This enables them to share and keep track of items within a group. The project is a self-maintained web-based application managed by the community. Keeping track of the movement of the items enables the members to know about the whereabouts and details of the items. Also the system allows searching the items and mandates the movement of items between different members of the system. The application can be easily customized to adapt to various communities. The site works well to be run for non-profit purpose but also could be extended for commercial use.

## ACKNOWLEDGMENTS

I thank the faculty of Computer Science and Engineering Department for giving me an opportunity to pursue my Masters in Computer Science at the California State University, San Bernardino. I express my sincere appreciation to my graduate advisor, Dr. Ernesto Gomez who offered me assistance and directed me through the entire effort. I also thank my other committee members, Dr. Tong Lai Yu and Dr. Kerstin Voigt for their valuable support.

## TABLE OF CONTENTS

CHAPTER FIVE: SYSTEM VALIDATION

CHAPTER SIX: MAINTENANCE MANUAL

CHAPTER SEVEN: CONCLUSION AND FUTURE DIRECTIONS

# LIST OF TABLES

## LIST OF FIGURES

# CHAPTER ONE

# INTRODUCTION

## 1.1 Introduction

The system Share World - Web-Based Reuse Management System (WRMS) is an online website that enables members to share various items between themselves and keep track of them. Any individual can be a member of the system, once they are approved by the reference they select. The reference is any member of the group that the new member wishes to join. The members of the system can post various items, keep track of those items and view the items they hold with themselves. The items are categorized and added for easy search. Each of the members can view the details of the item as well as edit their profile and details of the items they own. The members can revoke the items they own and even remove themselves from being a member of the system at any time. Additional administrative abilities like "able to update" and "delete the entities" are given to the administrators for the maintenance of the system. The members can add advertisements links to the system, which serve as a major source of revenue from donations by which the cost of maintenance of the system could be covered. The system also takes care of various security measures. These security measures include restricting the unsigned members.

## 1.2 Purpose of This Project

The purpose of this project is to provide a well maintained system where individuals could share items and keep track of them. The system encourages the idea of reusability, which could be a good contribution from any individual in giving back to the environment. The need for such a system is also relevant in cases where individuals have different items that are not being used and do not want to go through the hassle of reselling them. The system mandates the movement of the items within the members thus ensuring that the item is being reused. The owner is able to revoke and edit his items at any time. The project helps the members to reach out to different people and help the community.

## 1.3 Background of Project

Being able to share and reuse items is something that might be useful to many people. This brings the need for a system that aids in the management and coordination of these shared items. Existing similar sites like "E-Bay" allows for the users to get and post items between other users. But the site serves for a commercial purpose. Sites like "Craigslist" helps the users post advertisements of any item. But the system does not keep any account for any user or item. Or it cannot be a community-driven. Developing a site that keeps track of the user of the items which serves as a community driven project is a unique idea.

## 1.4 Project Products

This project would lead to the following products:

- Implementation of WRMS. A working website is based on J2EE technology. The website uses HTML, CSS, Dojo, JavaScript, Ajax as the font-end technologies. The middle-tier implementation is done with J2EE (JSP, Servlets), Web Services [3], JDBC. The application is connected to the database MySQL [8] using the JDBC connectivity standard.
- Users Manual. An implementation manual will be available for the user. Every page on the website has a link to a HELP guide. This HELP guide will answer

all the frequently asked questions and also highlight ways of using the system.

- Systems Manual. A project report (this report) will be available with design details and specifications.

## 1.5 Acronyms and Abbreviations

| | |
|---|---|
| Ajax | Asynchronous JavaScript and XML |
| CSS | Cascaded Style Sheet |
| HTML | Hypertext MarkUp Language |
| JDBC | Java Database Connectivity |
| JSP | Java Servlet Pages |
| MVC | Model View Controller |
| RSS | Really Simple Syndication |
| REST | Representation State Transfer |
| SHA1 | Secure Hash Algorithm 1 |
| XML | Extensible MarkUp Language |
| API | Application Programming Interface |
| GUI | Graphical User Interface |

# CHAPTER TWO

# WEB BASED REUSE MANAGEMENT SYSTEM ARCHITECTURE

## 2.1 Project Design

This project Share World implements a web system to share goods with other members. The components needed to implement WRMS are a graphical user interface, a web server, a database interface, Application Programming Interface (API), a database server. The following criteria were used to choose the components for building WRMS:

- The components should be shareware, means they should be freely available for non-commercial purposes.
- An independent database server, so that different and new versions of the server can be plugged in easily.
- Be a part of the standard that they do not depend on a specific operating system and hence are portable easily across systems with ease.

The project is implemented in a Java Platform. The project uses different J2EE [6] technologies such as Servlets and JSP. The project is implemented using the Model-View-Controller [6] pattern. MVC is both a design and architectural pattern used in the field of software engineering. Successful use of the MVC pattern isolates the business logic from the user interface considerations, resulting in an application where it is easier to modify either the visual appearance of the application or the underlying business rules without affecting the other. In MVC, the Model corresponds to the information, which means the data of the application and the business rules used to manipulate the data. The View represents the elements of the user interface and the Controller manages details involving the communication to the model from the user interactions. Figure 1 shows the Model-View-Controller diagram.

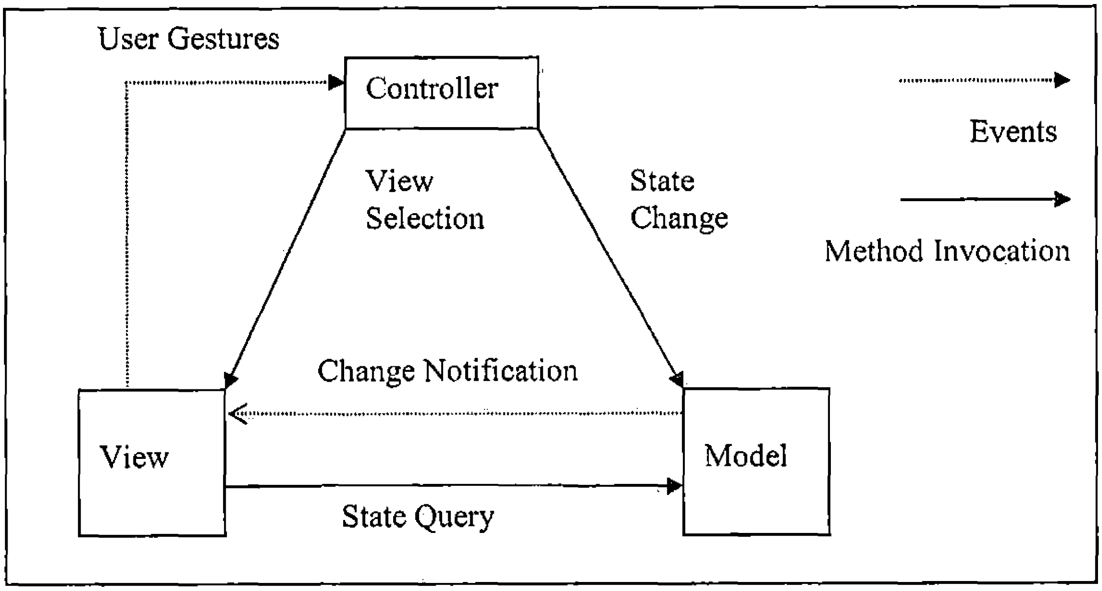


Figure 1. Model View Controller Diagram

The presentation GUI will be implemented with HTML, CSS, JavaScript, Dojo, Ajax, and JSP. The scripting of the application was done using JavaScript. The middle tier is implemented with JSP, Servlets. The system consumes a few external Web Services [3]. The database of the application is the JDBC compliant MySQL. Tomcat [7] serves as the Web Server. Figure 2 explains the 3-tiers.

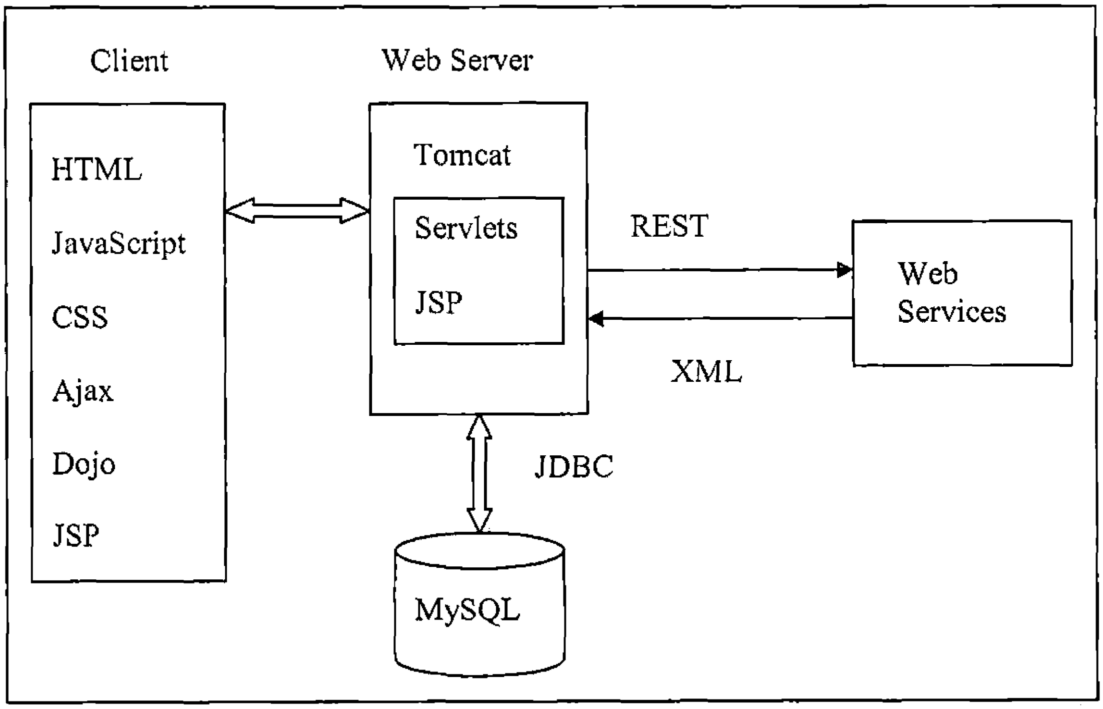


Figure 2. 3-Tier Architecture

The user interface components are built using HTML 6.0, Forms [3], Frames [3], CSS, Dojo, JavaScript, JSP, and

Ajax. HTML [3] provides a means to describe the structure of text-based information in a document. HTML is written with tags, which are surrounded by angle brackets. HTML can include embedded scripting language code such as JavaScript which can affect the way web browsers and other HTML processors behave. JavaScript [4] is a scripting language widely used for client-side web development. There is always the need to validate the input data before accepting them and stored in the database. Client side validations can prevent unnecessary network traffic. Otherwise the validations had to be done on the server side before inserting to the database.

In web development, Cascading Style Sheets (CSS) [3] is a style Sheet language used to describe styles to the element tags written in a markup language. Dojo [2] is the Open Source, portable JavaScript toolkit that lets you build interactive modules quickly, and construct Ajax requests with ease. Ajax (Asynchronous JavaScript and XML) [4], is a combination of web development techniques used for creating interactive web applications. A primary characteristic is the increased responsiveness and interactivity of web pages achieved by exchanging small amounts of data with the server "behind the scenes". The entire web page do not have to be reloaded every time there is a need to fetch data from the server. This increases the web page's interactivity, speed, functionality and usability. The password entered by the user for login is message digested using SHA1 encryption algorithm [6] and stored in the database. Message Digest [6] is the representation of text converted in the form of a string of digits, created using a formula called a one-way hash function. Each time a user enter a password, the system creates a digest of the password and checks with the digest saved in the database.

The Servlets [6] invoke the service request and processes the service response. Servlets add dynamic content to a Web Server using Java Platform. The Servlets connect to the database through the Java Database connectivity (JDBC) [6] standard. The project uses a component type called filters. Filters [6] are used for blocking request based on user identity. A filter dynamically intercepts the requests and responses to transform or use the information contained in the requests or responses. Filters typically do not themselves create responses. Instead the filters provide universal functions that can be attached to any type of Servlets or JSP page.

The database used for WRMS is MySQL. MySQL is a real multi-user database and is open source software. To use it, simply activate MySQL in Linux [9]. Since MySQL is compliant to the JDBC standard, the same code could be used to link with another database by changing to the proper JDBC driver, thereby making it database independent.

## 2.2 Software Interfaces

| | |
|---|---|
| Internet Browser | : Mozilla Firefox 3.0 or higher,<br>Internet Explorer 6.0 or higher. |
| Operating System | : Windows 98/2000/XP or Linux |
| Database | : MySQL 4.0 or higher |
| GUI | : HTML, JSP, Dojo, Ajax, CSS and JavaScript |
| Programming Language | : Java 1.5, J2EE 1.4 |
| Database Connector | : JDBC |
| Web Server | : Tomcat 1.5 |

# CHAPTER THREE

# DATABASE DESIGN

## 3.1 Data Analysis

The data designing and implementing the schema of the database depends on the properties of pages and users. The user data needed by the system are the user name, password, address, zip code, email and phone. The role of the user is determined by the system at the time of login and the user is displayed appropriate message and directed to the respective pages. All the data entered are checked before entering to the database. Some of the error messages displayed include pointing to the user about the validity of the data entered. A unique number identifies every record stored in the database.

## 3.2 Database Schema Conceptual Model

In designing the schema for the WRMS database, distinct entities have been identified. The first includes entities having relationship between them. The second is the description of the entities and their attributes.

