



The Rooms and Colloquium System

CS706, Analysis of Software Artifacts
Rooms Team
Fall 2001

Table of Contents

The Rooms and Colloquium System.....	1
Table of Contents.....	2
Table of Figures	4
Introduction.....	5
Purpose.....	5
Document Conventions and Definitions	5
Users.....	5
References	5
Group Mailing List.....	5
Group Web Page	5
Team.....	6
Class Professor	6
Group Mentors	6
Group Members	6
Glossary	7
Requirements	8
Primary Requirements.....	8
Usability.....	8
Viewing.....	8
Reservation.....	8
Administration	8
Security	9
Secondary Requirements.....	9
Usability.....	9
Reservation.....	9
Documentation Requirements.....	10
Use Cases	11
Visitor.....	11
Account User.....	11
Administrator	11
User Case Diagram.....	12
System Design.....	13
Architecture Decisions	13
Front-End	13
Middle-Tier	13
Back-End.....	13
Logical Architecture Design Diagram.....	14
Targets.....	14
Physical Architecture Design Diagram.....	15
Physical Class Diagram	16
Sequence Diagrams.....	17
View today's schedule, View schedule for a specific day	17
View room(s) for specific information.....	18

The Rooms and Colloquium System

View schedule for a specific week.....	19
View information specific reservation for a user-defined time range, identified by purpose, identified by the "maker" of the reservation	20
Reserve a room, Reserve a room for a recurring event.....	21
Make colloquium information available	23
Associate colloquium information with a reservation.....	23
Reserve a room for colloquium.....	24
Modify a reservation.....	24
Cancel a Reservation, Cancel part of a recurring reservation, Cancel the whole recurring reservation.....	25
Change a user's authorization list.....	25
Change a user's authorization level.....	26
Make a room available for reservations	26
Make a room unavailable for reservations	27
Cancel any reservation.....	27
Modify any reservation.....	28
Purge old records (remove "old" reservations)	28
Change the set of room attributes	29
Database Schema	30
User Interface Screen Shot.....	31
Test Plan.....	32
Code Review and System Testing	32
Introduction to Code Review	32
Code Walkthrough.....	32
Code Inspection.....	32
Introduction to System Testing	33
Steps in System Testing	33
Test documentation.....	33
System Testing Reports	34
Package Documentation.....	59
Class User Interface	59
User Manual.....	62
Rooms Help.....	62
Maintenance Plan.....	71
Index.....	72

Table of Figures

Figure 1 Use Case Diagram 1	12
Figure 2 Logical Architecture Design.....	14
Figure 3 Physical Architecture Design	15
Figure 4 General Physical Class Diagram	16
Figure 5 Sequence Diagram, View Today's Schedule, View Schedule for a Specific Day	17
Figure 6 Sequence Diagram: View room(s) for specific information	18
Figure 7 Sequence Diagram: View schedule for a specific week.....	19
Figure 8 Sequence Diagram: View information specific reservation for a user-defined time range, identified by purpose, identified by the "maker" of the reservation.....	20
Figure 9 Sequence Diagram: Reserve a Room, Reserve a Room for a Recurring Event, Cancel a Reservation for UnSuccessful.	21
Figure 10 Figure 6 Sequence Diagram: Reserve a Room, Reserve a Room for a Recurring Event, Cancel a Reservation for Successful.	22
Figure 11 Sequence Diagram Make colloquium information available	23
Figure 12 Sequence Diagram Associate colloquium information with a reservation.....	23
Figure 13 Sequence Diagram Reserve a room for colloquium	24
Figure 14 Sequence Diagram: Modify a reservation.....	24
Figure 15 Sequence Diagram Cancel a Reservation, Cancel part of a recurring reservation, Cancel the whole recurring reservation.....	25
Figure 16 Sequence Diagram Change a user's authorization list.....	25
Figure 17 Sequence Diagram Change a user's authorization level.....	26
Figure 18 Sequence Diagram Make a room available for reservations	26
Figure 19 Sequence Diagram Make a room unavailable for reservations	27
Figure 20 Sequence Diagram: Cancel any reservation.....	27
Figure 21 Sequence Diagram: Modify any reservation.....	28
Figure 22 Sequence Diagram Purge old records (remove "old" reservations).....	28
Figure 23 Sequence Diagram Change the set of room attributes.....	29

Introduction

Purpose

The purpose of the Rooms and Colloquium System is to provide a way for people associated with the Department of Computer Sciences to view, reserve, or to administrate rooms in the Computer Sciences and Statistics Building, 1210 W. Dayton St., Madison, WI. The system also allows a user to make Colloquium information available to be viewed, and to associate that information with a reservation. The Rooms and Colloquium System is geared for three different types of users: public viewers, authenticated users, and authenticated privileged administrators.

Document Conventions and Definitions

Terms used in this document are defined in the companion glossary document.

Users

The users of this program are people in the CS department (anybody with a CS login).

References

This is a group project for The University of Wisconsin – Madison Department of Computer Sciences course CS706, Analysis of Software Artifacts, Fall 2001, Somesh Jha.

Group Mailing List

http://www.cs.wisc.edu/lists/archive/rooms_cs706/

Group Web Page

<http://www.cs.wisc.edu/~minyi/room.html>

Team

The Rooms team is made up of the following people:

Class Professor

- Somesh Jha (jha@cs.wisc.edu)

Group Mentors

- Will Benton (willb@cs.wisc.edu),
- Jerry Tutsch (tutsch@execpc.com)

Group Members

- Brian Bowers (blbowers@cs.wisc.edu),
- Andrew Palmer (palmeran@biostat.wisc.edu),
- Hongwei Zhu (hzhu@cs.wisc.edu),
- Ming Li (mingl@cs.wisc.edu),
- Minyi Xu (minyi@cs.wisc.edu),
- Naijun Zhou (naijun@cs.wisc.edu),
- Keith Noto (noto@cs.wisc.edu)

Glossary

- **User** : A person that uses the rooms program. Either a visitor, acctuser, or room czar.
- **Visitor** : A user who cannot make/change reservations, but can view reservations.
- **Acctuser**: A user with authority to make reservations and to edit and cancel their own reservations.
- **Room Czar (Administrator)**: An Acctuser with authority to perform any operation provided by the rooms program.
- **Reservation**: A period of time associated with a room and a contact. If a room has a reservation at a given time, there can be no other reservations at that time.
- **Contact** : For a given reservation, the user who created the reservation.
- **Room Attributes**: Information associated with each room. This information includes (at minimum): Room number, seating capacity, whether or not there is a table, whether or not there is a projector, whether or not there is a white board, whether or not there is a black board.
- **Colloquium**: an academic meeting at which specialists deliver addresses on a topic or on related topics and then answer questions relating to them

Requirements

These are the requirements for the system gathered from the system customer.

Primary Requirements

Essential Requirements that must be in the system for it to be conceded complete.

Usability

1. A web-based (HTML) user interface. This interface will be accessible via (at least) Lynx and Netscape browsers. The web-based interface will not include frames. The web-based interface will conform to W3C's Web Content Accessibility Guidelines 1.0 ([link](#) as of October 8, 2001). (*User(s): All Users*)
2. An interface with the colloq program. (*User(s): N/A*)

Viewing

1. View the current room reservations by {room/time/contact/purpose} (time includes date). (*User(s) Visitors*)
2. View the attributes of a room (i.e. seating capacity) (*User(s): AcctUsers*)

Reservation

1. Reserve a given room (identified by room number) for a specified time range. Each reservation must be associated with a contact (whomever makes the reservation) and a purpose (a brief piece of text). (*User(s): AcctUsers*)
2. Reserve a given room (identified by room number) for a specified recurring time range (i.e. weekly), and this recurrence for a specified time range (i.e. the next 6 weeks). (*User(s): AcctUsers*)
3. Edit a given reservation that they have made (location, date/time, purpose, recurrence). If a reservation is a recurring reservation, the user will be able to edit the recurrence itself, as opposed to editing each instance of the reservation. Editing a recurrence includes the ability to cancel single instances of reservations in a recurring pattern of reservations. (*User(s): AcctUsers*)
4. Cancel a reservation that they have made. (*User(s): AcctUsers*)

Administration

1. Cancel a reservation that any user has made, and send an email to that informing that user of the situation. (*User(s): Room Czar*)
2. Edit a given reservation that any user has made (location, date/time, purpose, recurrence) and inform the user of the situation. (*User(s): Room Czar*)
3. Add new rooms to the system. (*User(s): Room Czar*)
4. Remove existing rooms from the system. (*User(s): Room Czar*)
5. Edit the attributes of an existing room. (*User(s): Room Czar*)

Security

1. Provide authentication for users. (*User(s): AcctUsers*)
2. Provide a way to cancel a reservation transaction. Between defining and committing a room reservation, an acctuser will be able to cancel the reservation without needing to re-authenticate herself/himself before making new reservations. (*User(s): AcctUsers*)
3. Provide a way to retrieve a forgotten password. (*User(s): AcctUsers*)
4. Correctly handle the case where two users reserve the same room for the same time. (*User(s): N/A*)

Secondary Requirements

These are desired requirements for the system, but not system critical ones.