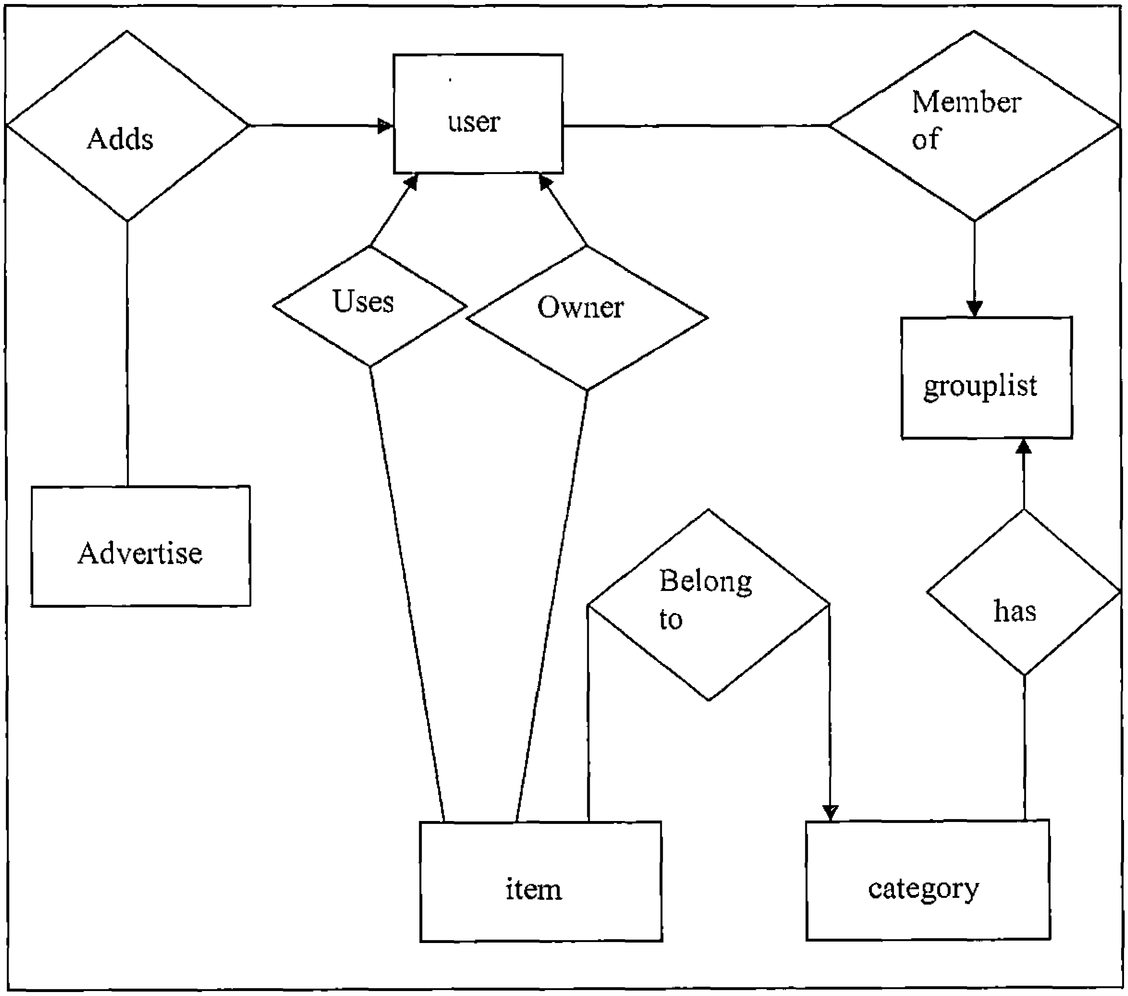


Figure 3. Entity Relationship Diagram

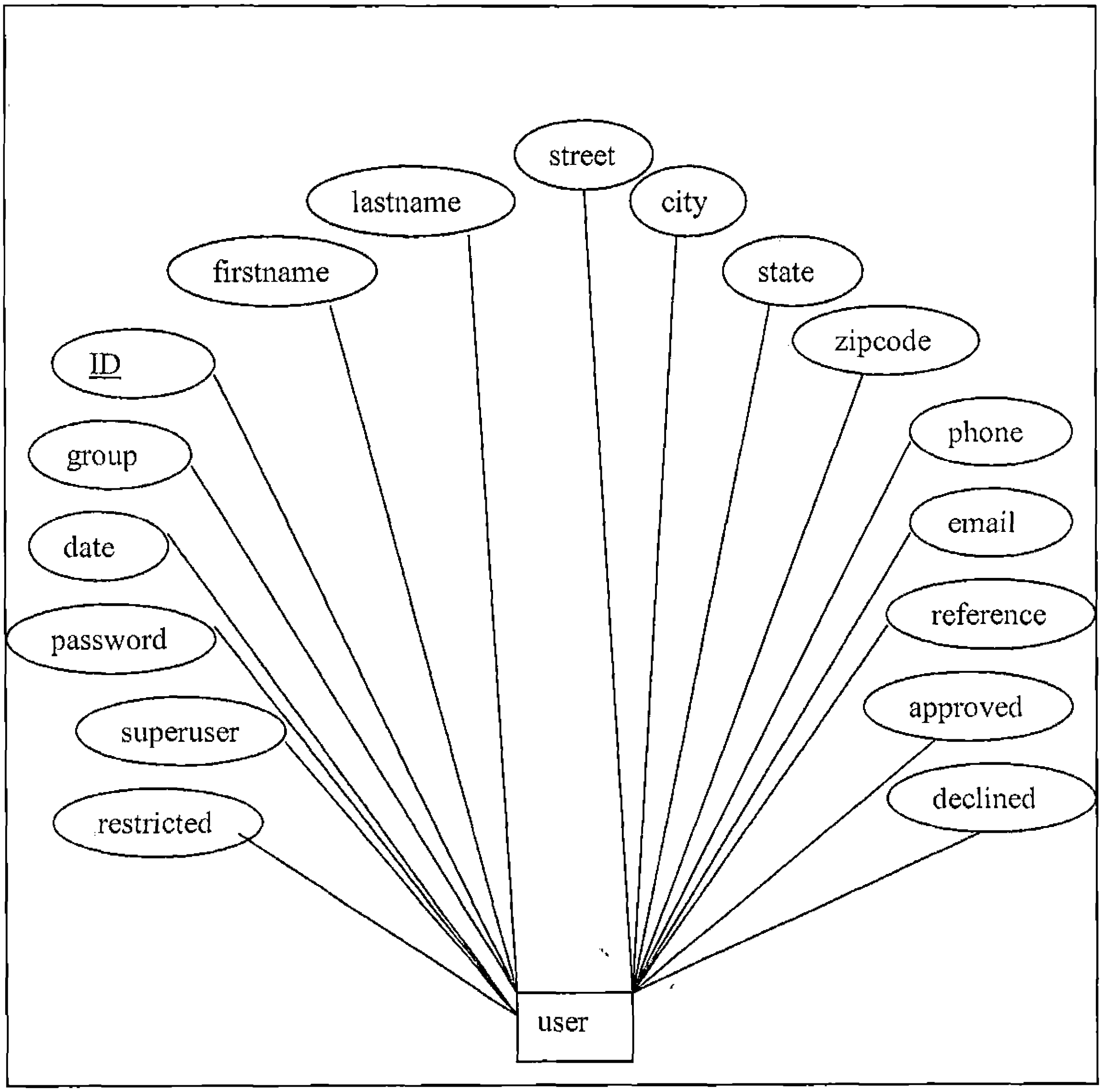


Figure 4. User Entity and Attributes

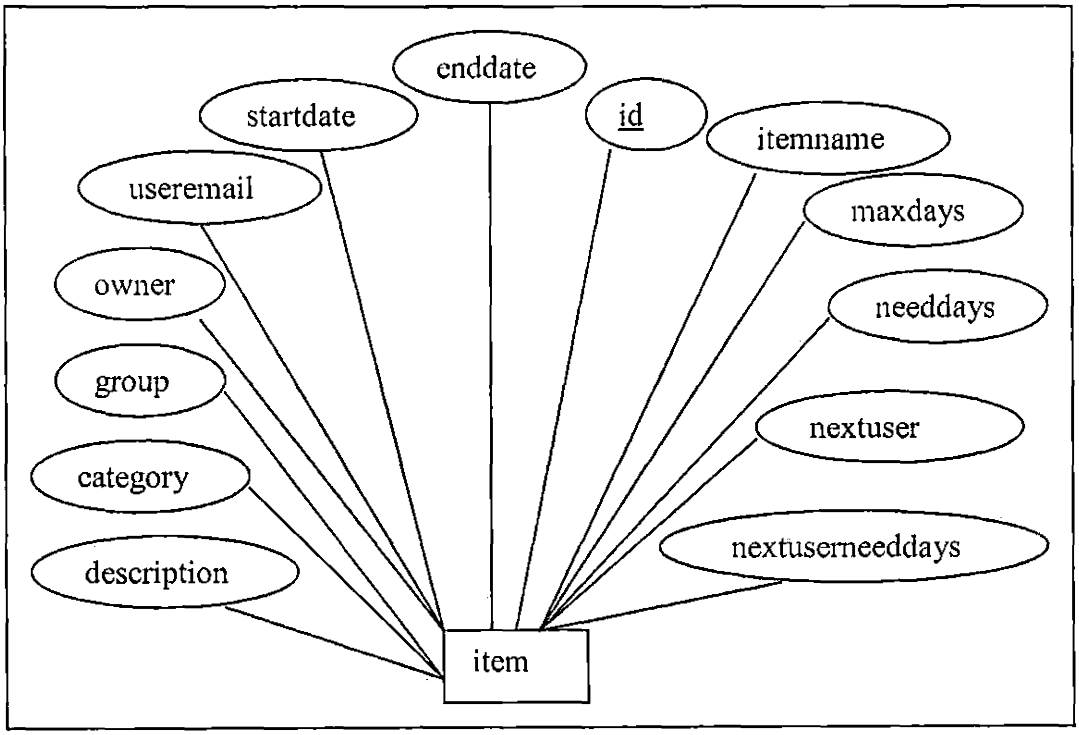


Figure 5. Item Entity and Attributes

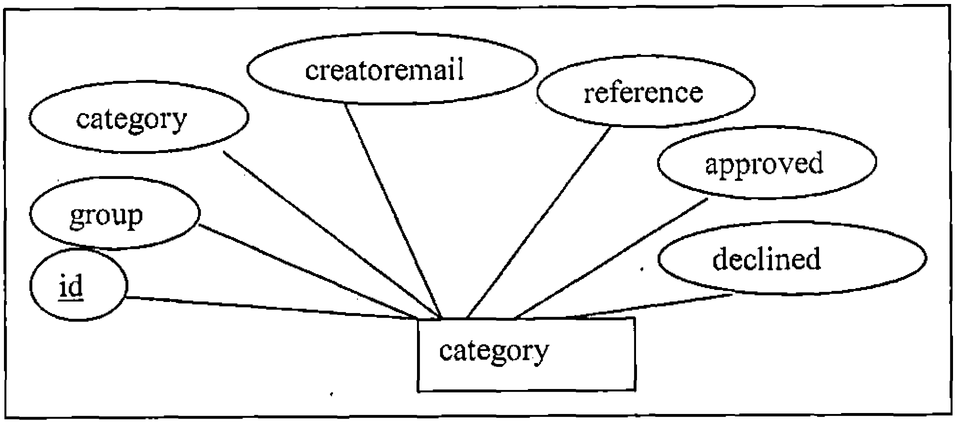


Figure 6. Category Entity and Attributes

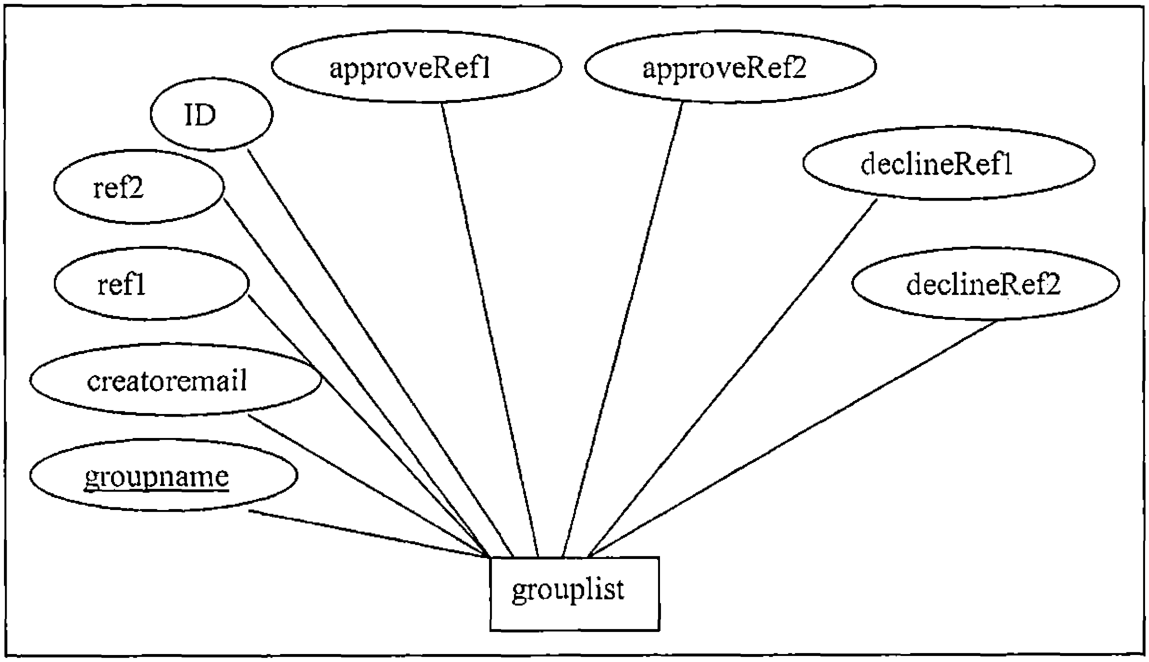


Figure 7. Grouplist Entity and Attributes

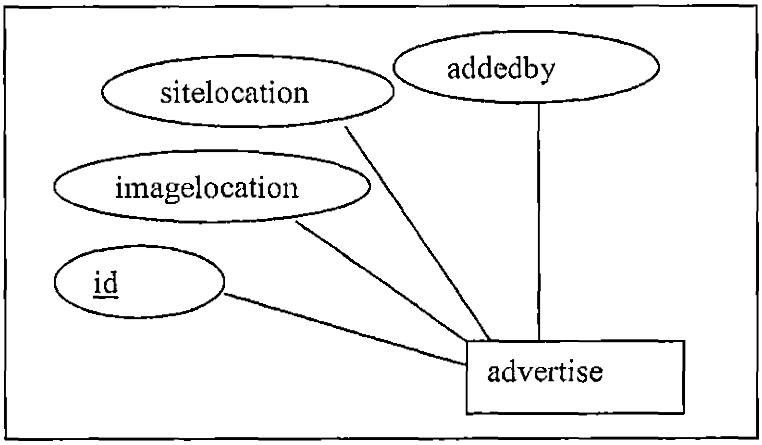


Figure 8. Advertise Entity and Attributes

In the ER diagram the rectangles represents the entity sets, the ellipses represents the attributes belonging to entity sets and the diamonds represent the relationship sets.

"Adds relationship set" is defined as one to many from user to Advertise entity set because a user may have to advertise many times but advertise should belong to one user. "Member of relationship set" is defined as one to many from group to user because a group may have many users registered but user should belong to only one group. "Owner relationship set" is defined as many to one from item to user entity set because one user may be the owner of many items and an item should be owner by one user. Similarly, "uses relationship set" is defined as many to one from item to user entity set because one user may be using many items and an item could be used by only one user. "Has relationship set" is defined as one to many from group to category because a group will be having many categories but a category should belong to one group. "Belong to relationship set" is defined as one to many from category to item entity set because a category may have many items and an item should belong to a category.

## 3.3 Database Schema Logical Model

The conceptual model ER diagram maps into the following relational table design. In the following tables(1 -5), underlined fields indicate the primary key.

Table 1. Structure of Table Grouplist

| Field | Type | Description |
|---|---|---|
| Groupname | varchar(150) | Name of the group |
| creatoremail | varchar(50) | Email of group creator |
| ref1 | varchar(100) | Name of first reference requested for creating group |
| ref2 | varchar(100) | Name of second reference who was requested for creating the group |
| ID | int(10) | Auto incremented id |
| approveRef1 | tinyint(4) | 1 - first reference approved request<br>0 - default |
| approveRef2 | tinyint(4) | 1 - second reference approved request<br>0 - default |
| declineRef1 | tinyint(4) | 1 - first reference declined request<br>0 - default |
| declineRef2 | tinyint(4) | 1 - first reference approved request<br>0 - default |

Table 2. Structure of Table Item

| Field | Type | Description |
|---|---|---|
| group | varchar(50) | Group item belongs |
| category | varchar(50) | Category to which item belongs. |
| description | text | Detailed description of the item |
| owner | varchar(50) | Owner of the item |
| useremail | varchar(50) | Email of item user |
| startdate | date | Date when the current user started using the item |
| enddate | date | Date the item will be displayed while searching. |
| id | int(10) | Auto incremented id |
| itemname | varchar(50) | Name of the item to be searched for. |
| maxdays | int(10) | Maximum Days an item could be kept by a user, set by owner. |
| needdays | int(10) | Needed days as specified by  user who gets item |
| nextuser | varchar(50) | Email of requested user of item |
| nextuserneeddays | int(10) | Needed days as specified by user who request item |

Table 3. Structure of Table User

| Field | Type | Description |
|---|---|---|
| ID | varchar(50) | Auto incremented id |
| firstname | varchar(50) | First Name of User |
| lastname | varchar(50) | Last Name of User |
| street | varchar(50) | Street of User |
| city | varchar(50) | City of User |
| state | varchar(50) | State of USer |
| zipcode | varchar(15) | ZipCode of User |
| phone | varchar(15) | Contact User Phone number |
| email | varchar(50) | Unique Contact email of User |
| reference | varchar(100) | Name of the reference to whom request sent to join the group. |
| group | varchar(150) | Group the user wishes to join |
| date | date | Join Date of user |
| password | varchar(50) | Password for user to login |
| approved | tinyint(4) | 1 - reference approved request<br>0 - default |
| declined | tinyint(4) | 1 - reference declined request<br>0 - default |
| superuser | tinyint(1) | 1 - Have Admin Privileges<br>0 - default |
| restricted | tinyint(1) | 1 - user denied access by Admin<br>0 - default |

Table 4. Structure of Table Category

| Field | Type | Description |
|---|---|---|
| id | int(50) | Auto incremented id |
| group | varchar(100) | Group which the category belongs |
| category | varchar(100) | Category name added |
| creatoremail | varchar(100) | category creator email |
| reference | varchar(100) | Name of the user whose reference requested for creating category |
| approved | tinyint(4) | 1 - reference approved request<br>0 - default |
| declined | tinyint(4) | 1 - reference declined request<br>0 - default |

Table 5. Structure of Table Advertise

| Field | Type | Description |
|---|---|---|
| id | int (10) | Auto incremented id |
| Imagelocation | text | Image Location |
| Sitelocation | text | The site location which should be opened in new tab onClick of image |
| addedby | text | The email of the user who added this advertise fields |

# CHAPTER FOUR

# PROJECT IMPLEMENTATION

## 4.1 Introduction

The project is implemented using the MVC pattern. This chapter includes the explanation of the underlying action that takes place from each screen shot. Every page has an included content where he can search for any item, and a HELP link for accessing the FAQ's. If the user is logged in then, there is a logout link, for the user to logout from any page. There is always a link to the main option page which connects them to the page where the user can select the different options. For security considerations, the system checks for the cookies before loading any of the pages. This makes sure that only a logged-in person can view the pages allowed for them. Figure 9 shows the use case diagram for the system. Figure 10 shows the package diagram for the system. Figure 11 shows the services sub package diagram used for consuming the Web Services. Figure 12-17 shows the class diagrams of important classes.