Usability

1. A way to easily in force a reservation validation policy. That is, if it is decided that reservations should be removed/reviewed each semester, or that long-term reservations be allowed to last forever, a policy can be agreed upon, and it is possible to set the rooms project up to enforce it.

Reservation

2. The ability to express interest in a reservation, without committing to the reservation (to "pencil-in" a reservation). (*User(s): AcctUsers*)
3. The ability to search for rooms based on room attributes and availability. It should be easy to see a list of potential reservations by entering time and room attribute constraints. (*User(s): AcctUsers*)
4. The ability to search for reservations with text matching the text in the reservation's purpose.
5. The ability to create recurring reservations that recur up until a fixed time: The end of the current semester, the end of the current summer session.
6. The ability to enforce a policy which limits reservations to within a certain time period. For instance, no reservation can be made for later than one year after the time the reservation is made (calendar or academic year).
7. A record of a "reason" (text) associated with each reservation cancellation.
8. For each reservation, A list of alternate contacts, separate from the person who makes the reservation, who are the people to be contacted to request that the reservation be changed or revoked.
9. A list of world wide web URLs associated with each reservation. (For example, these can be used to link users to home pages associated with the event for which the room is reserved.)

Documentation Requirements

1. **First-level help** - Overview of which functionality the program has, and how to accomplish each.
2. **Low-level help** - Explanation of each part of the user interface.

Use Cases

From the Requirements, we broke the system into the following Use Cases:

Visitor

1. View today's schedule
2. View schedule for a specific day
3. View information for a specific room
4. View room(s) for specific information
5. View schedule for a specific week
6. View information specific reservation for a user-defined time range
 - identified by purpose
 - identified by the "maker" of the reservation

Account User

7. Reserve a room
8. Reserve a room for a recurring event
9. Cancel a reservation
10. Cancel part of a recurring reservation
11. Cancel the whole recurring reservation
12. Make colloquium information available
13. Associate colloquium information with a reservation
14. Reserve a room for colloquium
15. Modify a reservation

Note: Users with more than one person on their authorization lists may perform the actions above on behalf of any person on their authorization list

Administrator

16. Change a user's authorization list
17. Change a user's authorization level
18. Specify a room's attributes
19. Make a room available for reservations
20. Make a room unavailable for reservations
21. Cancel any reservation
22. Modify any reservation
23. Purge old records (remove "old" reservations)
24. Change the set of room attributes