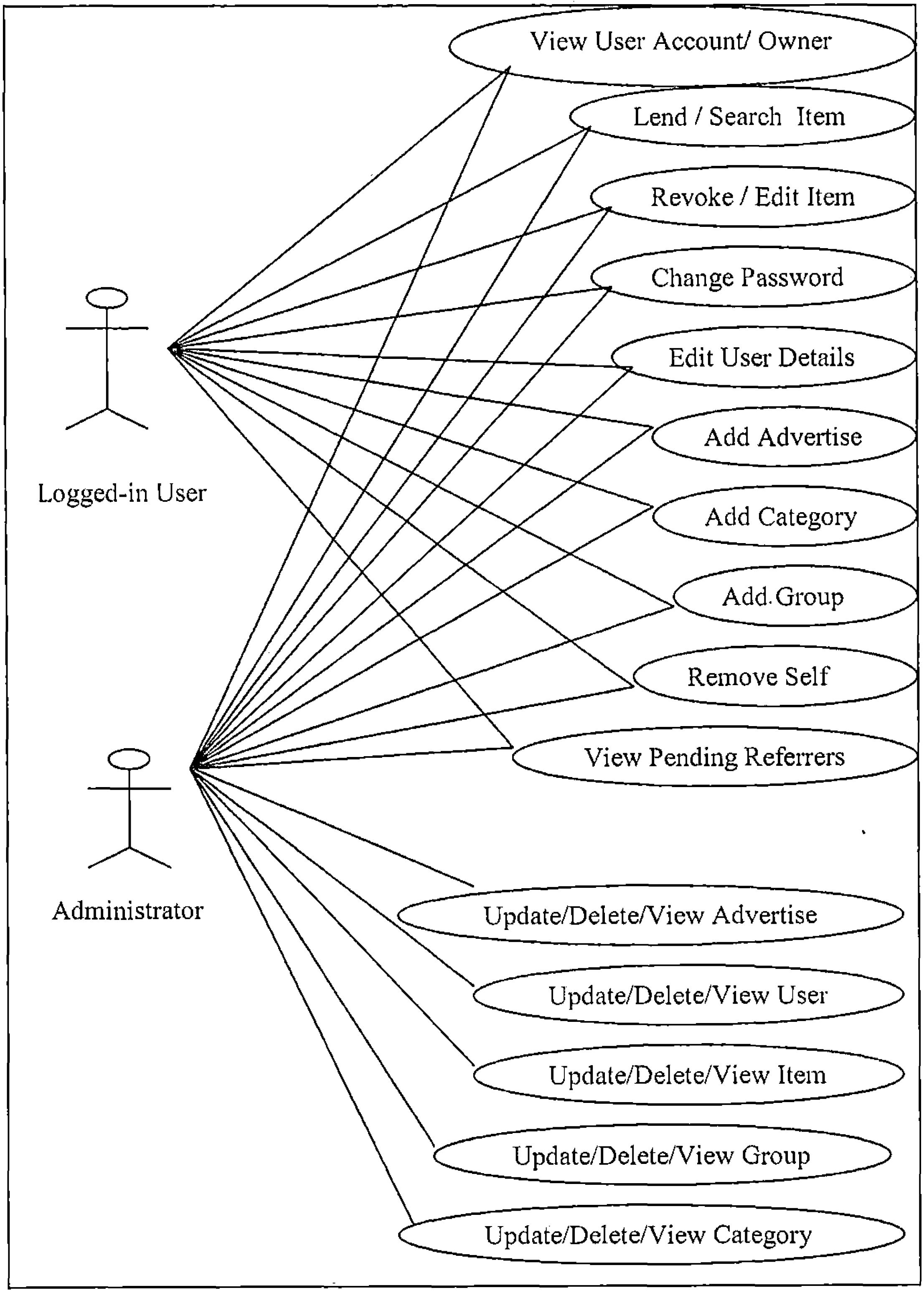


Figure 9. Use Case Diagram

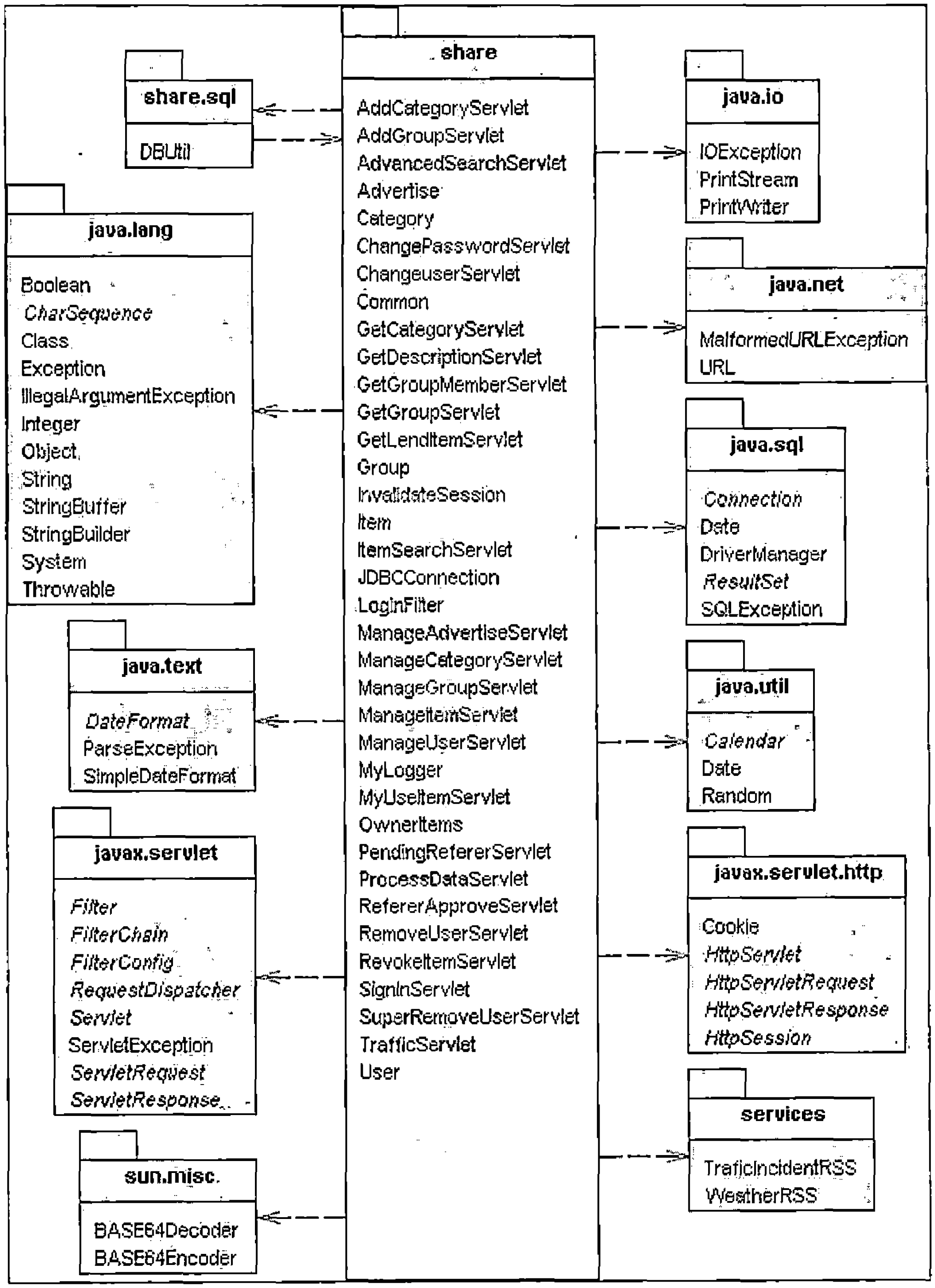


Figure 10. Package Diagram

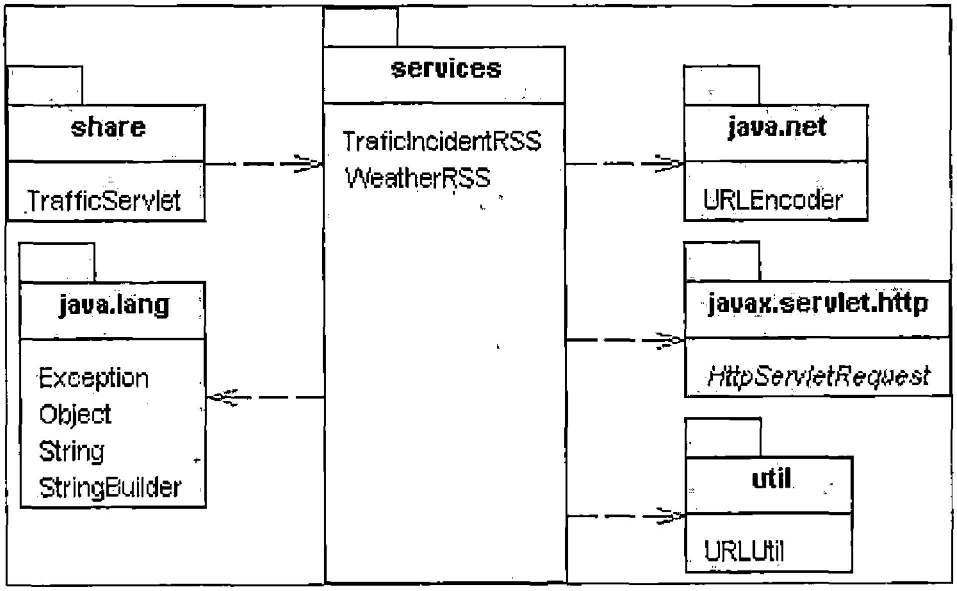


Figure 11. Services Sub-Package Diagram

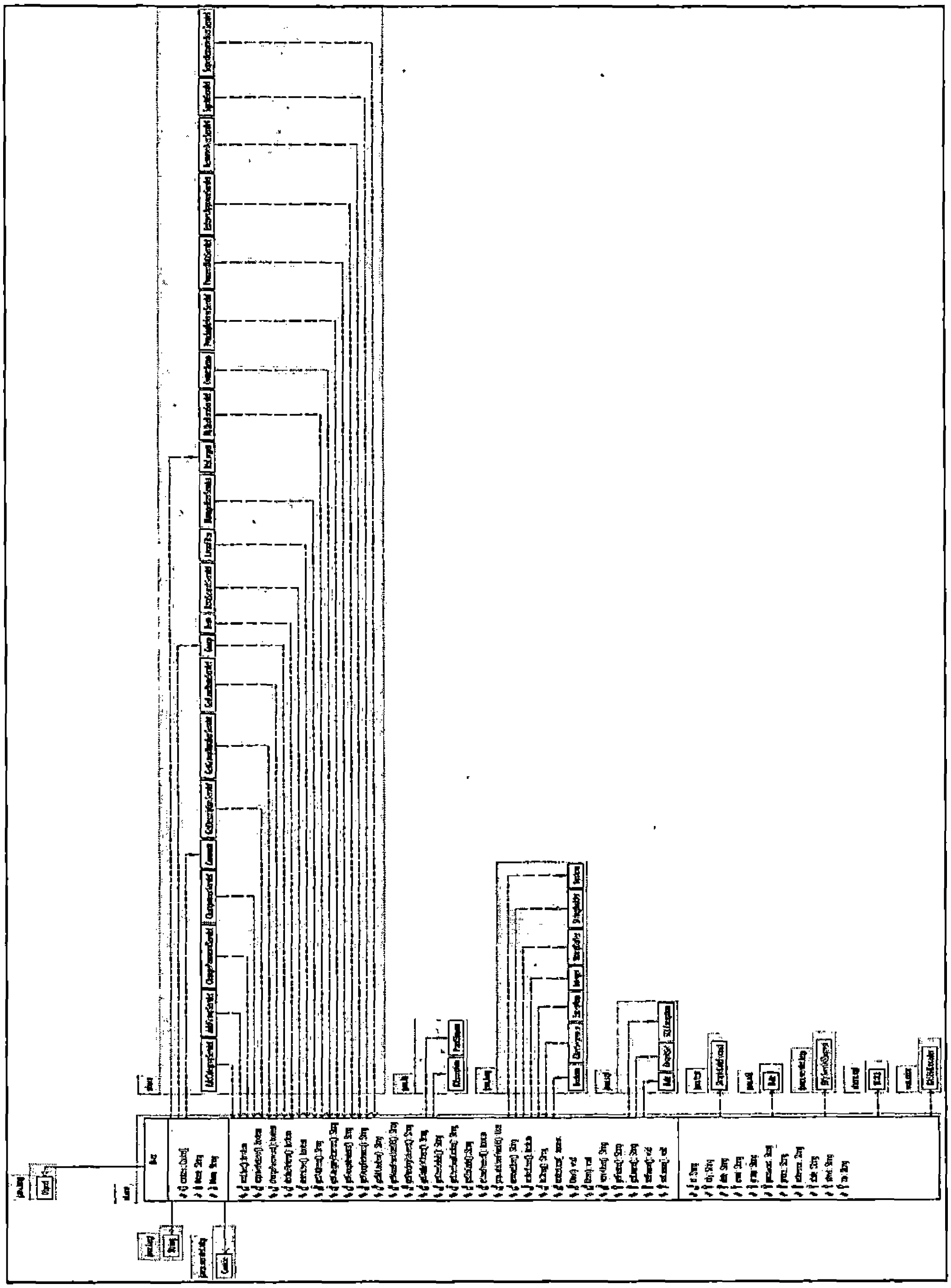


Figure 12. User Class Diagram

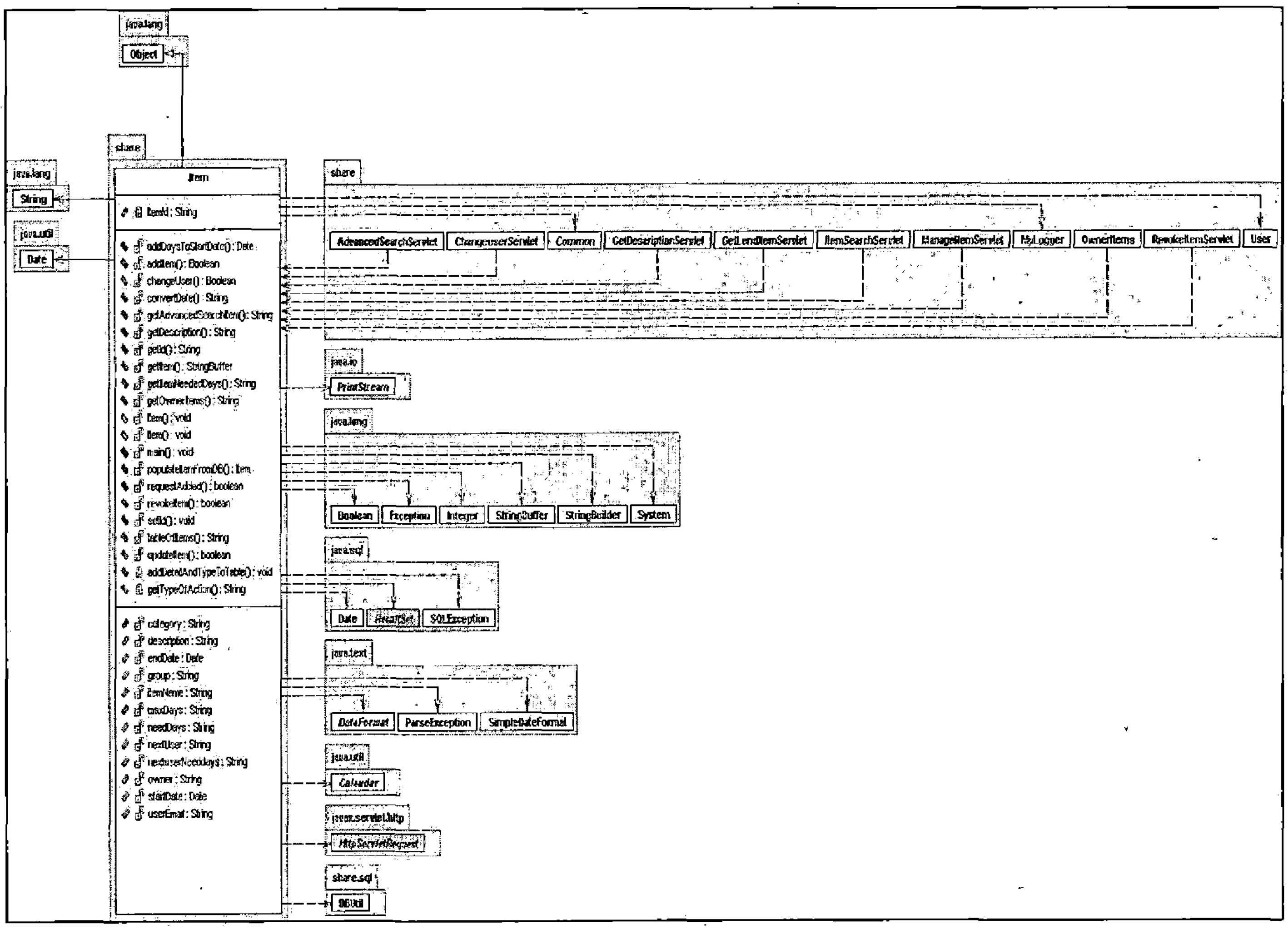


Figure 13. Item Class Diagram

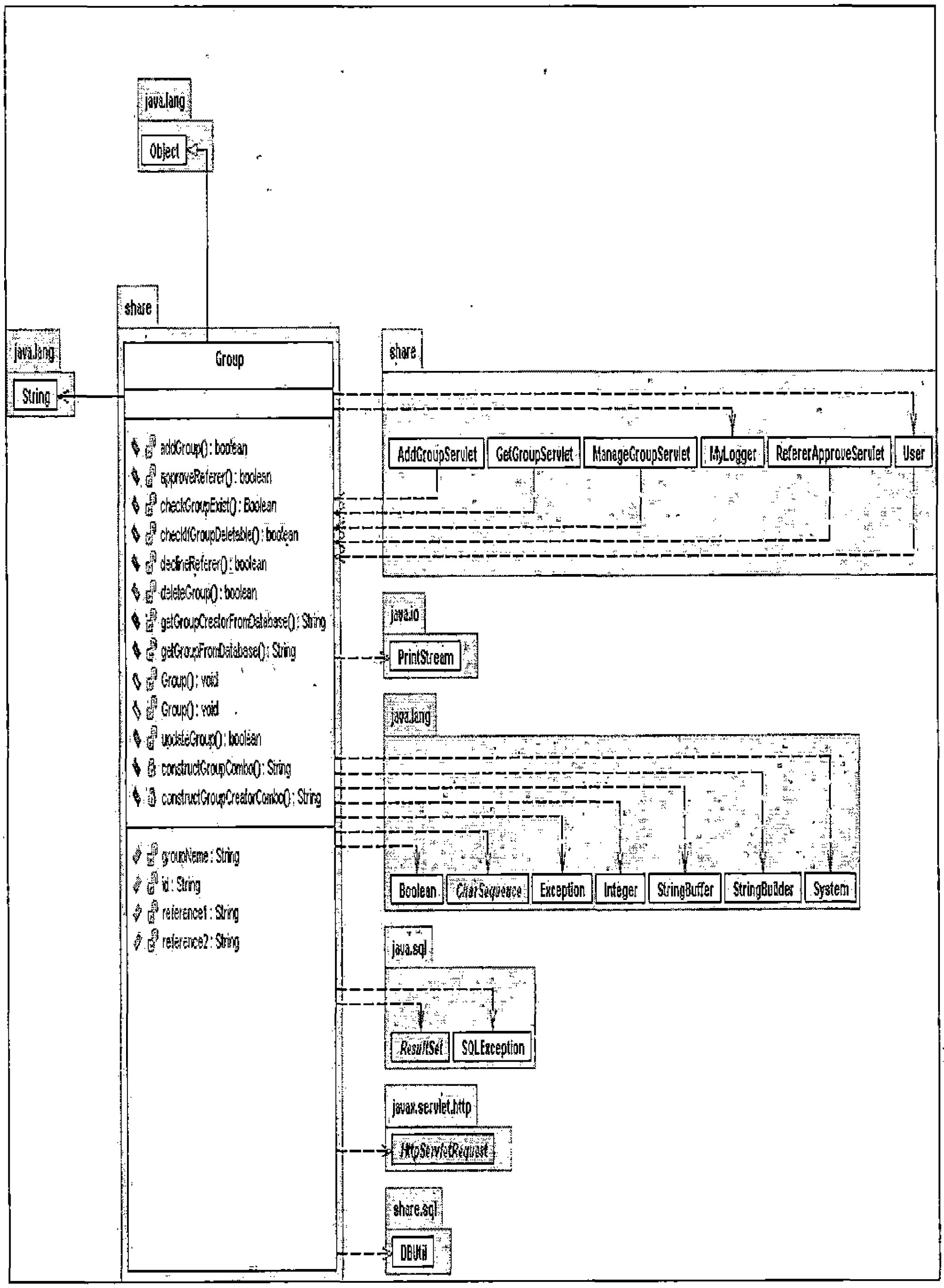


Figure 14. Group Class Diagram

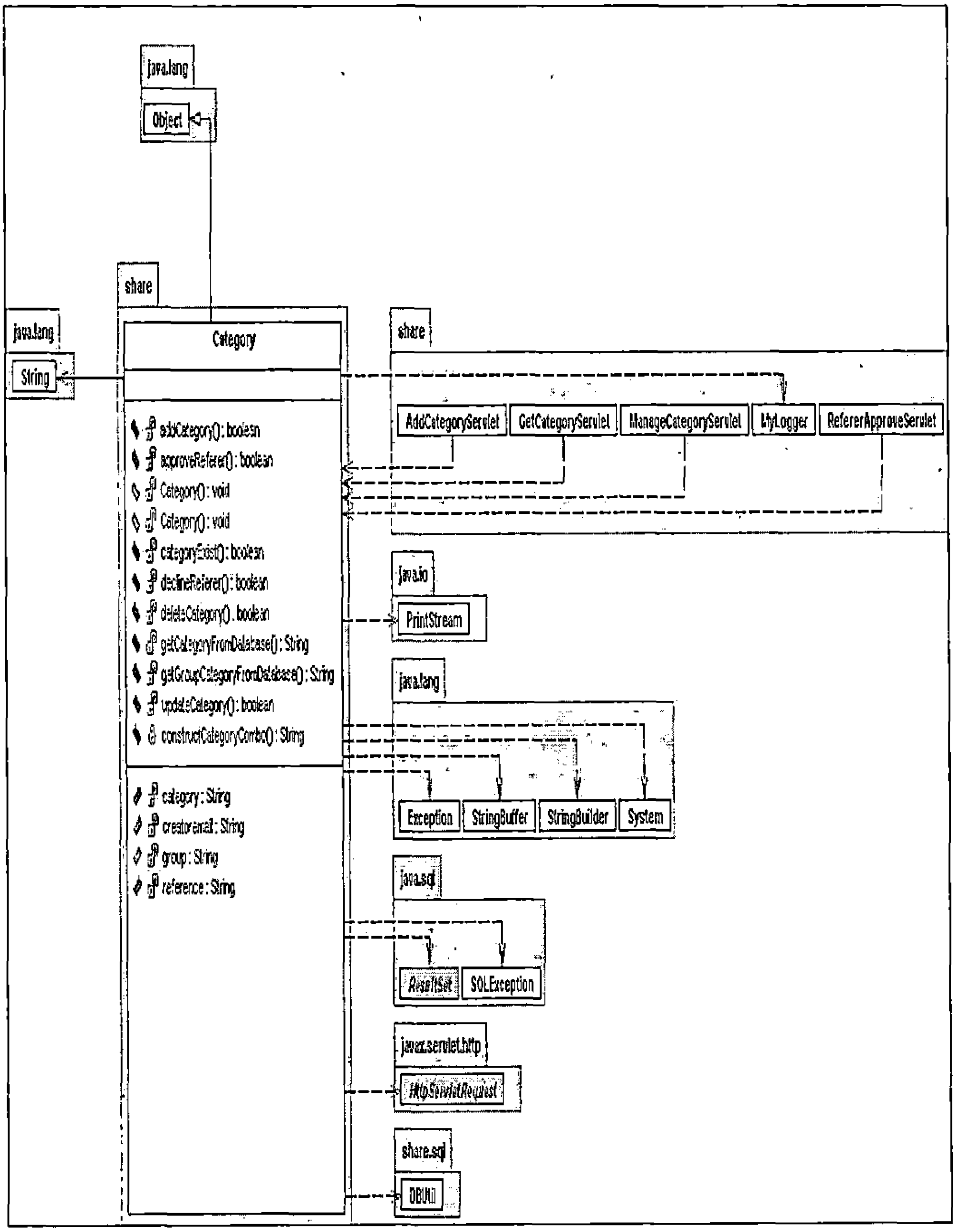


Figure 15. Category Class Diagram

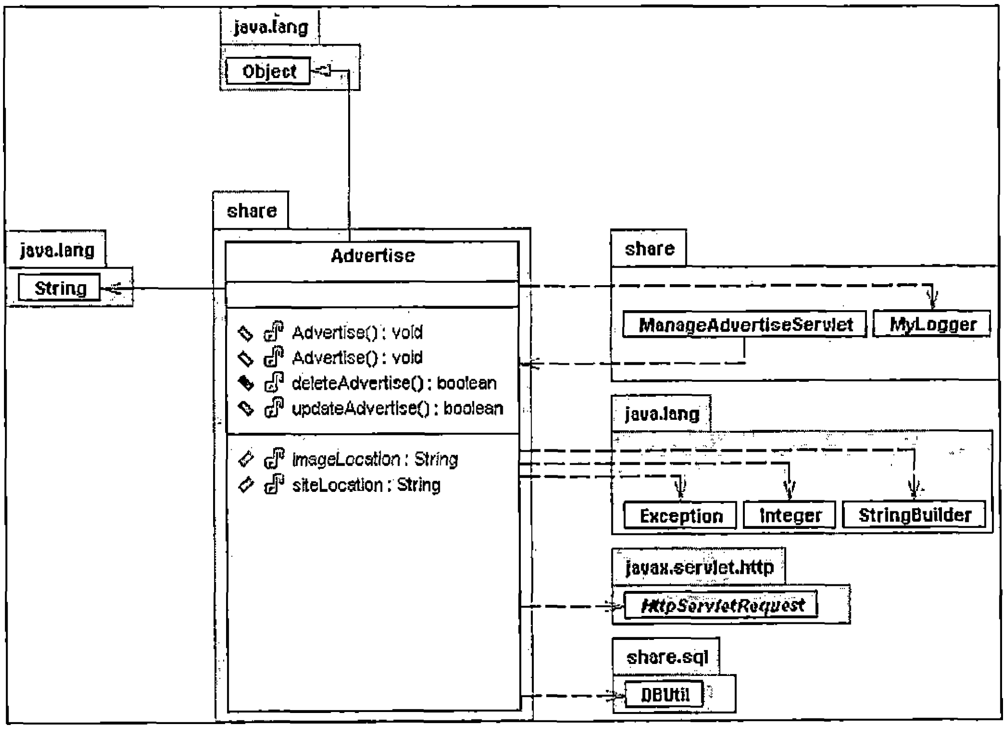


Figure 16. Advertise Class Diagram

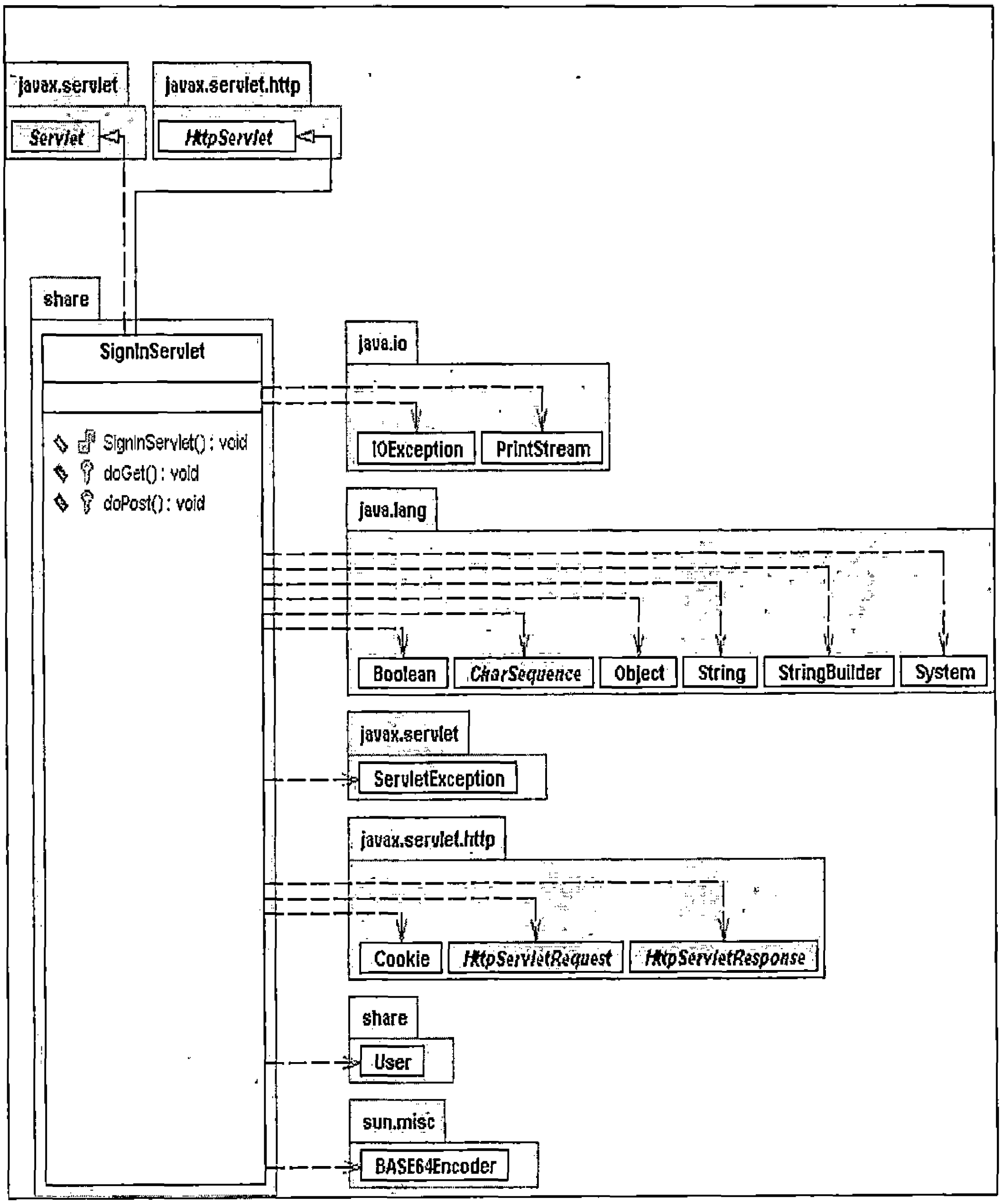


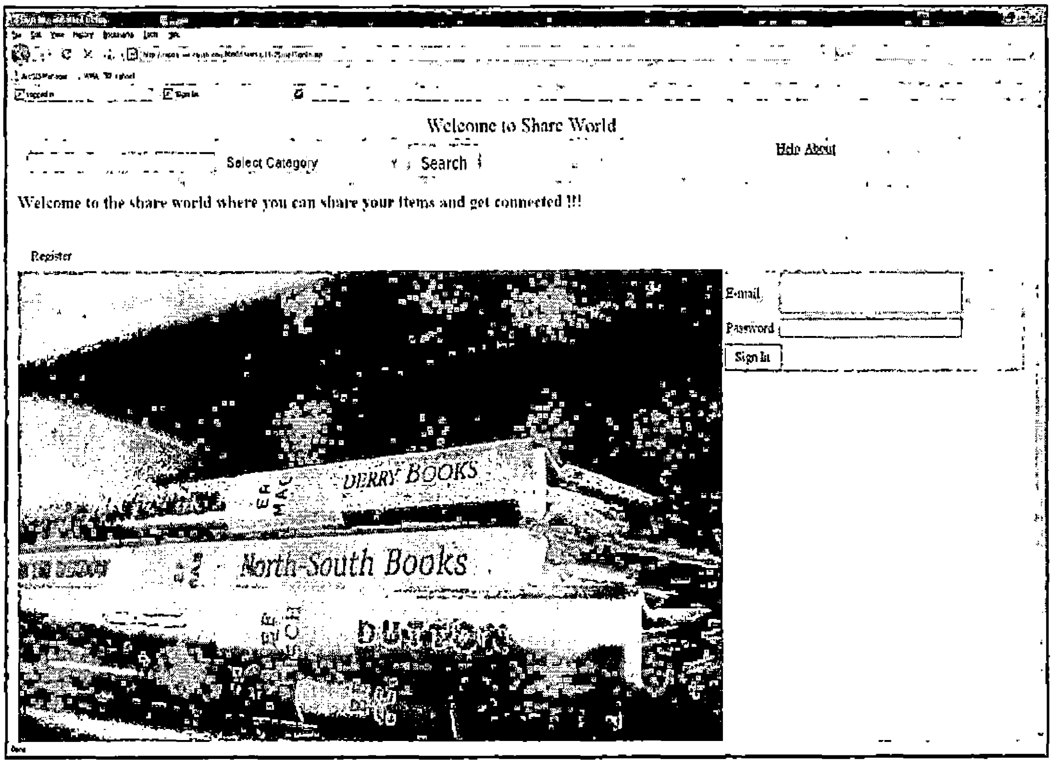
Figure 17. SignInServlet Class Diagram

### 4.2. WRMS Graphical User Interface Design

WRMS GUI is easy to use. The GUI is written using Hyper Text Markup Language (HTML). All the functions that the logged-in user has are placed in the menu part. Client-side validations are done using the scripting language, JavaScript. Hence, the WRMS GUI is executable with browsers that support JavaScript. The combo boxes are dynamically filled in from the database by sending an Ajax request to the server. The following sub sections explain the GUI work and details.