User Case Diagram

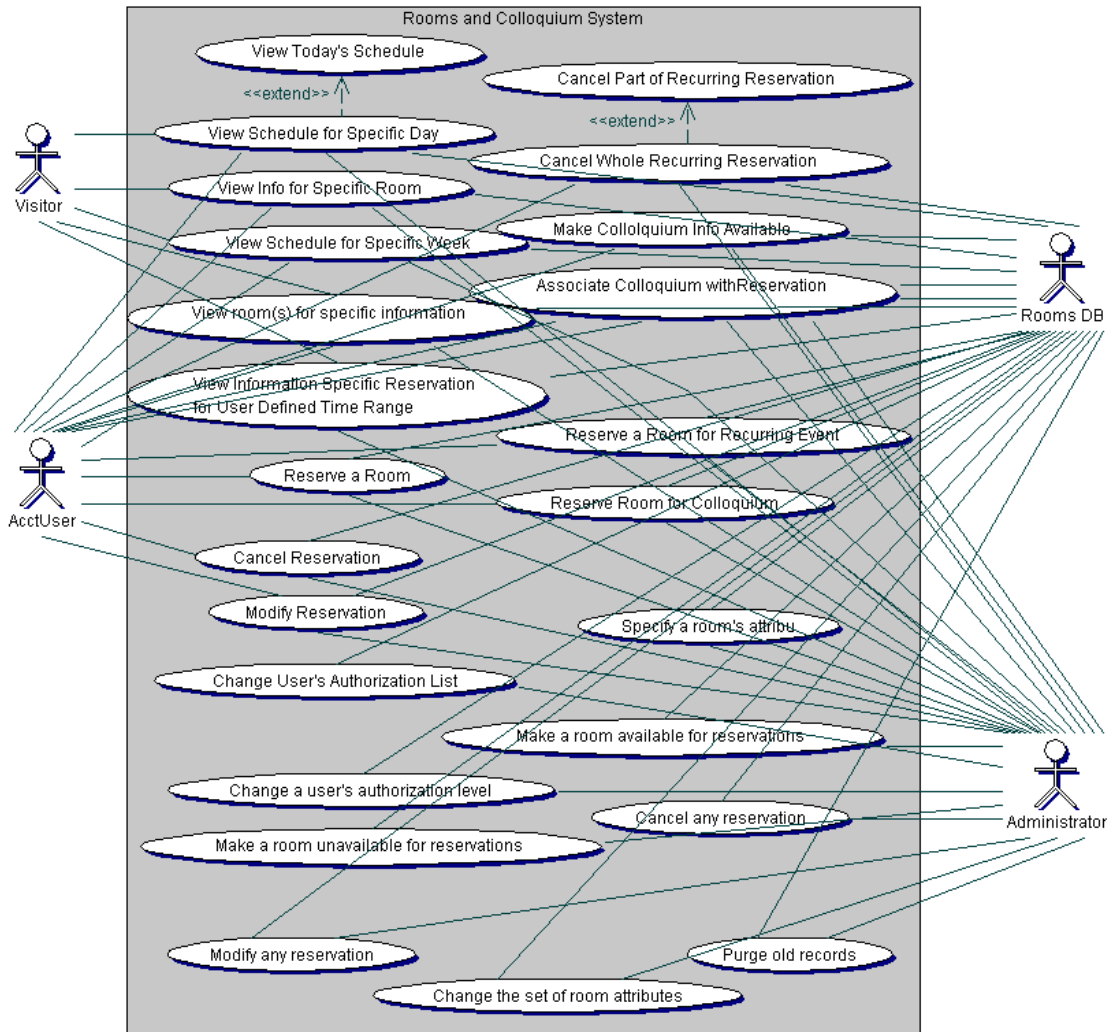


Figure 1 Use Case Diagram 1

System Design

This is the design of the system as determined from the customer requirements, budget, time availability, and accessible resources.

Architecture Decisions

Front-End

We discussed using either straight HTML or an Applet. After discussing this with Dave Parter, we decided to focus on straight HTML as our primary front-end. We have kept our HTML in line with the University mandated accessibility guidelines.

Middle-Tier

We discussed using a stand-alone Java application and using a Java servlet. Since our mentors required a web front-end, we chose to leverage the work already done by the Tomcat project to use Servlets through the Apache web server.

Back-End

We discussed using a custom text file, Java serialization, and a stand-alone relational database. We chose to use the PostgreSQL relational database because the CSL has a database server already in service that we can use. This reduced the burden of coding (to an extent) while it allowed us to use industry standard SQL. Using a standard database also allowed us to migrate to a different system with less pain than other choices.

Logical Architecture Design Diagram

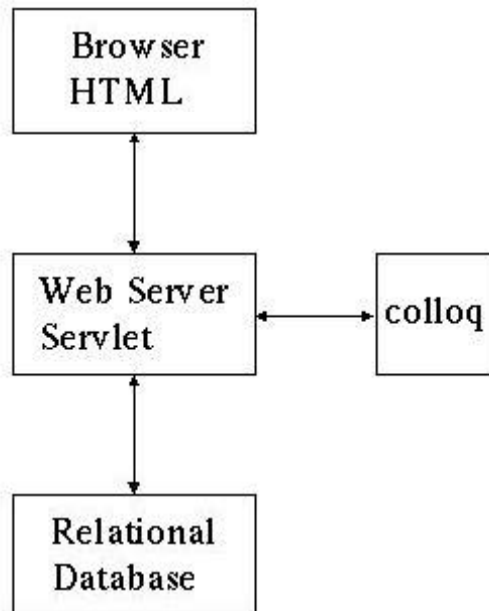


Figure 2 Logical Architecture Design

Targets

These are the targets for the system.

Client Target:

- Intel x86 machine
- RedHat Linux (6.1) OS
- Lynx web browser (HTML 3.2 compliant)

Server Target:

- Intel x86 machine
- RedHat Linux (6.1) OS
- Apache web server
- Servlet engine (JSDK 2.0 compliant)

Relational Database Target:

- Intel x86 machine
- RedHat Linux (6.1) OS
- PostgreSQL 7.1

Physical Architecture Design Diagram

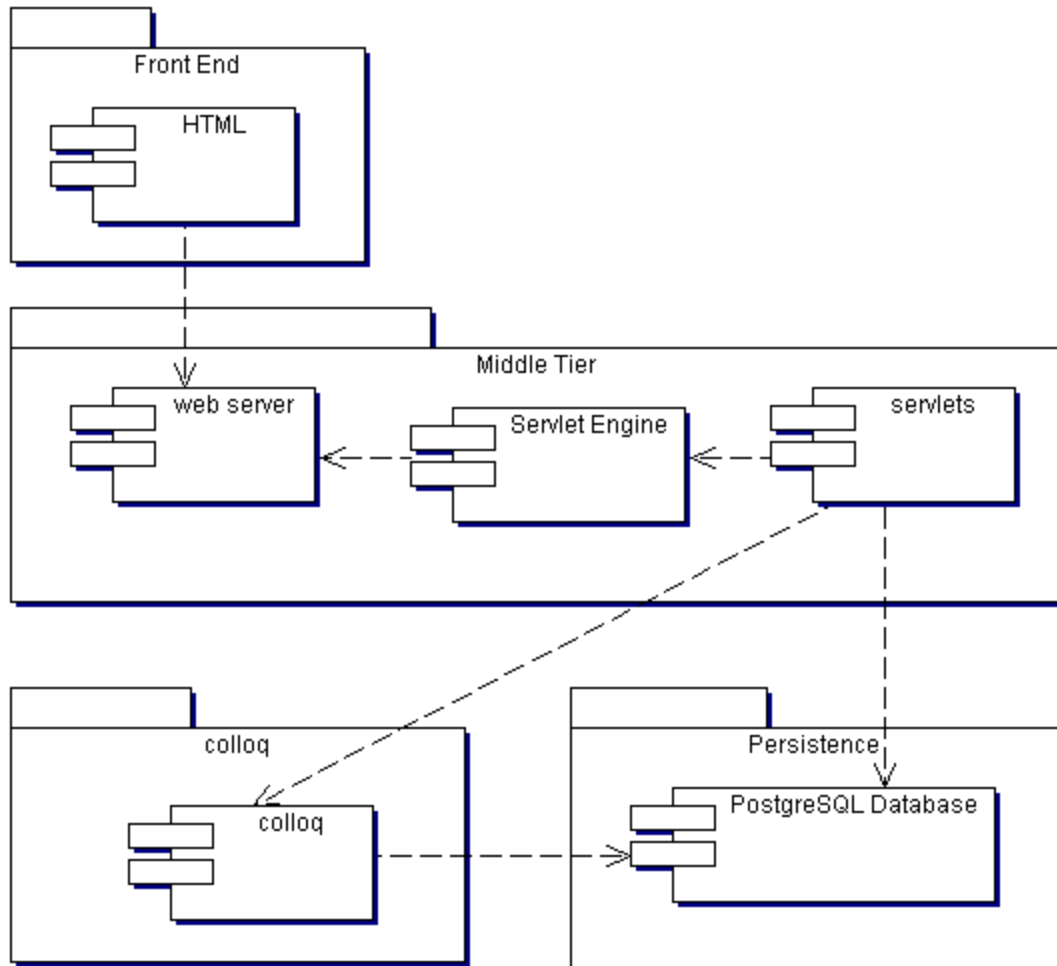


Figure 3 Physical Architecture Design

Physical Class Diagram

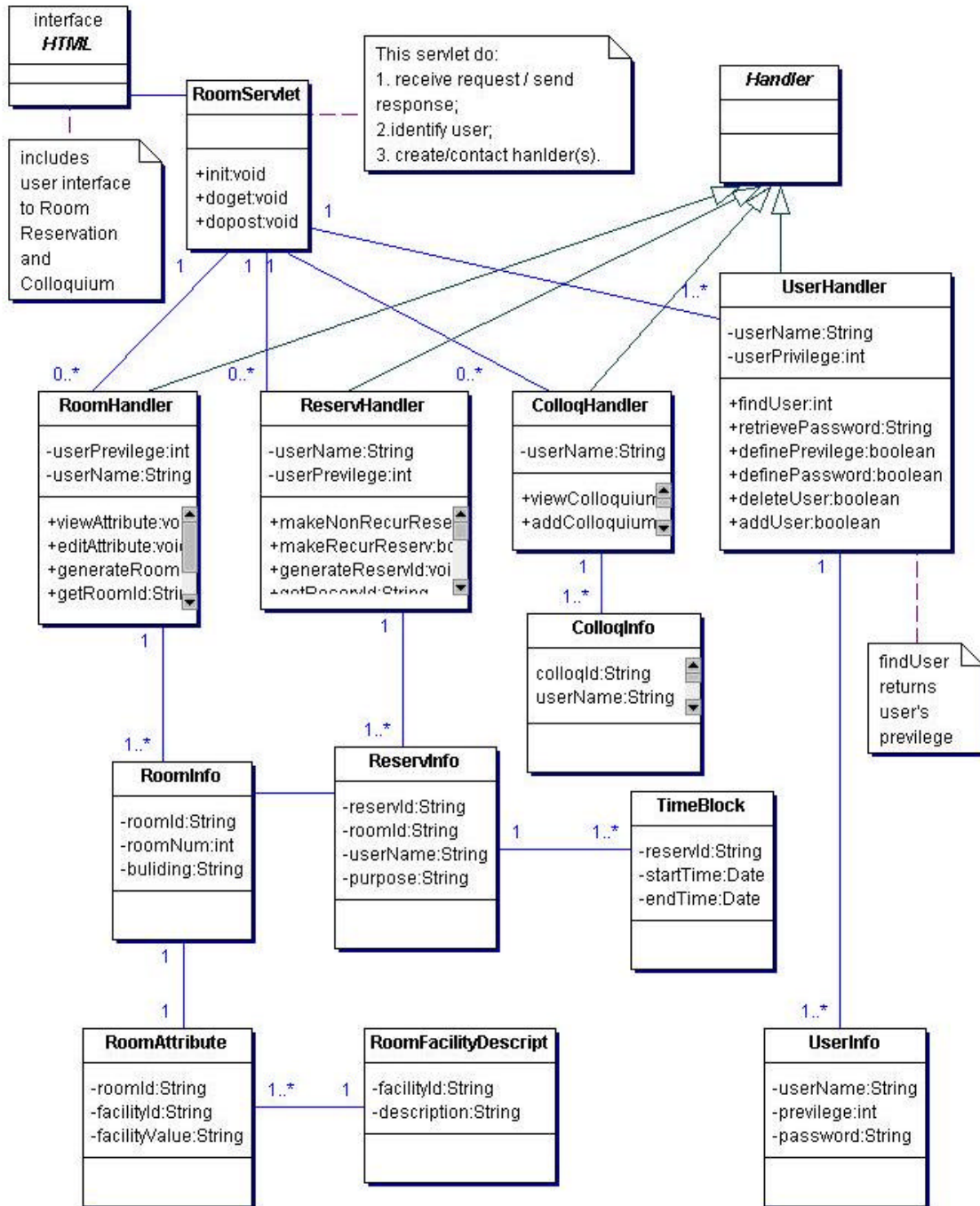


Figure 4 General Physical Class Diagram