### 4.2.1 Home Page

This page will be the first page that all the users will see when they enter WRMS. The page has the HELP and ABOUT links. The HELP link has the FAQ's for the user. This page provides the fields for the registered users to login. The password entered by the user is message digested. For non-registered users, they can use the register button to register, which directs them to a form where they enter the user details. They can also search for the items by selecting the category of the items. The categories existing are dynamically loaded in a combo box from the category table of the database by sending Ajax request. If the user has no much idea of the item to be searched, he/she can leave the field blank, which displays all the items in the specified category. When the user signs-in, the system will check for a matching user-email and password. The system checks by checking the email and password validity. If valid the system will check for the approved, restricted, super user flags. If approved is true and restricted false then the user is directed to the logged-in page. Otherwise appropriate messages are conveyed from the server side for the user information. If super user flag is true, it gives access to restrict user page which is displayed only for the Admin. This helps the Admin to have control over the users.

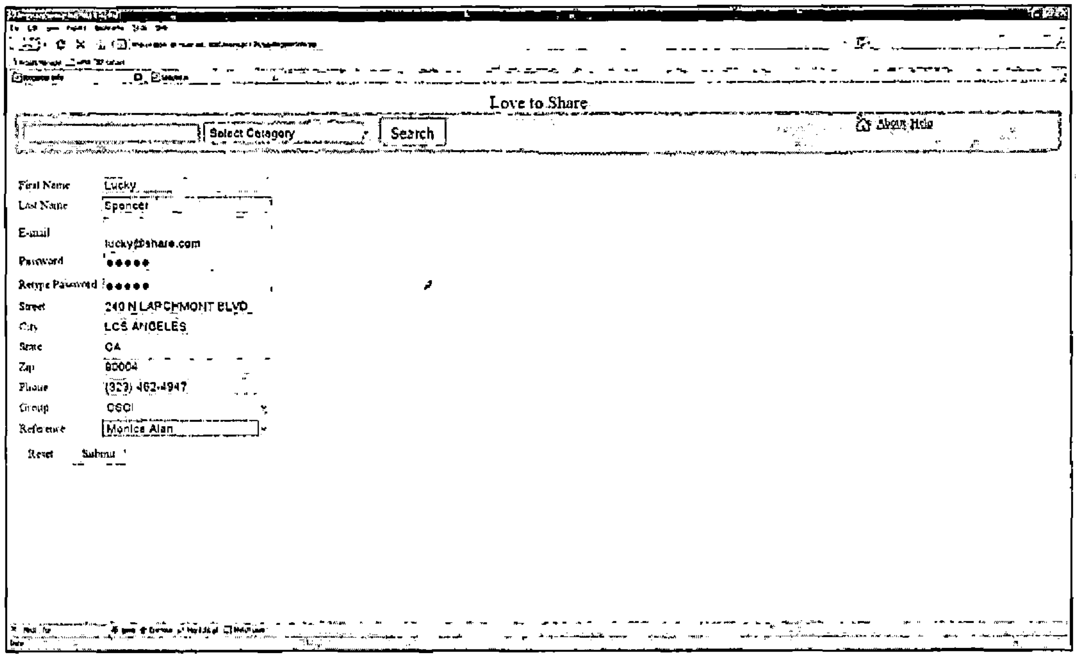

Welcome to Share World

Welcome to the share world where you can share your items and get connected !!!

Figure 18. Home Page

### 4.2.2 Register User Page

The user can enter the requested information through the form. A group combo box is dynamically filled from the group table of the database on load of the page. When the user selects any group, he can view the group members, which he can select for requesting the approval from the reference. The group members are filled in the combo box by the Ajax request. Upon submitting, the system checks whether the user-email already exists. If the mail-id does not exist, the register information is inserted to the database.



| | |
|---|---|
| First Name | Lucky |
| Last Name | Spencer |
| E-mail | lucky@share.com |
| Password | ●●●●● |
| Retype Password | ●●●●● |
| Street | 240 N LARCHMONT BLVD |
| City | LOS ANGELES |
| State | CA |
| Zip | 90004 |
| Phone | (323) 462-4947 |
| Group | CSCI |
| Reference | Monica Alan |

Figure 19. Registration Page

4.2.3 Logged-In Page

When a user can sign in, the Servlets create a cookie and is send to the browser. The cookie will be deleted either by exiting the browser or by clicking the logout link. The logged-in page is accessible only to the signed in user. This is possible by the use of Servlets filters. The filter filters the request that has the cookie. The logged-in user has different menus based on the privileges granted to the user.

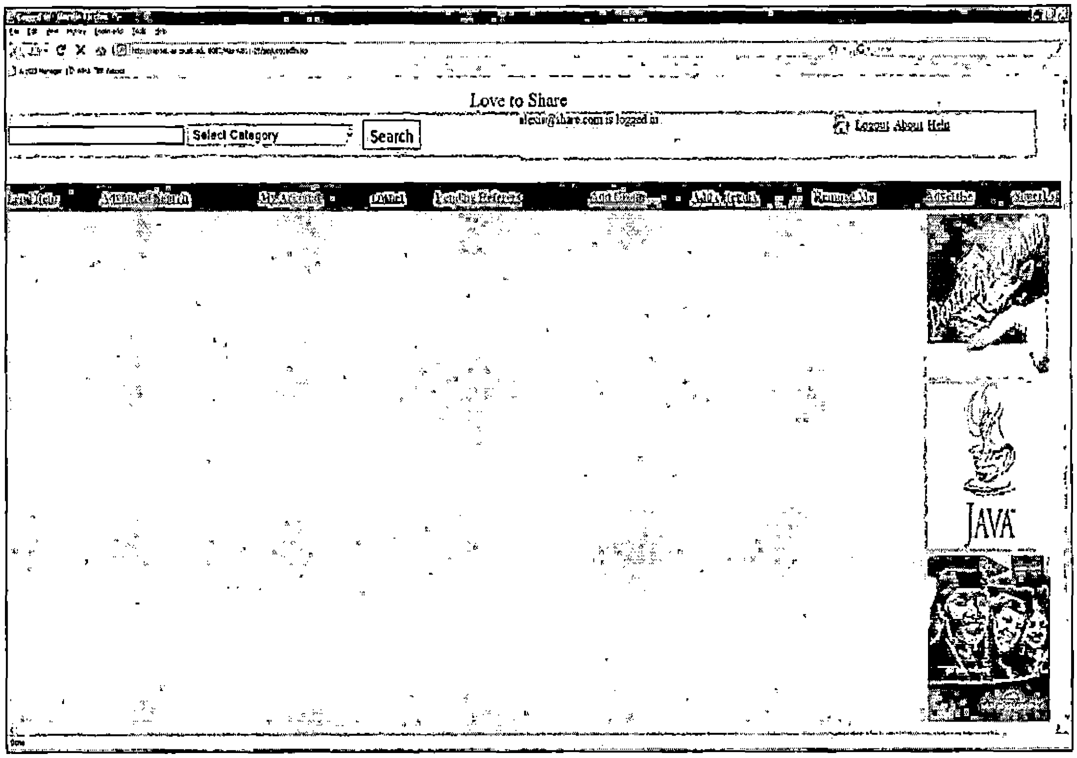
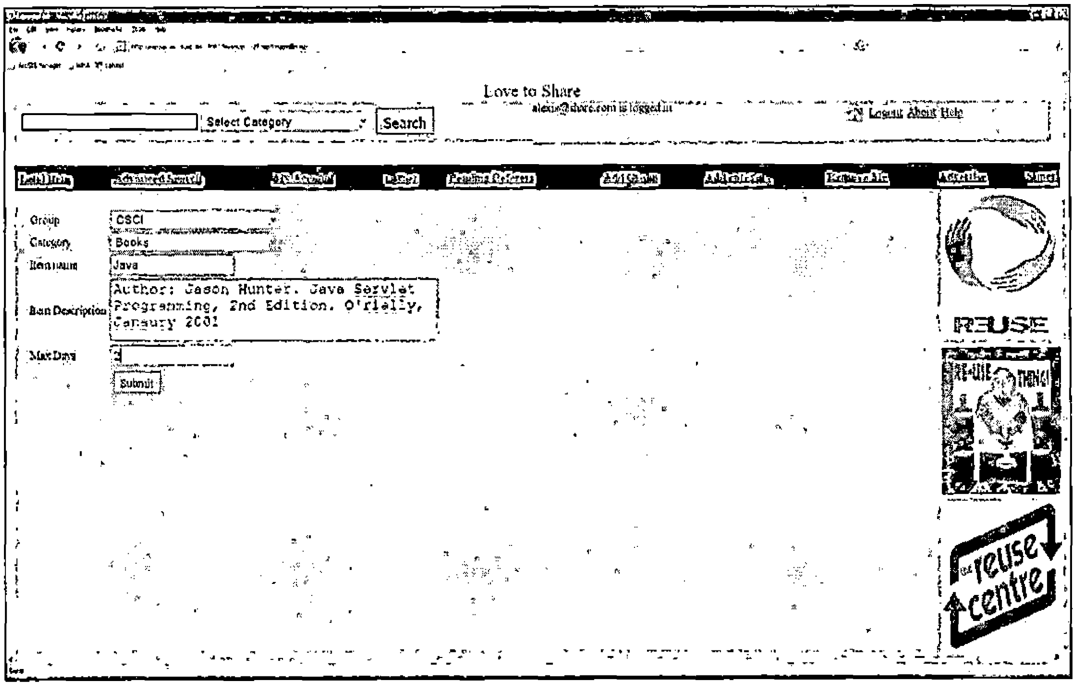


Figure 20. Logged-In Page

## 4.2.4 Lend Item Page

The user can post their item here. The Client-side validations are done using JavaScript by checking the document fields for null or empty strings. The maximum days any user can use the item should be a minimum of 2 days and maximum of 120 days, ie, the user should be able to use the item for at least 2 days and could be used for only 120 days. The request is send to the server, which inserts the item and its related details to the database. Once the item is inserted, the server sends a message back to the client.



alexis@share.com is logged in

Group: CSCI

Category: Books

Item name: Java

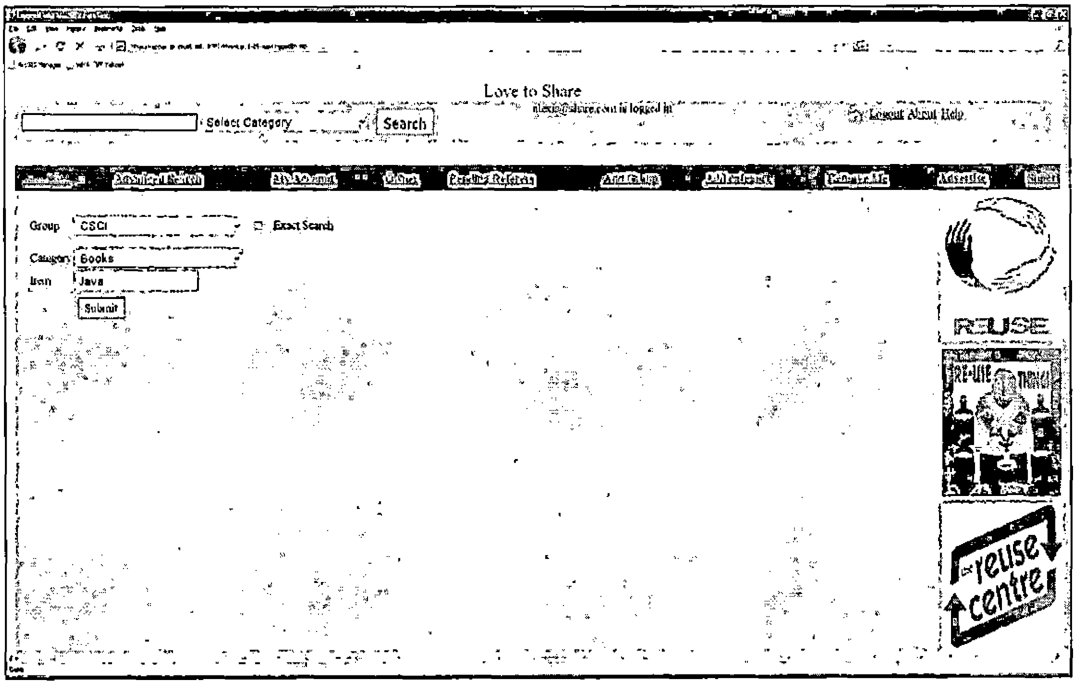
Item Description: Author: Jason Hunter. Java Servlet Programming, 2nd Edition. O'rielly, January 2001

Max Days: 2

Figure 21. Lend Item Page

### 4.2.5 Search Page

The user has different options to refine his search within a group and category. The group combo is filled in dynamically by sending an "onLoad Ajax request" to the server.



Group CSCI

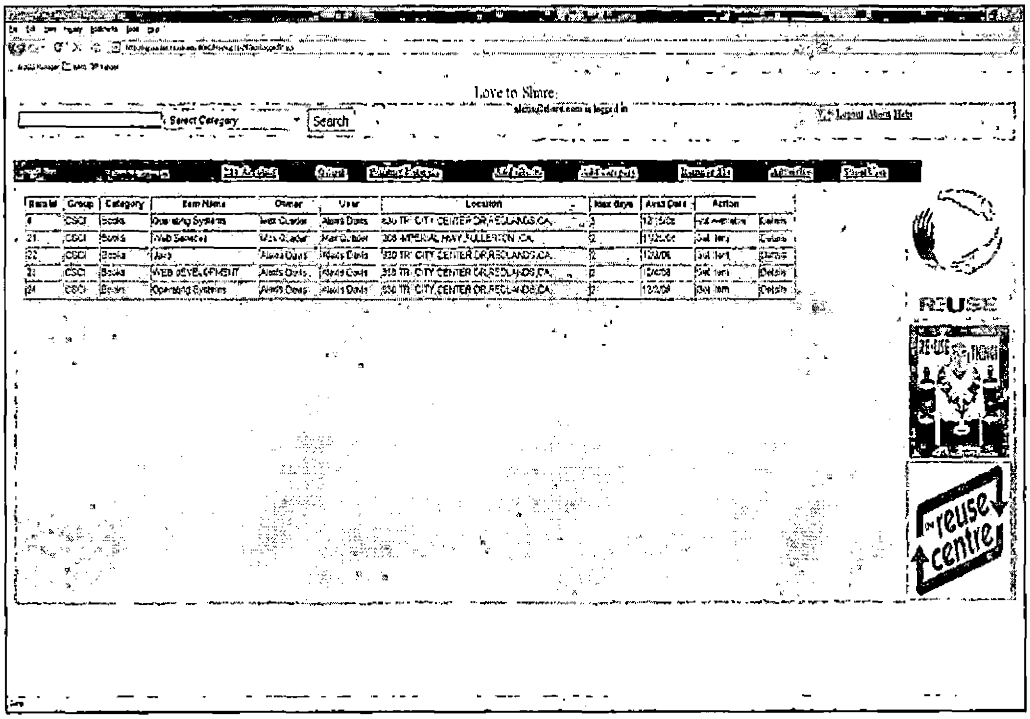
□ Exact Search

Category Books

Item Java

Figure 22. Search Request Page

Upon selecting the group, an Ajax request is send to the server, which fills in the categories that are approved under the specified group. Some of the features available for the search are able to carry out an exact search of the item, find all the items that contain the search name, or simply display all the items in the selected group and category. The user gets to see the details of the user, the availability dates and various other information that are relevant. The system compares the current date with the end date of the current user. If the current date is more than the user's end date and there is no next user, the user can get the item. If the current date is less than the user's end date, then the logged-in user can only request the item. If the searched item is currently being used and another user has already requested for the item, then the item appears as "not available". The logged-in user cannot get or request for the item.

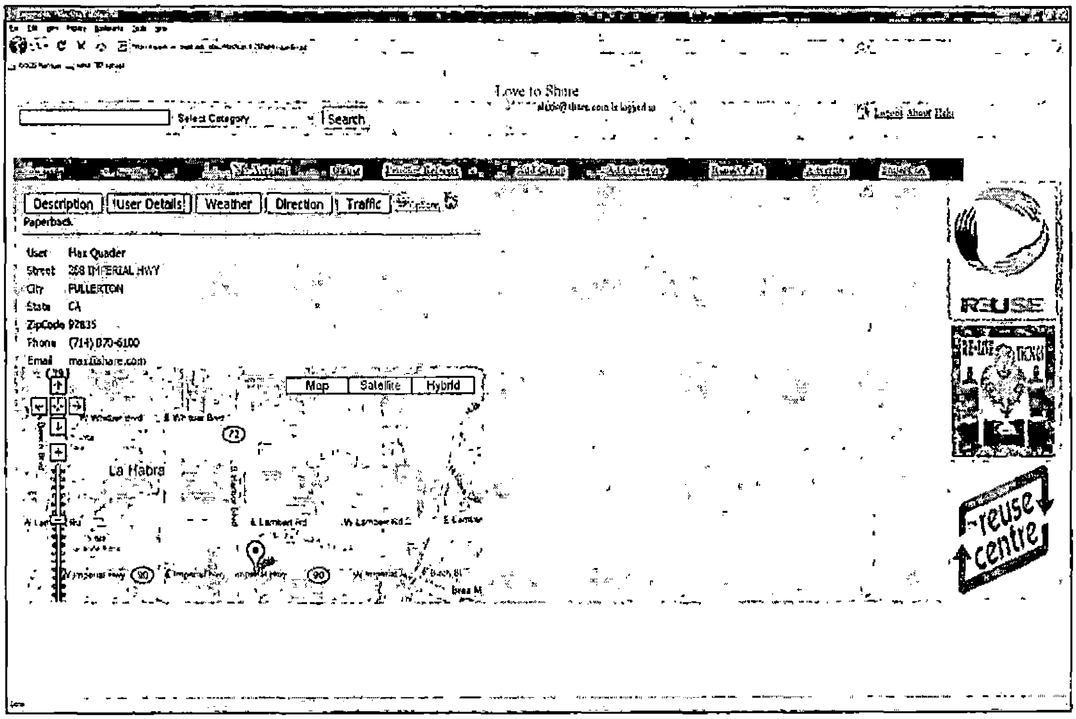


| Rental Id | Group | Category | Item Name | Owner | User | Location | Max days | Avail Date | Action | |
|---|---|---|---|---|---|---|---|---|---|---|
| 4 | CSCI | Books | Operating Systems | Max Gulder | Alexis Davis | 330 TRI CITY CENTER DR,REDLANDS,CA | 3 | 12/5/08 | Not Available | Details |
| 21 | CSCI | Books | Web Services | Max Gulder | Max Gulder | 308 IMPERIAL HWY,FULLERTON,CA | 2 | 11/25/08 | Get Item | Details |
| 22 | CSCI | Books | Java | Alexis Davis | Alexis Davis | 330 TRI CITY CENTER DR,REDLANDS,CA | 2 | 12/3/08 | Get Item | Details |
| 23 | CSCI | Books | WEB DEVELOPMENT | Alexis Davis | Alexis Davis | 330 TRI CITY CENTER DR,REDLANDS,CA | 2 | 12/4/08 | Get Item | Details |
| 24 | CSCI | Books | Operating Systems | Alexis Davis | Alexis Davis | 330 TRI CITY CENTER DR,REDLANDS,CA | 2 | 12/2/08 | Get Item | Details |

Figure 23. Search Response Page

### 4.2.6 Details Page

This page geocodes the current item location and displays the Google maps [11]. On click of the description button, the logged-in user can view the description of the item as entered by the owner of the item. On double click the description can be hidden. Similarly, the logged-in member can view and hide the details of the user of the item.

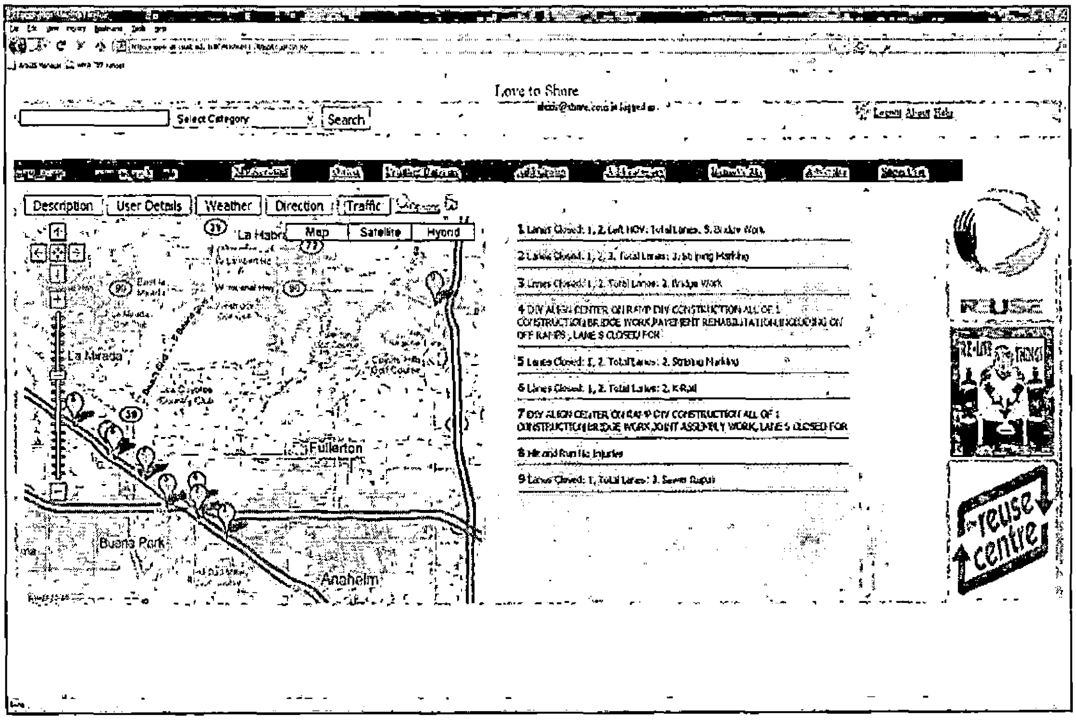


Paperback

User Max Quader
Street 268 IMPERIAL HWY
City FULLERTON
State CA
ZipCode 92835
Phone (714) 870-6100
Email max@share.com

Figure 24. User Details Response Page

The logged-in member can view any live traffic incidents overlayed on the Google Maps as well as the description for each of the traffic incidents. These traffic incidents data are obtained by consuming Web Services from the Yahoo Traffic RESTful Service [1]. The client sends an Ajax request for the traffic incident data. Since the client cannot connect to any other server other than the originating server because of the client-side restrictions, the client request is processed by the Servlets. The Servlets connect to the Yahoo server by sending a REST request to the server. The connection to the Yahoo server could be obtained by opening an URLConnection [6] to the server. The Yahoo server responds back by sending an XML response back. This XML [3] response is send to the client. The XML response is parsed and the elements are obtained. The XML response has the elements of description, title, latitude, longitude. The locations of the traffic incidents are geocoded with markers using the latitude and longitudes. On click of the markers, an info window will be opened, giving a brief description about the geocoded location. A more detailed printed description of the incidents could be viewed on the screen. The traffic incidents can be hidden on double click of the traffic button.



1 Lanes Closed: 1, 2, Left HOV; Total Lanes: 5, Bridge Work

2 Lanes Closed: 1, 2, 3, Total Lanes: 3, Striping Marking

3 Lanes Closed: 1, 2, Total Lanes: 2, Bridge Work

4 DIV ALIGN CENTER ON RAMP DIV CONSTRUCTION ALL OF 1 CONSTRUCTION BRIDGE WORK,PAVEMENT REHABILITATION,INCLUDING ON OFF RAMPS , LANE S CLOSED FOR

5 Lanes Closed: 1, 2, Total Lanes: 2, Striping Marking

6 Lanes Closed: 1, 2, Total Lanes: 2, K-Rail

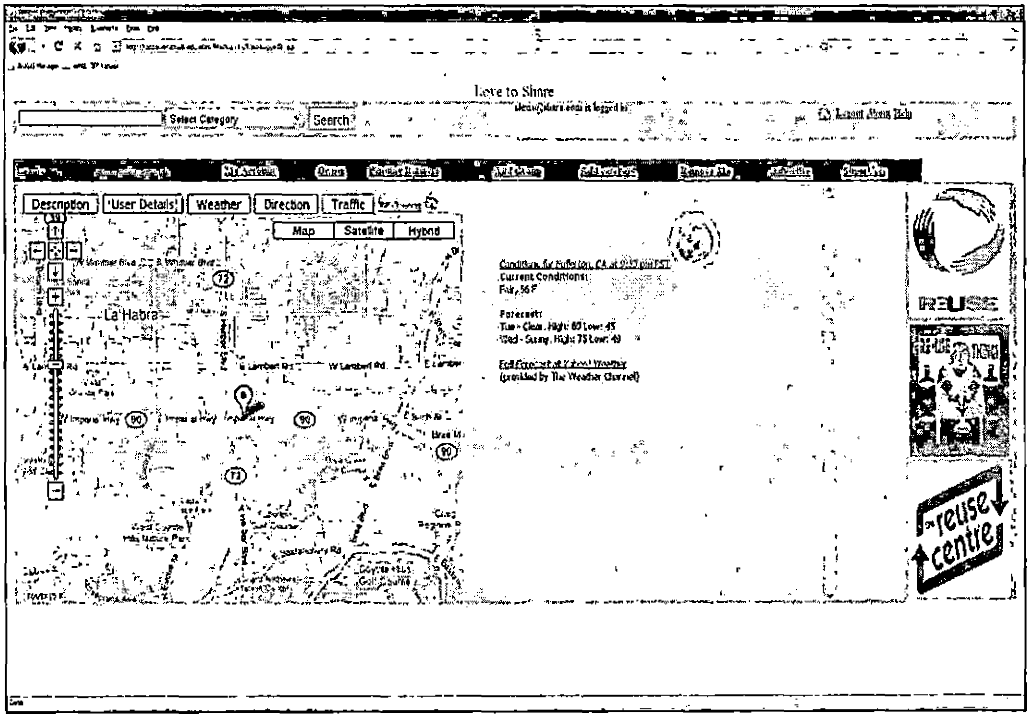
7 DIV ALIGN CENTER ON RAMP DIV CONSTRUCTION ALL OF 1 CONSTRUCTION BRIDGE WORK,JOINT ASSEMBLY WORK, LANE S CLOSED FOR

8 Hit and Run No Injuries

9 Lanes Closed: 1, Total Lanes: 3, Sewer Repair

Figure 25. Traffic Response Page

The system also consumes Yahoo Weather Service [1]. The logged-in user can get the weather information of the current item location. The request for weather is sent in a similar way to the Yahoo server. The Yahoo server gives the response back as an RSS feed. On double click the weather display could be hidden.



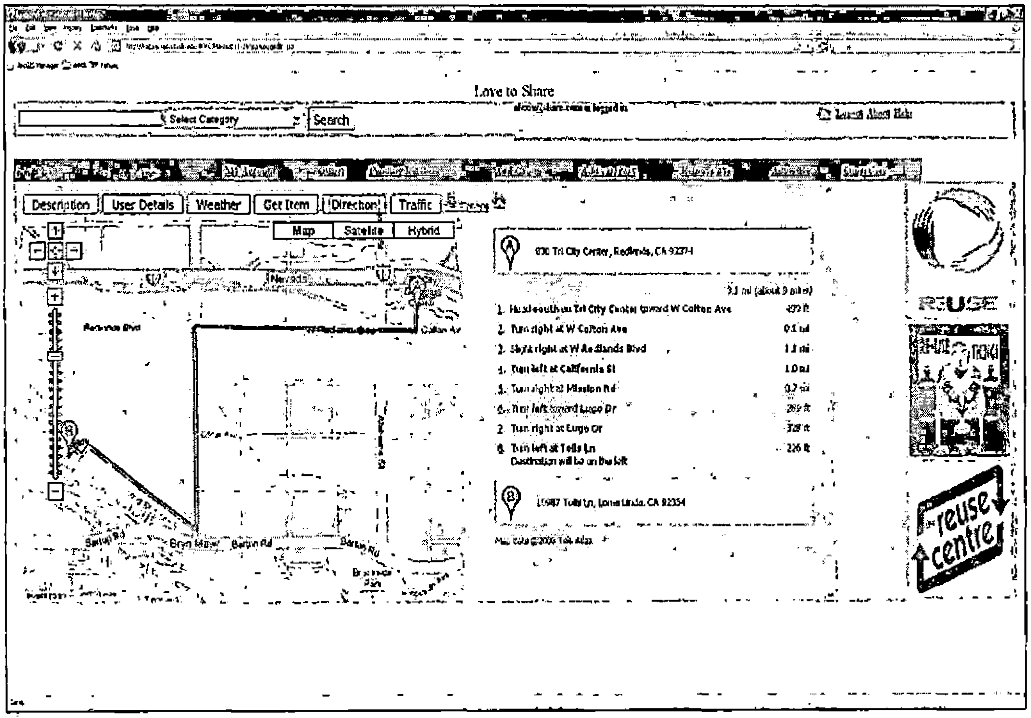
Conditions for Fullerton, CA at 9:57 pm PST
Current Conditions:
Fair, 56 F

Forecast:
Tue - Clear. High: 69 Low: 45
Wed - Sunny. High: 75 Low: 49

Full Forecast at Yahoo! Weather
(provided by The Weather Channel)

Figure 26. Weather Response Page

The logged-in member can view the driving direction from the logged-in user to the item-user location. The directions will be overlayed on the Google Maps [11]. Also a detailed printed description of the directions will also be displayed. On double click the display of the directions could be hidden.



630 Tri City Center, Redlands, CA 92374

9.1 mi (about 9 mins)

1. Head south on Tri City Center toward W Colton Ave 492 ft
2. Turn right at W Colton Ave 0.1 mi
3. Slight right at W Redlands Blvd 1.3 mi
4. Turn left at California St 1.0 mi
5. Turn right at Mission Rd 0.2 mi
6. Turn left toward Lugo Dr 289 ft
7. Turn right at Lugo Dr 328 ft
8. Turn left at Tolla Ln 226 ft
Destination will be on the left

10987 Tolla Ln, Loma Linda, CA 92354

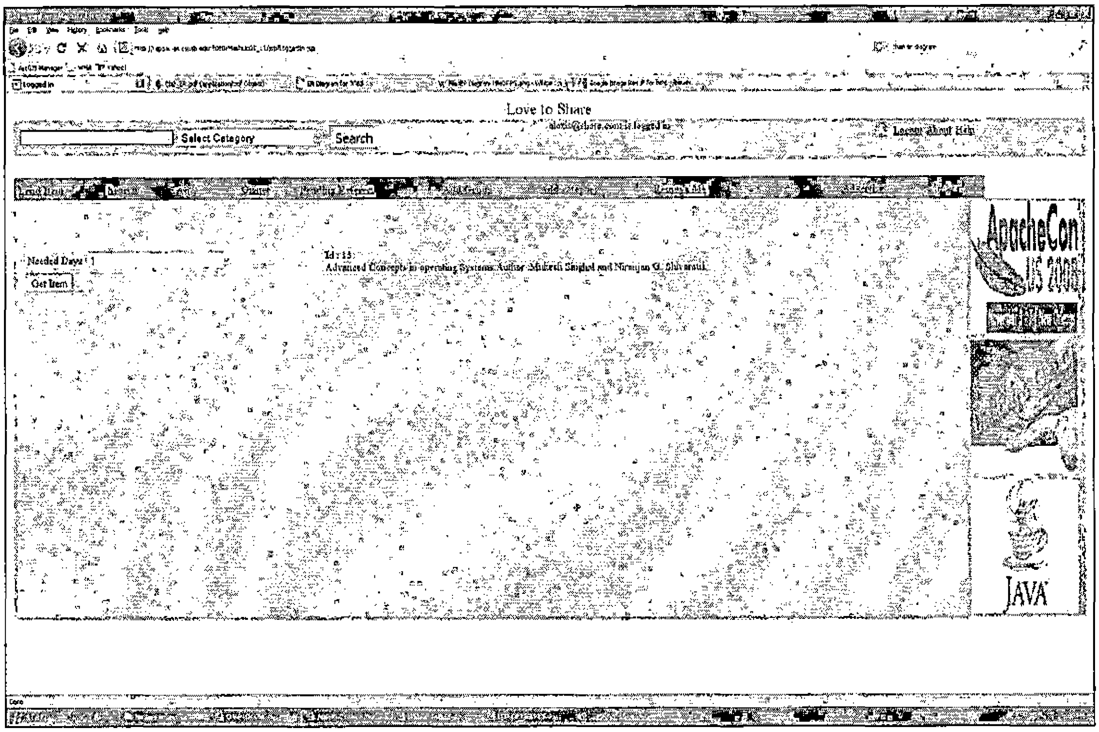
Map data ©2006 Tele Atlas

Figure 27. Directions Response Page

If the status of the item is "get item", then the "get item" button will be displayed. Otherwise, if it is "request item", then request item button will be displayed. If the status is "not available", then nothing can be done regarding the item. So no button will be displayed.

4.2.7 Needed Days Page

The user can access this page, if the status of the item is either "get item" or "request item". Here the user has to specify the needed days of the item. This is done by selecting the dynamic combo box which contains values up to the maximum allowed days for the item. On submit, if the status is "get item" the user of the item will be changed to the logged-in user else if the status is "request item" the logged-in user will be added as the next user in the database. This makes the logged-in user as the next-user. The logged-in user will be informed upon completion of the action. This makes the item to appear as "not available" on further searches for the item by other members.



alerts@share.com is logged in

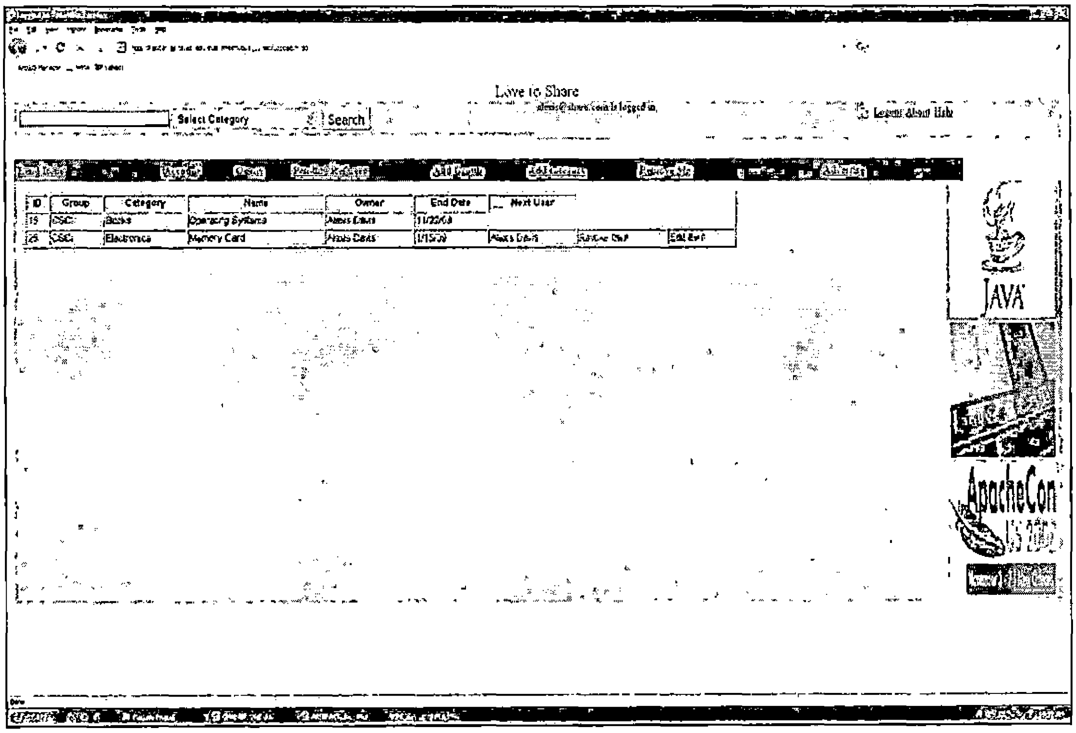
Needed Days : 1

Id : 15
Advanced Concepts in operating Systems Author :Mukesh Singhal and Niranjan G. Shivaratri

Figure 28. Needed Days Page

## 4.2.8 Account Page

Every logged-in user can view the brief details of all the items that he is currently holding. This includes the end date, any next user, owner, group, category and itemname.

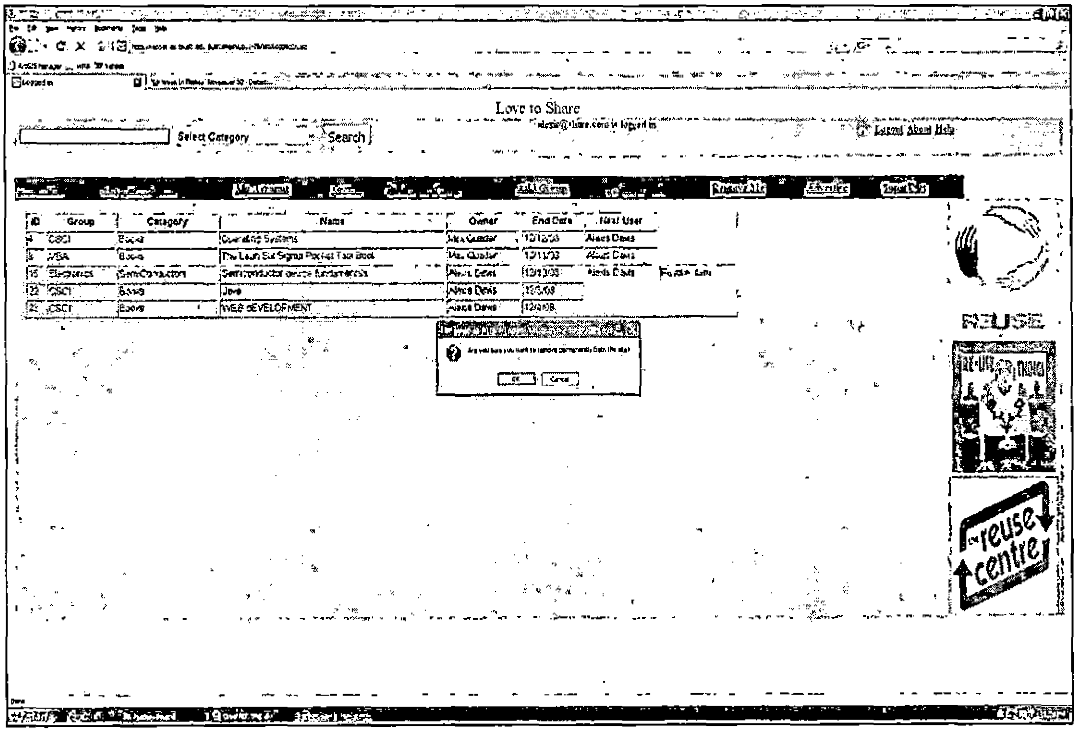

Love to Share

alexis@share.com is logged in,

| ID | Group | Category | Name | Owner | End Date | Next User | | |
|---|---|---|---|---|---|---|---|---|
| 15 | CSC | Books | Operating Systems | Alexis Davis | 11/22/08 | | | |
| 25 | CSC | Electronics | Memory Card | Alexis Davis | 1/15/[illegible] | Alexis Davis | [illegible] | Edit Item |

Figure 29. Account Page

The logged-in user gets to revoke his posted item. The "revoke item" link and the "edit item" will be displayed once he becomes the current user as well as the next user of the item. On click of the "revoke item" link, the logged-in user will be prompted a confirmation box to ensure that he really wanted to revoke the item. If yes, the item will be deleted from the database. On click of the "edit item" link, the user can edit the details of the item.

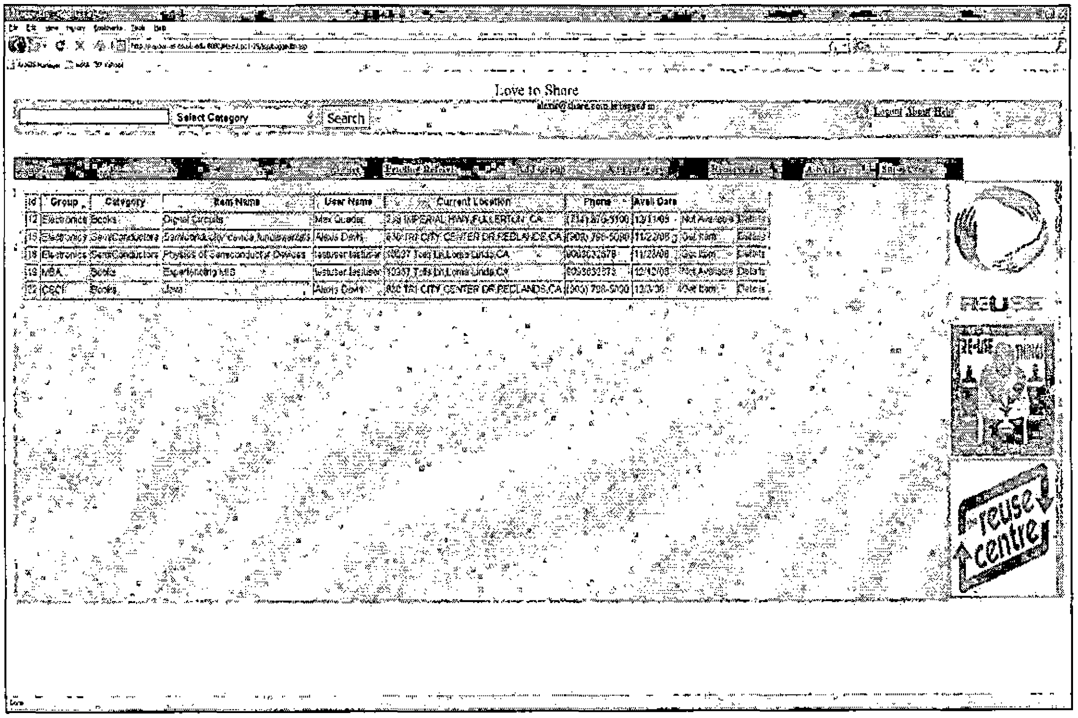

Love to Share

| ID | Group | Category | Name | Owner | End Date | Next User | |
|---|---|---|---|---|---|---|---|
| 4 | CSCI | Books | Operating Systems | [illegible] | [illegible] | Alexis Davis | |
| 5 | [illegible] | Books | The Lean Six Sigma Pocket Tool Book | [illegible] | [illegible] | Alexis Davis | |
| 16 | Electronics | SemiConductors | Semiconductor device fundamentals | Alexis Davis | [illegible] | Alexis Davis | [illegible] |
| 22 | CSCI | Books | Java | Alexis Davis | [illegible] | | |
| 2[illegible] | CSCI | Books | WEB DEVELOPMENT | Alexis Davis | [illegible] | | |

Figure 30. Revoke Item Function Page

### 4.2.9 Owner Page

The logged-in user can view the details of all the items that he owns. This includes the details of the current user, the availability date, status and the details of the item. The owner of these items can get or request for the available items by following the details link.