Sequence Diagrams

These are the Sequence Diagrams created from each Use Case in relation to the Class Diagram.

View today's schedule, View schedule for a specific day

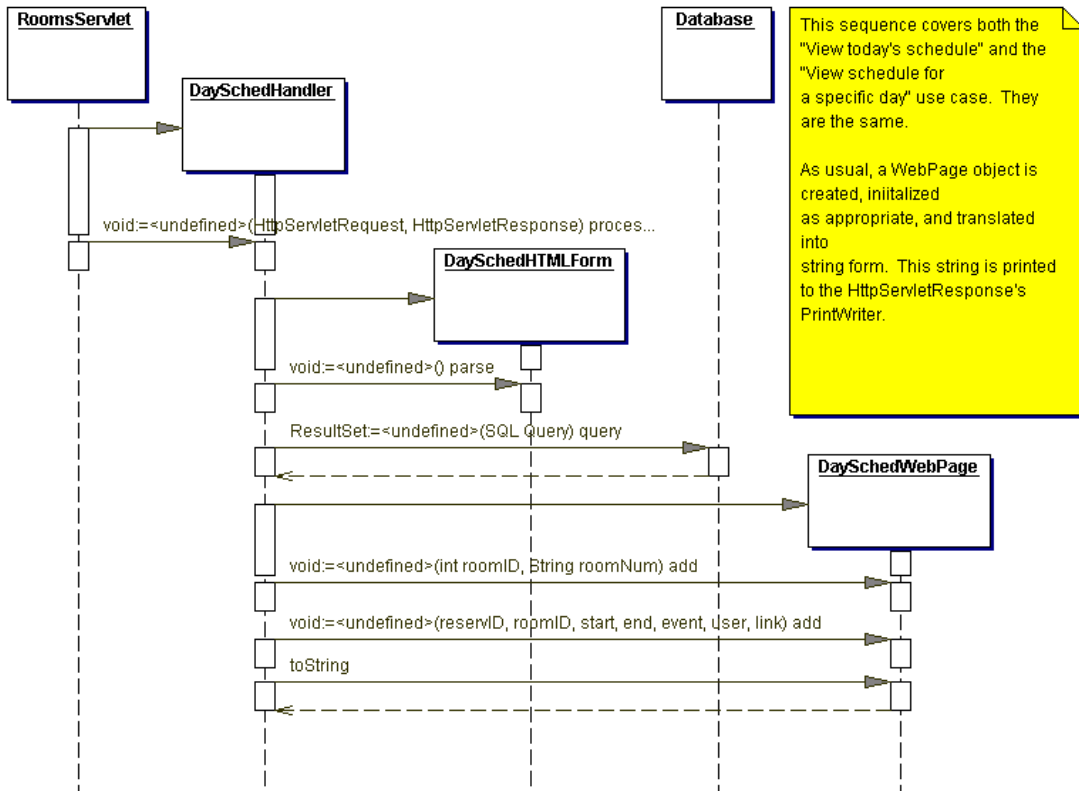


Figure 5 Sequence Diagram, View Today's Schedule, View Schedule for a Specific Day

View room(s) for specific information

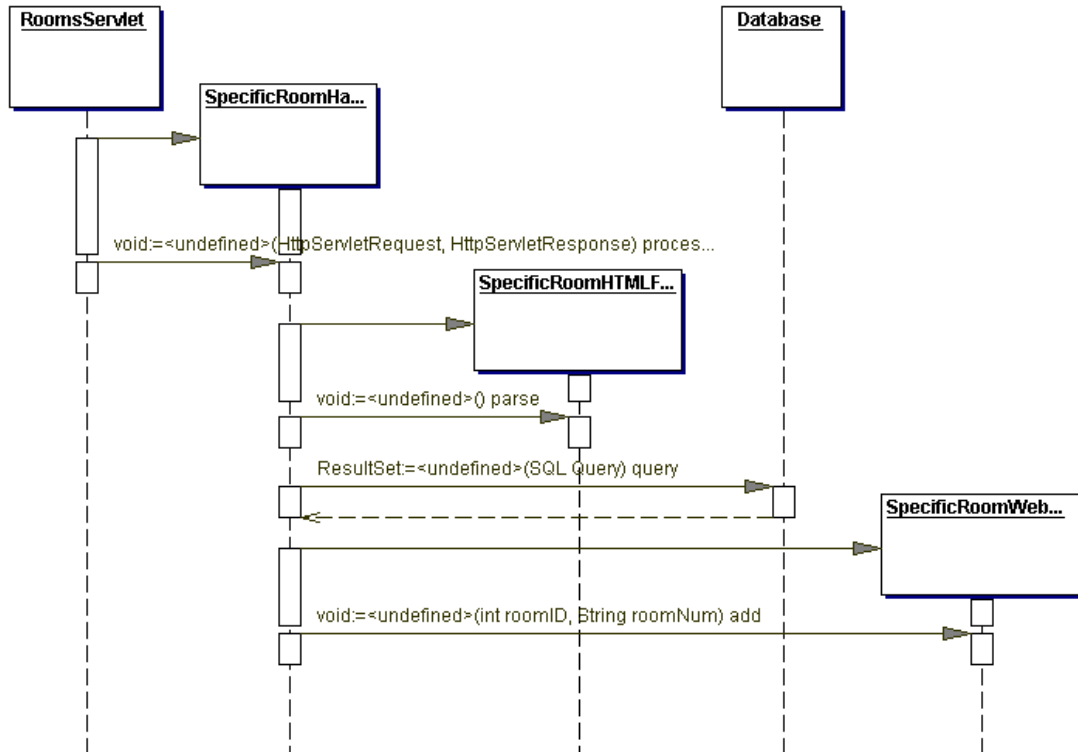


Figure 6 Sequence Diagram: View room(s) for specific information

View schedule for a specific week

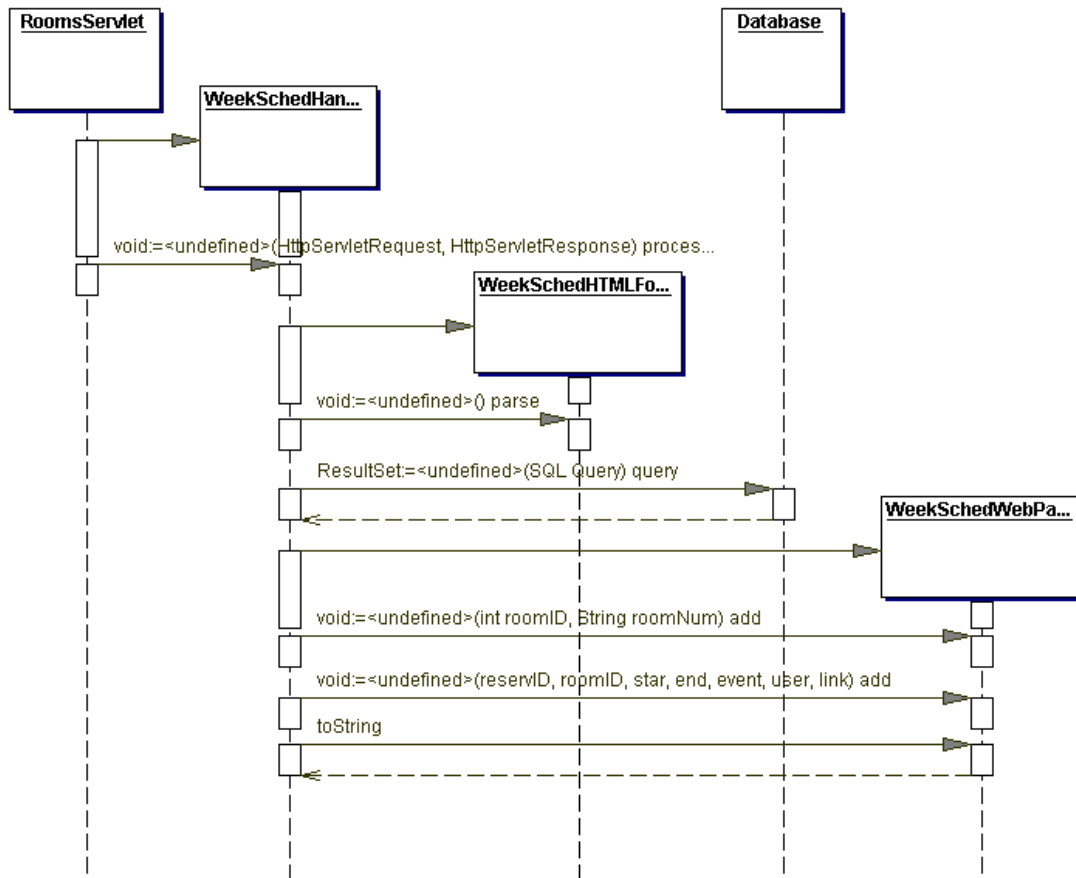


Figure 7 Sequence Diagram: View schedule for a specific week