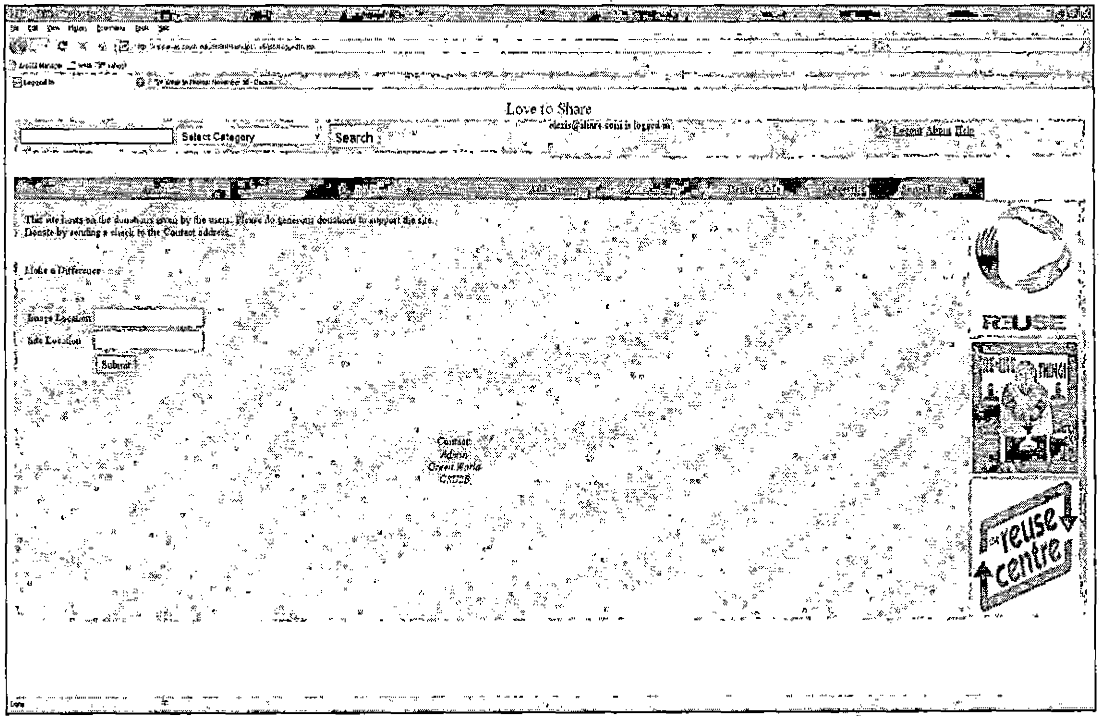


| Id | Group | Category | Item Name | User Name | Current Location | Phone | Avail Date | | |
|---|---|---|---|---|---|---|---|---|---|
| 12 | Electronics | Books | Digital Circuits | Max Queder | 2?? IMPERIAL HWY,FULLERTON, CA | (714) 879-5100 | 12/11/08 | Not Available | Details |
| 15 | Electronics | SemiConductors | Semiconductor Device Fundamentals | Alexis Davis | 630 TRI CITY CENTER DR,REDLANDS,CA | (909) 798-5090 | 11/22/08 | Get Item | Details |
| 18 | Electronics | SemiConductors | Physics of Semiconductor Devices | testuser testuser | 10337 Toss Ln,Loma Linda,CA | 9093632678 | 11/23/08 | Get Item | Details |
| 19 | MBA | Books | Experiencing MIS | testuser testuser | 10337 Toss Ln,Loma Linda,CA | 9093632673 | 12/12/08 | Not Available | Details |
| 22 | CSCI | Books | Java | Alexis Davis | 630 TRI CITY CENTER DR,REDLANDS,CA | (909) 798-5090 | 12/3/08 | Get Item | Details |

Figure 31. Owner Page

### 4.2.10 Advertise Page

WRMS has been developed to be operated as a non-profit site. But still it might need some minimal income to be able to spend on maintenance cost. Obtaining this income is possible through donations made by check to the contact address. The system allows the users to provide links to their respective website for advertisements. The user can provide the URL for the location of the image to be displayed as well as the URL for the site to which the link should be redirected to. The link will be opened in a new tab. The system performs the validations of the form fields. This is done by checking whether the input fields confirm to the URL format. The system tries to open an URLConnection [6]. If the connection could be opened, the form fields are inserted to the database; else the user is informed to enter a correct URL. The image location, site location and the logged-in user name are added to the database. The system picks three images on random from the database every time the page is reloaded and displays the links on screen.

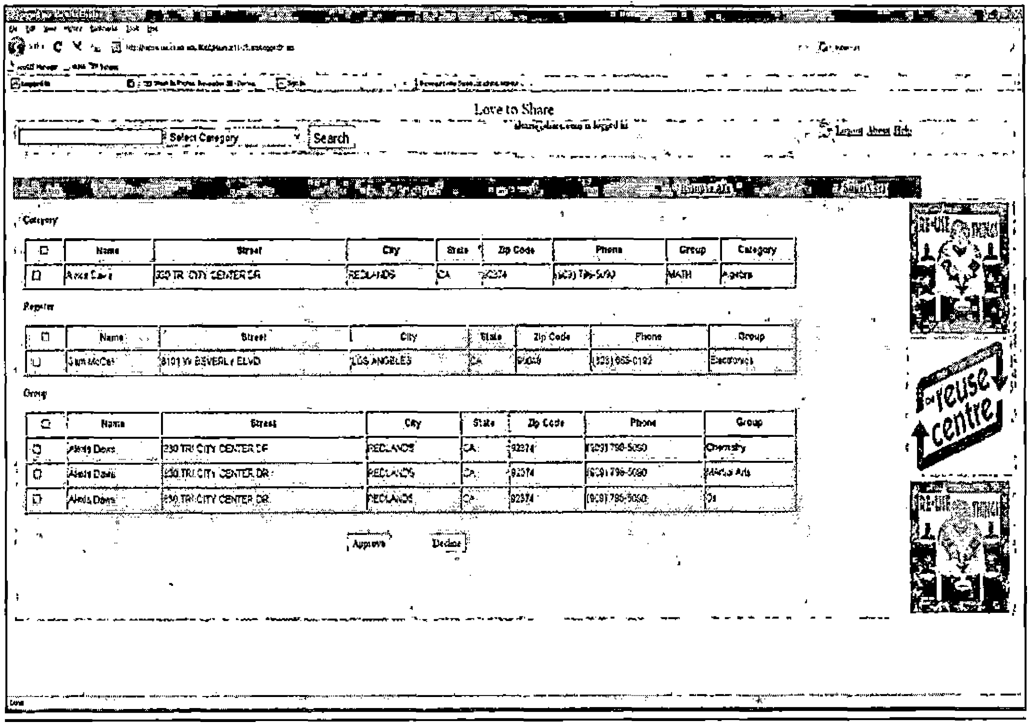


This site lasts on the donations given by the users. Please do generous donations to support the site.
Donate by sending a check to the Contact address.

Figure 32. Advertise Page

### 4.2.11 Pending Referrers Page

Each newly registering user has to get the reference approval of the requested member of the group. Each of the members can view the reference request on the pending referrers' page. This page gives the listing of all the requests that were made to this member for registering to the system, adding a group and also for adding a category. The member can get the details of the user who made the request as well as the reason for the request. The member can approve or decline all requests by checking the check box at the header of each of the tables or can approve or decline individual requests. Once the requests has been approved or declined, the logged-in member will be displayed the information about the action. The approved users will be allowed to sign-in.



Love to Share

Category

| □ | Name | Street | City | State | Zip Code | Phone | Group | Category |
|---|---|---|---|---|---|---|---|---|
| □ | Alexis Davis | 230 TRI CITY CENTER DR | REDLANDS | CA | 92374 | (909) 796-5090 | MATH | Algebra |

Register

| □ | Name | Street | City | State | Zip Code | Phone | Group |
|---|---|---|---|---|---|---|---|
| □ | Sam McCall | 8101 W BEVERLY BLVD | LOS ANGELES | CA | 90048 | (323) 655-0192 | Electronics |

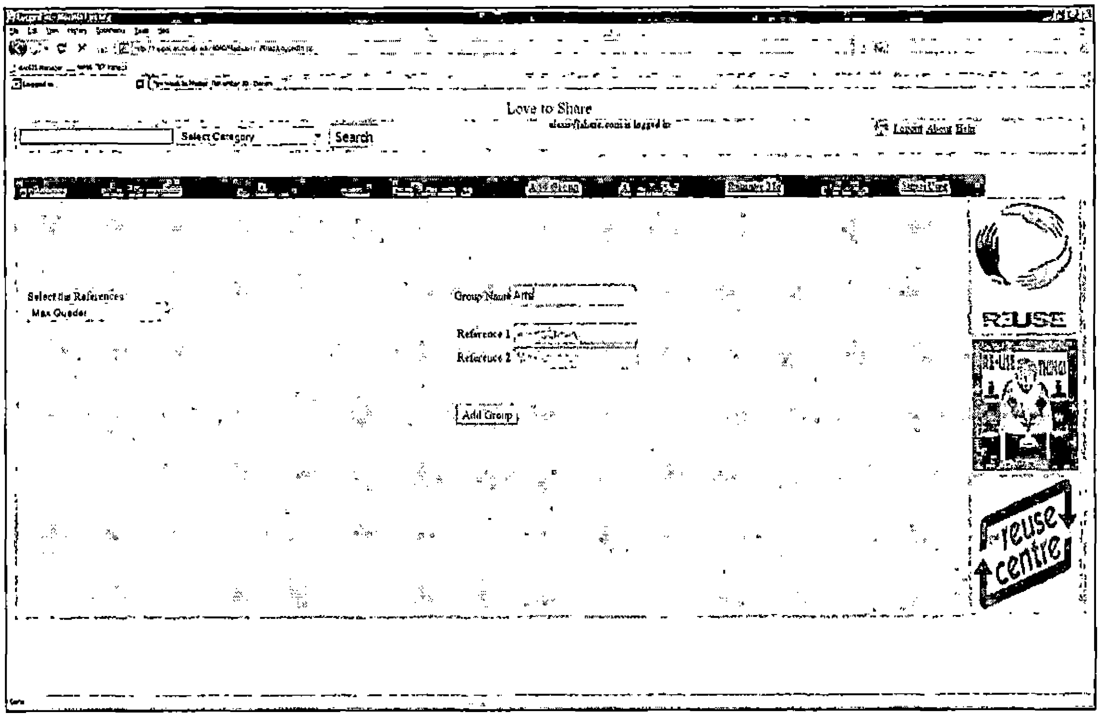
Group

| □ | Name | Street | City | State | Zip Code | Phone | Group |
|---|---|---|---|---|---|---|---|
| □ | Alexis Davis | 230 TRI CITY CENTER DR | REDLANDS | CA | 92374 | (909) 796-5090 | Chemistry |
| □ | Alexis Davis | 230 TRI CITY CENTER DR | REDLANDS | CA | 92374 | (909) 796-5090 | Martial Arts |
| □ | Alexis Davis | 230 TRI CITY CENTER DR | REDLANDS | CA | 92374 | (909) 796-5090 | Oil |

Figure 33. Pending Referrers Page

## 4.2.12 Add Group Page

The members can create a new group in this Add Group page. The members name the group and request the approval of two references. The approval of the references has to be obtained from the super users. The combo box for the superusers is loaded on Page load dynamically through an Ajax request to server, which fills in the data from the database. The selected references will be displayed in a disabled text box. Two same references cannot be submitted by the user. Once Add Group request has been submitted, the system checks if the group already exist. If the group already exists, the system displays a message. The group is displayed, only after being accepted by both the requested references.



Select the References: Max Quader

Group Name: Arts

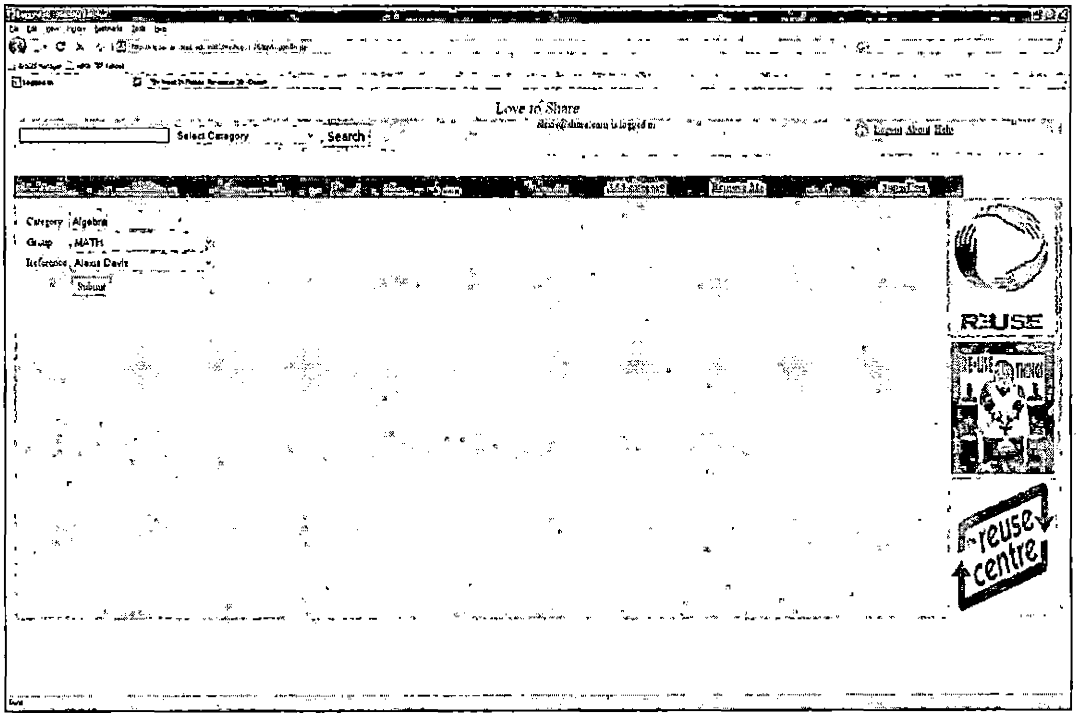
Reference 1

Reference 2

Figure 34. Add Group Page

### 4.2.13 Add Category Page

The members can add a new category by clicking on the Add Category page. The user can select the group he wants to add the category to. The combo box is filled in dynamically on Page load by an Ajax request to the server. The server selects all approved group from the database and fills in the combo box. Once the user selects the group, the system sends an Ajax request to the server. The server selects all the members of the group and dynamically fills in the combo box. The system checks for the input field validations to make sure, the user does not submit empty category. Once submitted, the system checks if the category already exists in the database. If the category already exists, the user is given the appropriate message; otherwise the submitted category will be added to the database. The category will be displayed after the requested references have approved the request.



Category Algebra

Group MATH

Reference Alexis Davis

Figure 35. Add Category Page

### 4.2.14 Remove Me Link

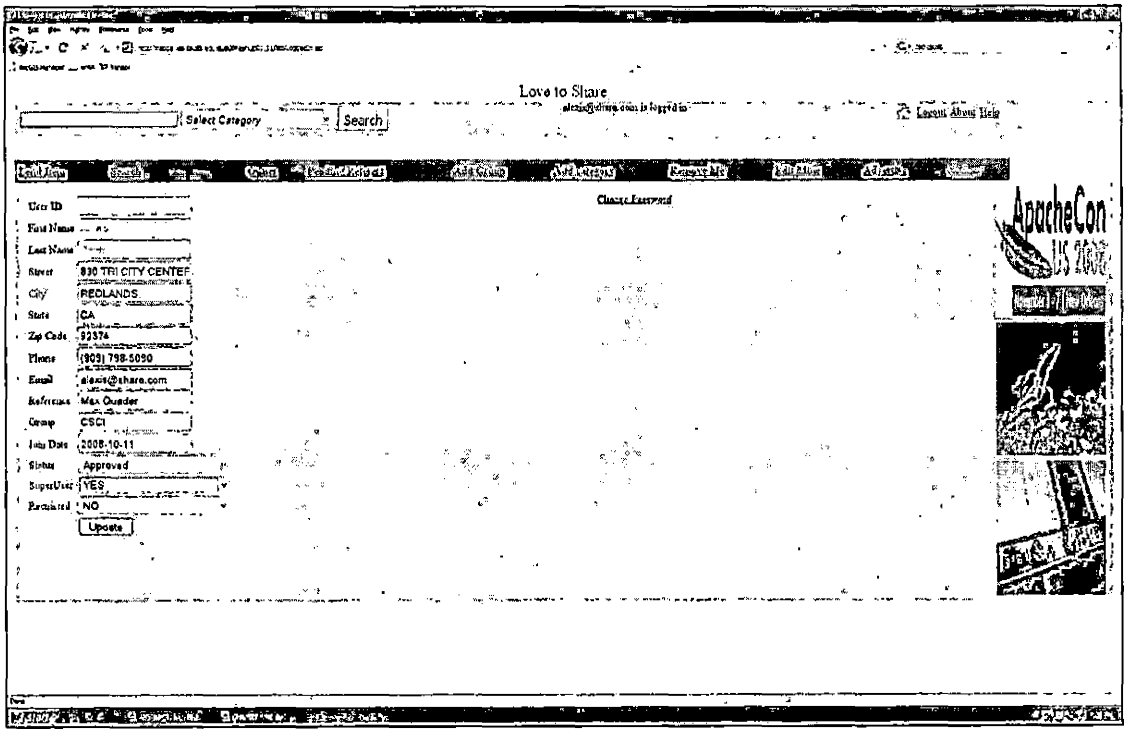
The user can check out from the system by clicking on the "Remove Me" link.  On click of this link, the system prompts a confirmation box. If ok, then the system checks different conditions like checking whether the user

- Has revoked all items he own.
- Is no longer the current user or the next user of any of the items within the system?
- Has approved or declined all reference requests.

If all the conditions are true, then the user is allowed to remove from the site permanently. Otherwise, the user is informed of the reason for not able to remove from the site.

### 4.2.15 Edit Mine Page

The user can edit his/hers details and view the status. The form provides a link for changing the password of the logged-in user.



User ID
First Name
Last Name
Street 930 TRI CITY CENTER
City REDLANDS
State CA
Zip Code 92374
Phone (909) 798-5090
Email alexis@share.com
Reference Max Quader
Group CSCI
Join Date 2008-10-11
Status Approved
SuperUser YES
Restricted NO
Update

Figure 36. Edit Mine Page

### 4.2.16 Change Password Page

The user is prompted to enter the password. The entered password is message digested in the client side and stored in the database.

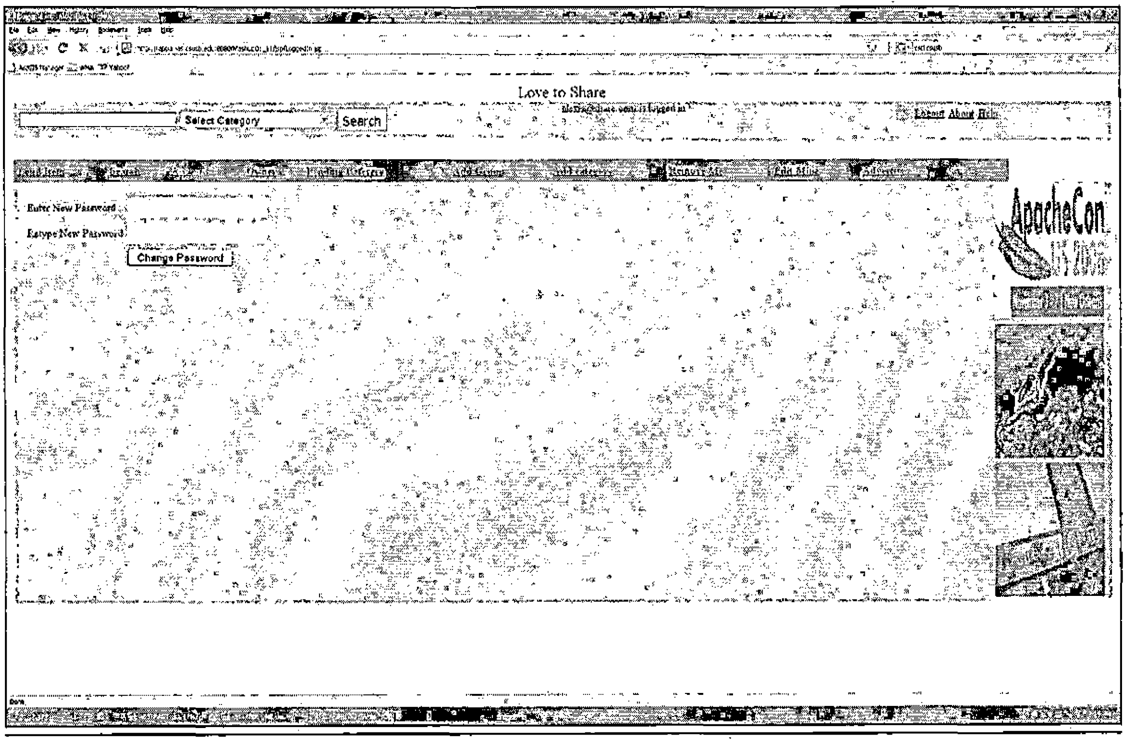


Figure 37. Change Password Page

### 4.2.17 Admin Page

All the users that become the member of the site are approved by the reference they select. There may be times when some of the details of the database tables needs to updated or deleted. The super users get to access all the database tables and do appropriate actions. When the user sign-in, the system loóks for whether the user is super user. If the user is a super user, then the logged-in page will be displaying an additional link that directs him/her to the admin page. This page is accessible only for super users. The super user could update or delete each of the entities appropriately. A message Navigation links are provided for the super user to get back to the original page.

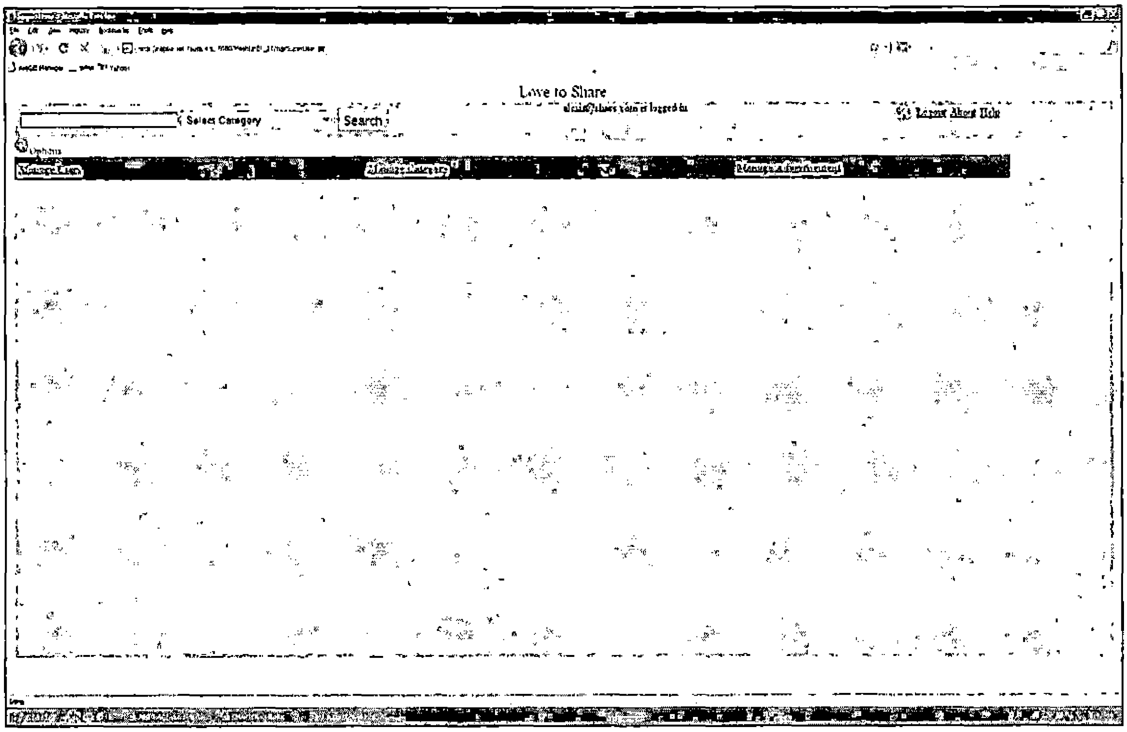
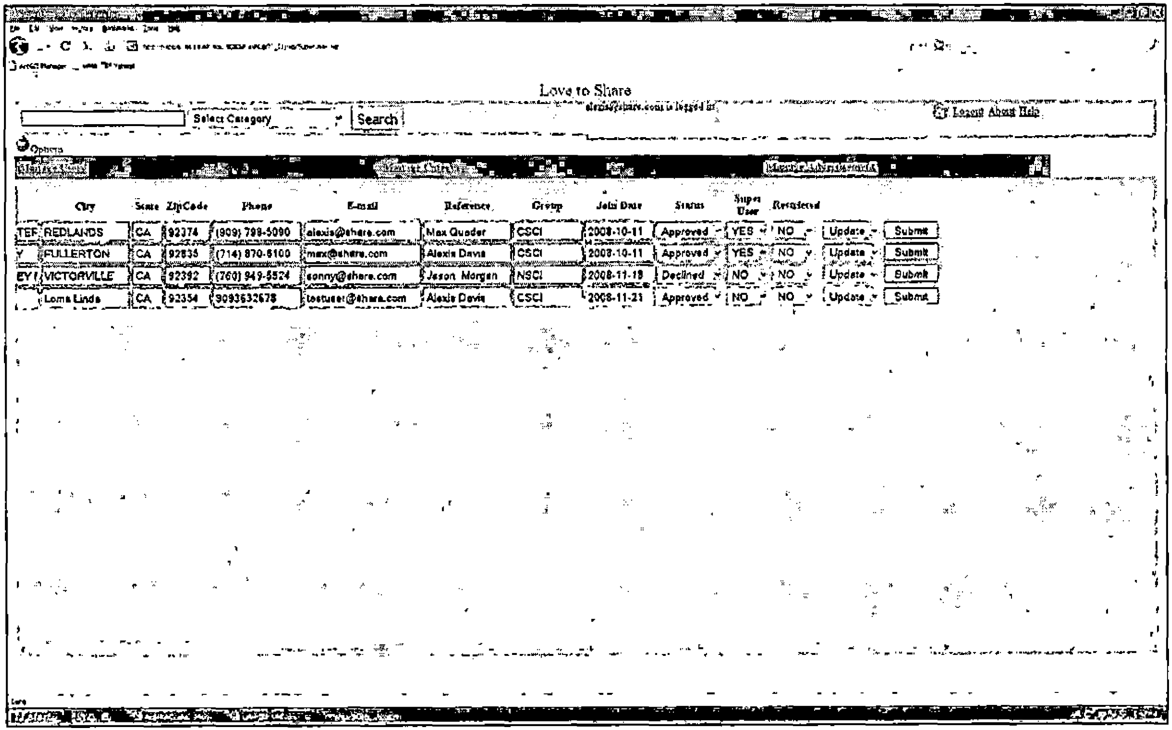


Figure 38. Admin Page

### 4.2.18 Manage Users Page

The page displays all the users and their details in editable form. All the declined users can be deleted or updated. The details as well as the status of all the users could be updated.

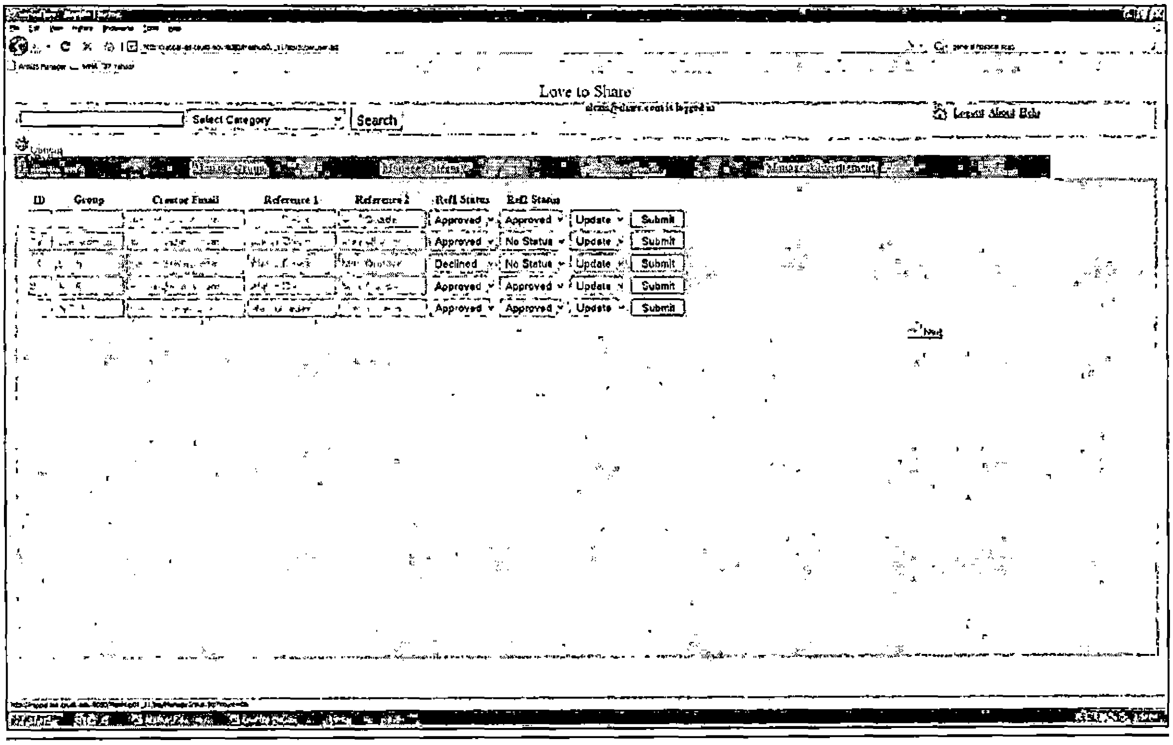


| | City | State | ZipCode | Phone | E-mail | Reference | Group | Join Date | Status | Super User | Restricted | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TEF | REDLANDS | CA | 92374 | (909) 798-5090 | alexis@share.com | Max Quader | CSCI | 2008-10-11 | Approved | YES | NO | Update | Submit |
| Y | FULLERTON | CA | 92835 | (714) 870-6100 | max@share.com | Alexis Davis | CSCI | 2008-10-11 | Approved | YES | NO | Update | Submit |
| EY I | VICTORVILLE | CA | 92392 | (760) 949-5524 | sonny@share.com | Jason Morgan | NSCI | 2008-11-19 | Declined | NO | NO | Update | Submit |
| | Loma Linda | CA | 92354 | 9093632678 | testuser@share.com | Alexis Davis | CSCI | 2008-11-21 | Approved | NO | NO | Update | Submit |

Figure 39. Manage Users Page

### 4.2.19 Manage Group Page

The administrator can view the entire list of groups and its details. If the group has not been approved by both the references or been approved by only one of the references, then the admin can update the approval status or delete the group.

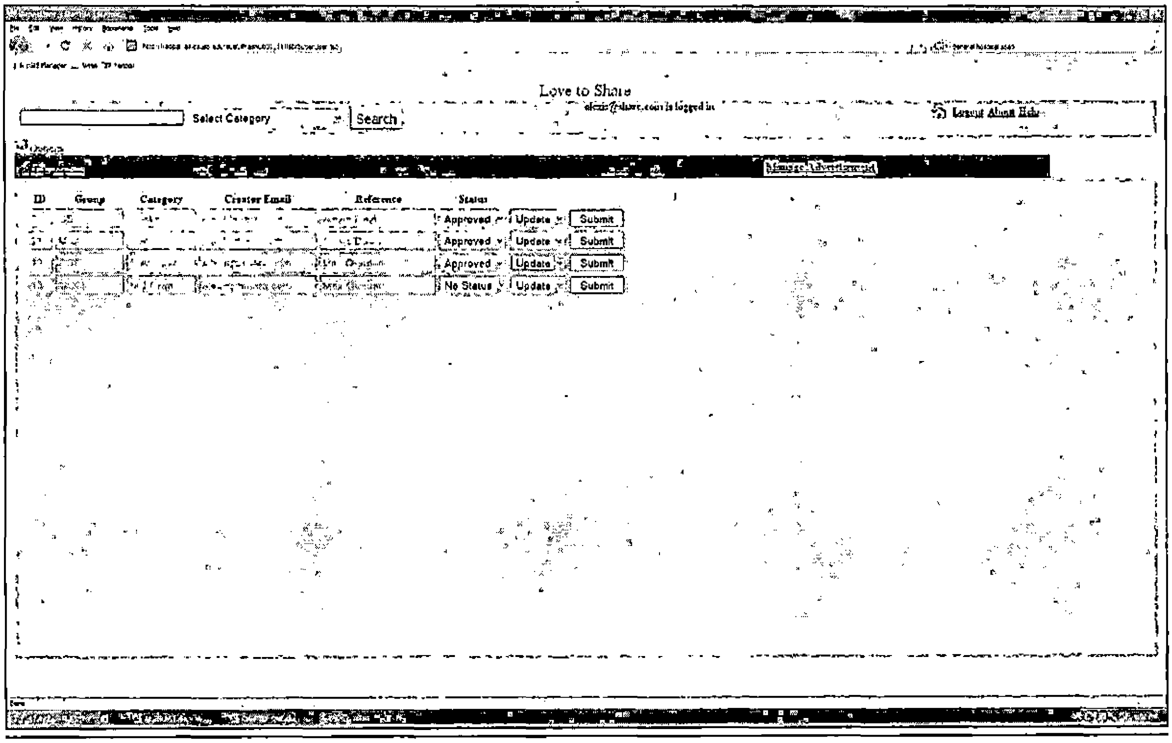


| ID | Group | Creator Email | Reference 1 | Reference 2 | Ref1 Status | Ref2 Status | | |
|---|---|---|---|---|---|---|---|---|
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | Approved | Approved | Update | Submit |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | Approved | No Status | Update | Submit |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | Declined | No Status | Update | Submit |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | Approved | Approved | Update | Submit |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | Approved | Approved | Update | Submit |

Figure 40. Manage Group Page

### 4.2.20 Manage Category Page

The administrator can view the entire list of categories and its details. If the category has not been approved by the reference, then the admin can update the approval status or delete the category.

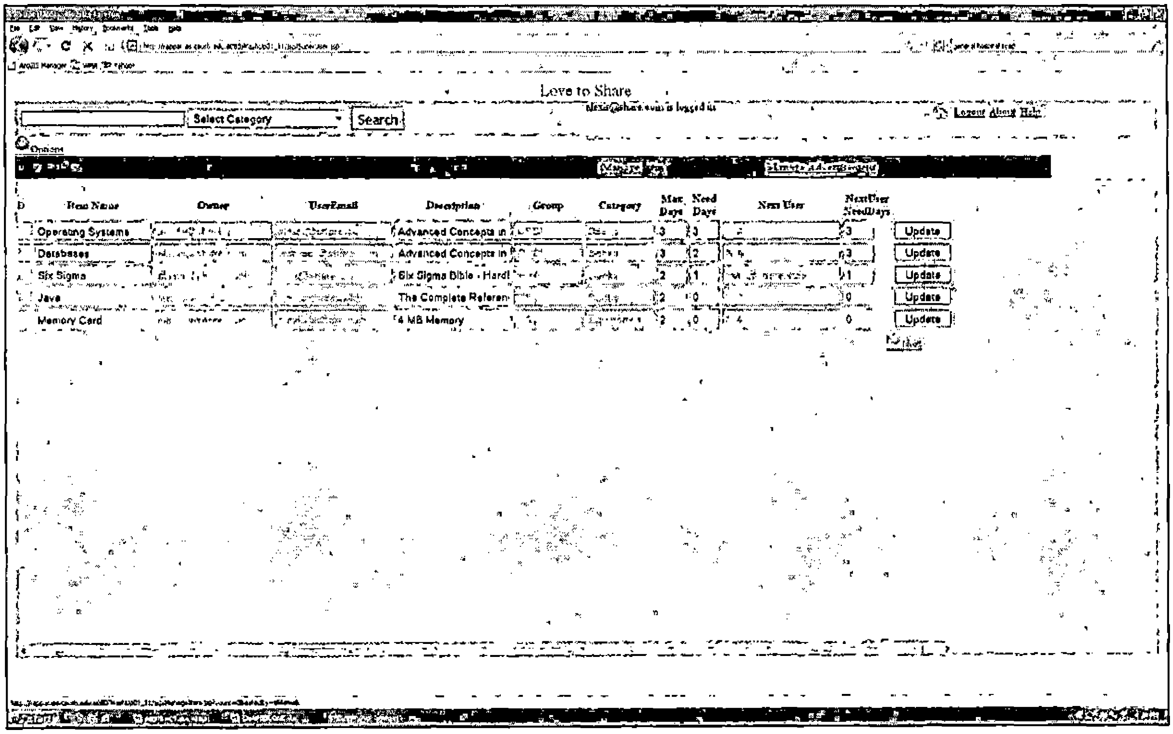


| ID | Group | Category | Creator Email | Reference | Status |
|---|---|---|---|---|---|
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | Approved |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | Approved |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | Approved |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | No Status |

Figure 41. Manage Category Page

### 4.2.21 Manage Item Page

The administrator can view and update all the items and its details. The details are displayed in editable forms.



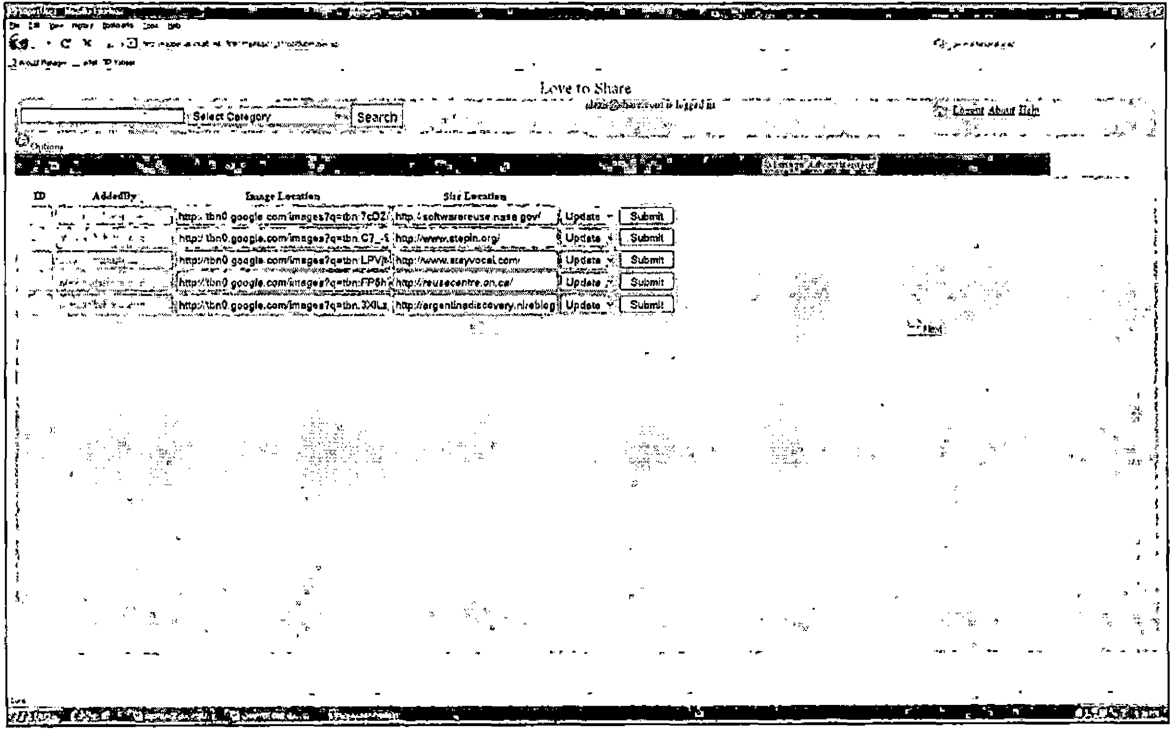
Love to Share

| ID | Item Name | Owner | UserEmail | Description | Group | Category | Max Days | Need Days | Next User | NextUser NeedDays | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [illegible] | Operating Systems | [illegible] | [illegible] | Advanced Concepts in | [illegible] | [illegible] | 3 | 3 | [illegible] | 3 | Update |
| [illegible] | Databases | [illegible] | [illegible] | Advanced Concepts in | [illegible] | [illegible] | 3 | 2 | [illegible] | 3 | Update |
| [illegible] | Six Sigma | [illegible] | [illegible] | Six Sigma Bible - Hard! | [illegible] | [illegible] | 2 | 1 | [illegible] | 1 | Update |
| [illegible] | Java | [illegible] | [illegible] | The Complete Referen | [illegible] | [illegible] | 2 | 0 | | 0 | Update |
| | Memory Card | [illegible] | [illegible] | 4 MB Memory | [illegible] | [illegible] | 2 | 0 | 4 | 0 | Update |

Figure 42. Manage Item Page

## 4.2.22 Manage Advertise Page

The administrator can view, update and delete all the advertisement links. The links are displayed in editable fields.

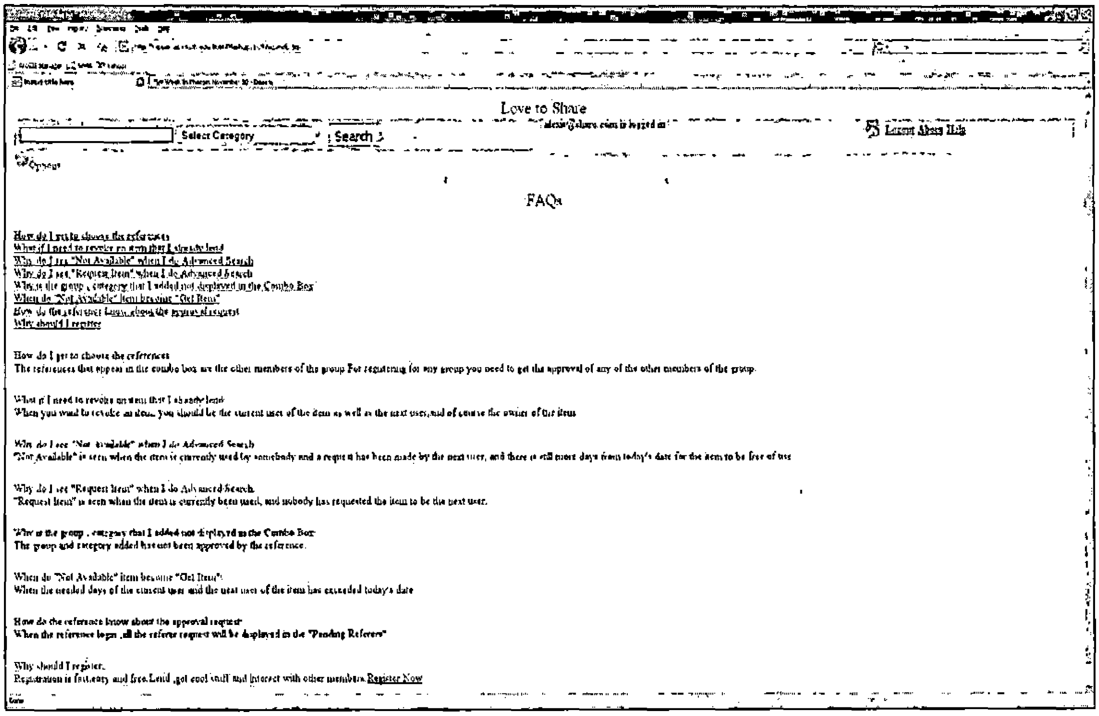


| ID | AddedBy | Image Location | Site Location | | |
|---|---|---|---|---|---|
| [illegible] | [illegible] | http://tbn0.google.com/images?q=tbn:7cOZ | http://softwarereuse.nasa.gov/ | Update | Submit |
| [illegible] | [illegible] | http://tbn0.google.com/images?q=tbn:C7_-S | http://www.stepin.org/ | Update | Submit |
| [illegible] | [illegible] | http://tbn0.google.com/images?q=tbn:LPVj | http://www.stayvocal.com/ | Update | Submit |
| [illegible] | [illegible] | http://tbn0.google.com/images?q=tbn:FP5h | http://reusecentre.on.ca/ | Update | Submit |
| [illegible] | [illegible] | http://tbn0.google.com/images?q=tbn:3XlLz | http://argentinadiscovery.nl/reblog | Update | Submit |

Figure 43. Manage Advertise Page

### 4.2.23 Help Page

The help page contains the FAQ’s that might arise for any user, while using the system.



FAQs

How do I get to choose the references
What if I need to revoke an item that I already lend
Why do I see "Not Available" when I do Advanced Search
Why do I see "Request Item" when I do Advanced Search
Why is the group , category that I added not displayed in the Combo Box
When do "Not Available" item become "Get Item"
How do the reference know about the approval of request
Why should I register

How do I get to choose the references
The references that appear in the combo box are the other members of the group For registering for any group you need to get the approval of any of the other members of the group.

What if I need to revoke an item that I already lend
When you want to revoke an item, you should be the current user of the item as well as the next user,and of course the owner of the item

Why do I see "Not Available" when I do Advanced Search
"Not Available" is seen when the item is currently used by somebody and a request has been made by the next user, and there is still more days from today's date for the item to be free of use

Why do I see "Request Item" when I do Advanced Search
"Request Item" is seen when the item is currently been used, and nobody has requested the item to be the next user.

Why is the group , category that I added not displayed in the Combo Box
The group and category added has not been approved by the reference.

When do "Not Available" item become "Get Item"
When the needed days of the current user and the next user of the item has exceeded today's date

How do the reference know about the approval request
When the reference logs ,all the referee request will be displayed in the "Pending Referees"

Why should I register.
Registration is fast,easy and free.Lend ,get cool stuff and interact with other members.Register Now

Figure 44. Help Page

### 4.2.24 Logout Link

When the logout link is pressed, the user is logged out. The current session is invalidated and the cookie is removed from the browser. The user is redirected to the home page. On press of the back button, the system checks for the cookie on load of each page, and if cookie is not present, the page is redirected to the home page.

# CHAPTER FIVE

# SYSTEM VALIDATION

The system validation test is a kind of test process that ensures that the software program meets the expectation of the user. The purpose of the system validation is to provide the highest degree of assurance that a particular process will produce consistent results and meet predetermined specifications and quality attributes. This can also guarantee the system performance and reliability. Lots of bugs from the code base could be eliminated through the validation process, which makes the software bug free. Meeting security standards is not possible without having a validation process and protecting sensitive user data from malicious users is the primary goal of testing.

## 5.1 Unit Test

Unit test is the basic level of testing where individual components are tested to ensure that they operate correctly. These individual components could be classes, methods, etc. The unit testing results of WRMS are shown in table 6.

Table 6. Unit Test Results (Forms)

| Forms | Tests Performed | Results |
|---|---|---|
| SignIn Page | • Verify that the username and password match and only approved and allowed users are able to login.<br>• Verify that the messages send back to the user on error are the correct messages.<br>• Validate the input fields.<br>• Verify that the dynamic combo boxes are filled in as expected. | Pass |
| Register Info Page | • Verify that the system allows only unique mail-ids to be entered.<br>• Check whether the on click of group combo box is filled with respective members. | PASS |
| LoggedIn Page | • Check if the user-email is correctly assigned as cookie.<br>• Verify that link for admin is displayed, and only displayed when admin is logged in.<br>• Verify that the included header appears for every page.<br>• Verify that the filters work as expected, preventing access to the page without logging-in. | PASS |
| Advanced Search Page | • Check whether the exact search function works as expected.<br>• Verify that the validations are done on submit, even if category combo is not selected.<br>• Verify all items are displayed.<br>• Verify that the page navigation works as expected. | PASS |
| Edit Mine Page | • Check if all of the logged-in users' details appear in the form. | PASS |

| Forms | Tests Performed | Results |
|---|---|---|
| Change Password Page | • Check if passwords entered in the fields match.<br>• Check all button work properly. | PASS |
| Owner Page | • Checks weather all the items that the logged-in members own are displayed. | PASS |
| Details Page | • Verify that the correct item location is geocoded.<br>• Check all button work properly.<br>• Verify that the services called are working as expected.<br>• Check if the links of the option page directs to the correct page. | PASS |
| Needed Days Page | • Check whether the contents of the dynamic combo box generated include within the max days.<br>• Check weather the id and description displayed are of selected item. | PASS |
| My Account Page | • Verify revoke item link appears only if owner, user, next user are the logged-in user.<br>• Verify confirmation box is prompted on click of revoke item.<br>• Check whether the item is removed from the database.<br>• Check whether all of the items that user holds are displayed. | PASS |
| Pending Referrers Page | • Check whether all the referrers are displayed in relevant tables.<br>• Check all buttons work as expected.<br>• Verify the information displayed in the tables. | PASS |

| Forms | Tests Performed | Results |
|---|---|---|
| Add Group Page | • Verify that the dynamic combo box of group creators display properly.<br>• Verify that the form could not be submitted, unless the references are selected. | PASS |
| Add Category Page | • Verify that each newly added group is displayed in the group combo box.<br>• Verify that the validations are done on submit, even if category combo is not selected. | PASS |
| Lend Item Page | • Check whether the message is displayed to the user after adding the item.<br>• Validate the input fields.<br>• Verify that the item is added in the database. | PASS |
| Advertise Page | • Verify that contact information is displayed.<br>• Verify form data is inserted to the database and submitted to the server, only on url field input. | PASS |
| Admin Page | • Check if links of the orphan page directs to correct page.<br>• Verify that only super users are able to access this page. | PASS |
| Manage User Page | • Check whether the display contains all the users.<br>• Check if the users are updated and deleted properly. | PASS |
| Manage Item Page | • Verify that the update message appears once the item updated.<br>• Verify all items displayed. | PASS |

## 5.2 Sub System Testing

The next step in the testing process is the subsystem testing where all related units from a subsystem do a particular task. Thus interface errors from a front-end perspective and the particular functionalities from the back-end point of view could be easily detected. Table 7 shows the subsystem test results in detail.

Table 7. Sub System Test Results

| Sub System | Tests Performed | Results |
|---|---|---|
| Remove Me Sub System | • Make sure the system prompts a confirmation box before trying to remove the user.<br>• Make sure the system checks for the user in all other tables and prompts the user to take necessary steps, before removing from the database. | PASS |
| Browsing Sub System | • Verify the page shows properly after the user click on the page link.<br>• Check if the subsystem checks for user privileges before showing pages.<br>• Check if the filters filter out the request for a logged-in page from a user if the user is not signed in. | PASS |
| Password Digest | • Check if the password entered are digested and stored in the database. | PASS |

## 5.3 System Testing

System testing is a process of testing that uses real data that the system is intending to manipulate, to test the system. All unit tests, subsystem tests are integrated into one system. System testing is done by testing the system using various types of possible data, to see if the system works well for all the data. The test is usually done in a production environment, when the system is about to be deployed. The deployment is a replication of the tested system into production servers. To make sure everything is running, the tests on the production environment have to be done.

System testing of WRMS system begins with the following steps (Table 8):

Table 8. System Test Results

| System testing | Results |
|---|---|
| Install the WRMS system into server | PASS |
| Start up servers such as Apache Tomcat Web Server, MySQL database server. | PASS |
| Run tests on all forms using real data | PASS |

# CHAPTER SIX

# MAINTENANCE MANUAL

Maintenance manual is very important regardless of how easy the system is to use. The maintenance manual records all the information that is used to set up or back up the system. It is extremely important to follow the steps of the maintenance manual, to make sure the system works smoothly, meets the expectations of the users and maintain sensitive data securely. In WRMS there are four major issues: Hardware Selection, Software Installation, WRMS installation.

## 6.1 Software Installation

The deployment of the WRMS depends on many factors. The amount of load a system will have on a given day should be one of the considerations. Based on the requirements of the system, proper hardware should be dedicated to run WRMS. Since WRMS is implemented in Java, the system can be run on any Operating System like Windows XP, Windows 2000, Linux. WRMS uses Centos 5 Linux Operating System, JRE (Java Runtime Environment), Apache Tomcat Web Server, MySQL database and JDBC connection to run the program.

### 6.1.1 Centos 5.0 Installation

Centos [10] is a Linux base operating system that is offered freely and can be downloaded from the internet. The reason being Centos an open source and does not need to be purchased is one of the reasons for choosing the operating system for WRMS.

Download the latest version of the Centos 5 operating system from one of its mirror sites available in

http://mirror.centos.org/centos/5/isos/

and burn the files into a CD-ROM.

Delete the existing operating system if any.

Install the operating system by inserting CD 1 into the CD-ROM drive and start-up the machine.

The machine startup via CD-ROM to begin installing Centos 5

Enter the required information such as network setting and hardware environment. Unselect every package group selection.

### 6.1.2 Java Installation

To run java classes, JRE (Java Runtime Environment) has to be installed.

To install in Linux, enter as root

- Before installing, change to the directory in which you want to install.

  cd <directory path name>

  To install in the /usr/java/ directory, cd /usr/java/

- To change the permission of the file you downloaded to be executable, type:

  chmod a+x jre-5u<version>-linux-i586.bin

- Verification of permission to execute the file can be checked by typing: ls -l

- To start the installation process,

  type: ./jre-5u<version>-linux-i586.bin

The JRE is installed in jre1.5.0_<version> sub-directory under the current directory. In this case, the JRE is installed in the /usr/java/jre1.5.0_<version> directory.

The installation is completed.

### 6.1.3 Tomcat Installation

Tomcat Web Server [7] is an open source project of Apache Jakarta, which is a web container to process JSP and JAVA Servlet programs and to serve static web pages.

Download the tar.gz from

http://tomcat.apache.org/download-55.cgi to /usr/tomcat/

A directory called jakarta-tomcat-5.0.27 will be created and the Tomcat files will automatically expand under it.

- Java is installed into usr/java
- Tomcat is installed into usr/tomcat
- Insert the following lines inside /etc/profile

```
export JAVA_HOME=/usr/java
export CATALINA_HOME=/usr/tomcat
```

Now, save the file and reboot the system to ensure that all changes have taken effect. First ensure that CATALINA_HOME and JAVA_HOME are correctly set. To do this, open a terminal and type the following:

```
# echo $CATALINA_HOME
# echo $JAVA_HOME
```

If a blank line is seen first correct the environment variables before continuing. If everything is fine, start Tomcat with the following command. As root,

```
# $CATALINA_HOME/bin/startup.sh
```

Open a browser and point the URL to http://localhost:8080 , to check if Tomcat is running fine. The default Tomcat welcome page will be seen.

To stop Tomcat, as root,

```
# $CATALINA_HOME/bin/shutdown.sh
```

If Tomcat does not start after downloading the zip file, the cause is probably due to permissions. Ensure that the following files are executable: inside $CATALINA_HOME/bin directory,

```
# chmod +x startup.sh
# chmod +x shutdown.sh
# chmod +x tomcat.sh
```

After making the files executable, restart Tomcat.

### 6.1.4 Dojo Installation

Dojo [2] is an Open Source DHTML toolkit written in JavaScript. Dojo allows to easily build dynamic capabilities into web pages and any other environment that supports JavaScript. First choose the directory in your system, where Dojo has to be installed. If the directory is "C:\mydojo" move the downloaded file into the "C:\mydojo" directory and unzip it using WinZip tool or winrar tool. Download the Dojo toolkit directly from http://dojotoolkit.org/downloads. The JavaScript files of Dojo will be downloaded to the folder.

### 6.1.5 JAVA Database Connectivity (JDBC)

The API used to execute SQL statement is different for each database engine. But JAVA programmers are freed from such portability issues. For JAVA programmers, a Single API is present, the Java Database Connectivity API (JDBC), [6] which is portable between database engines. The JDBC library provides an interface for executing SQL statements. It provides the basic functionality for data access. First we can download the file

Mysql0jdbc.tar.gz at

http://dev.mysql.com/downloads/connector/j/5.1.html

and copy the file mysql-connector-java-5.1.7.tar.gz to /usr/tomcat/common/lib/.

### 6.1.6 MySQL Installation

The database used in WRMS is MySQL. MySQL [8] is an open source software. MySQL is a good choice for the database systems, because it provides easy connectivity with JAVA programs and the access performance is also good.

Because MySQL maybe installed on the Centos 5.0 when the operating system is installed, the first thing we have to do is to check if the MySQL is already in the operating system. Use the command to check if MySQL exists in the operating system:

```
rpm -q mysql
```

To connect to MySQL,

```
     mysql -h <servername> -u root -p
```

## 6.2 Variable Modification

We have to change some environment variables in the Linux system and server.xml in Tomcat server configuration directory.

### 6.2.1 System Variables

- Open the file "server.xml" in the directory "usr/tomcat/jakarta-tomcat-5.0.27/conf" via "vi" or any other editor.
- The variable "path" in Context indicates the context path of the web application. The default value would be "/wrms".
- The variable "docBase" in Context is the files directory for the web application. The default value would be "wrms"
- The variable "variable" in Logger is the absolute or relative pathname of a directory in which log files created by this logger will be placed. The default value would be "localhost_log".

Now let us look down at the parameter setting.

  - The parameter "contextPath" indicate the context path for the system, which is same as the value of path.

### 6.3 Installation/Migration

Application files are stored in webapps folder of Tomcat.

- All the JSP files are stored in $CATALINA_HOME/webapps/wrms/jsp/
- All the HTML files are stored in $CATALINA_HOME/webapps/wrms/html/
- All the CSS files are stored in $CATALINA_HOME/webapps/wrms/css/
- All the Javascript Validation files are stored in $CATALINA_HOME/webapps/wrms/js/Validations/
- All the Dojo files are stored in $CATALINA_HOME/webapps/wrms/js/Dojo/
- All the Javascript files used for consuming the Web-Services are stored in $CATALINA_HOME/webapps/wrms/js/ConsumingServices/
- All the images are stored in $CATALINA_HOME/webapps/wrms/images/
- All the classes are stored in $CATALINA_HOME/webapps/wrms/WEB-INF/classes/
- All the database files are stored in /MySQL/share

## 6.4 Backup

Backup is a very important action for an administrator. Backup can help recover the system to its original status. In WRMS, we need to backup two components, system files and database files.

### 6.4.1 System Backup

All the WRMS system files are in the directory "webapps/wrms/" and all its subdirectories. So the system files could be backup by backing up all the files in the subdirectories. The method that could be used to backup these files is to compress the directory of "/webapps/wrms" which includes all the files and the subdirectories within the "wrms" folder by the compress program "tar". The following command can be used to backup the system files:

```
tar -cf WRMS.tar /webapps/wrms
```

### 6.4.2 Database Backup

We could use the mysql_dump command to backup the database used by the WRMS. The following command is used to backup the database:

```
mysqldump share | gzip > share.zip
```

After executing the above backup command, the file share.zip would be the backup file of the database.

# CHAPTER SEVEN

# CONCLUSION AND FUTURE DIRECTIONS

## 7.1 Conclusion

We could achieve our aim of developing a Web-based system where the members could share their items and keep track of them. The members are able to view, edit and revoke all their items. The administrative abilities provide for a better management of the system. The system has been able to address many of the security issues.

## 7.2 Future Directions

WRMS provides an efficient system for the management and coordination of the items. As a future work on this project, effort could be made on to be able to send event driven e-mail to the users. The e-mail could be used for sending notifications of pending referrers of any member. E-mail notifications can also be made to the requested user, when the current user has finished using the item.

# APPENDIX A

# JAVA SOURCE CODE

```
                       //send REST request to the Server

      public static String sendRequest(String urlString) throws
Exception {
                  if(urlString==null){
                        throw new Exception("URL is null");
                  }
                  URL url = new URL(urlString);
                  URLConnection urlConnection = url.openConnection();
                  urlConnection.setDoInput(true);
                  urlConnection.setDoOutput(true);
                  urlConnection.setUseCaches(false);
                  urlConnection.connect();
                  InputStream ins = urlConnection.getInputStream();
                  InputStreamReader isr = new InputStreamReader(ins);
                  BufferedReader br = new BufferedReader(isr);
                  String line = null;
                  StringBuffer sb = new StringBuffer();
                  while ((line = br.readLine()) != null) {
                        sb.append(line);
                  }
                  ins.close();
                  return sb.toString();
            }


       function parseTrafficResponse(xmlDoc){
         var bounds = new GLatLngBounds();
            var items= xmlDoc.getElementsByTagName("Result").length;
       if (items==0){
         document.getElementById("traffic").innerHTML=
       "No traffic incidents";
           }
            for (i = 0; i < items; i++){
           var type= xmlDoc.getElementsByTagName("Result")[i].
getAttribute("type");
          var latitude=
xmlDoc.getElementsByTagName("Latitude")[i].childNodes[0]. nodeValue;
           var longitude=
xmlDoc.getElementsByTagName("Longitude")[i].childNodes[0]. nodeValue;
var title= xmlDoc.getElementsByTagName("Title")[i].
childNodes[0].nodeValue;
var description= xmlDoc.getElementsByTagName("Description")[i].
childNodes[0].nodeValue;
var severity= xmlDoc.getElementsByTagName("Severity")[i].
childNodes[0].nodeValue;
           displayDescription(i+1,description);
      var myGeographicCoordinates = new GLatLng(latitude, longitude);
      var marker = createGoogleMarker(myGeographicCoordinates,i,title);
           map.addOverlay(marker);
           bounds.extend(myGeographicCoordinates);
            map.setZoom(map.getBoundsZoomLevel(bounds));
            map.setCenter(bounds.getCenter());
                        }
```

```
                     }


      function parseWeatherResponse(xmlDoc){
      var node = xmlDoc.documentElement;
      var channel= node.getElementsByTagName('channel').item(0);
      var item=channel.getElementsByTagName('item').item(0);
      var title=
item.getElementsByTagName('title').item(0).firstChild.data;
      var link =
item.getElementsByTagName('link').item(0).firstChild.data;
      var content='<a href='+ link+ ' target=_blank>'+ title +
'</a></br></br></br>' ;

content+=item.getElementsByTagName('description').item(0).
firstChild.data;

document.getElementById("weather").innerHTML=content;
      }


      function displayDirections(){
           map.clearOverlays();
document.getElementById("traffic").style.visibility="visible";
      var itemaddress= "<%=request.getAttribute("itemaddress")%>";
      var
loggedInAddress="<%=request.getAttribute("loggedinaddress")%>";
             if(loggedInAddress==("")){
           alert("You have to log in");
           }
         document.getElementById("traffic").innerHTML="";
      var directionsPanel= document.getElementById("traffic");
      var directions = new GDirections(map, directionsPanel);
      directions.load(loggedInAddress + " to "+ itemaddress);
         }



      public String getUserEmailCookie() {
      try {
      if (cookies != null) {
      for (Cookie c : cookies) {
      if (c.getName().equals("userEmail")) {
      String owner = c.getValue();
      String decodedUserEmail = null;
      owner = owner.replace("_", "=");
      decodedUserEmail = new String(new
sun.misc.BASE64Decoder().decodeBuffer(owner));
            return decodedUserEmail;
            }
            }
            }
                  } catch (IOException e) {
                        e.printStackTrace();
                  }
                  return null;
            }


      public Date addDaysToStartDate(Date startDate, int maximumDays) {

                  Calendar c1 = Calendar.getInstance();
                  c1.set(Calendar.DATE, startDate.getDate());
                  c1.set(Calendar.MONTH,    startDate.getMonth());
                  c1.add(Calendar.DATE,maximumDays);
                  return c1.getTime();
            }


      public void deleteCookie(HttpServletRequest request,
HttpServletResponse response){
                  Cookie[] cookies=request.getCookies();
                  for(int i=0;i<cookies.length;i++){
                         Cookie cookie = cookies[i];
                         cookie.setMaxAge(0);
                         response.addCookie(cookie);
                  }
            }


            public void invalidateSession(HttpServletRequest request){
      HttpSession userSession= request.getSession(false);
                  if(userSession!=null) {
                        userSession.invalidate();
                        }
            }
```

# REFERENCES

1. Yahoo! Developer Network http://developer.yahoo.com/

2. The Dojo Toolkit http://www.dojotoolkit.org/

3. World Wide Web Consortium http://www.w3.org/

4. Mozilla Developer Center https://developer.mozilla.org/

5. S. Herbert, "The Complete Reference, Java2", Tata McGraw-Hill, CA, 2002

6. Developer Resources for Java Technologies http://java.sun.com/

7. Apache Tomcat http://tomcat.apache.org/

8. MySQL http://www.mysql.com/

9. Linux Online http://www.linux.org/

10. CentOS The Community Enterprise Operating System http://www.centos.org/

11. Google Maps API Concepts http://code.google.com/apis/maps/documentation/