View information specific reservation for a user-defined time range, identified by purpose, identified by the "maker" of the reservation

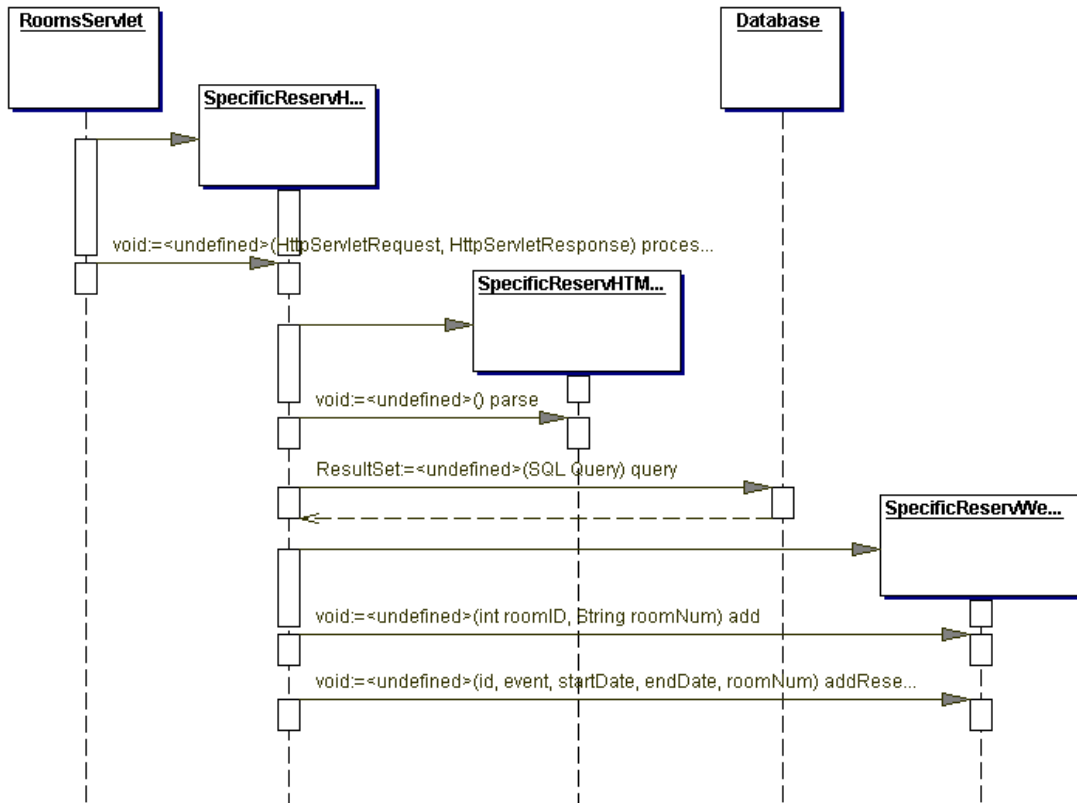


Figure 8 Sequence Diagram: View information specific reservation for a user-defined time range, identified by purpose, identified by the "maker" of the reservation

Reserve a room, Reserve a room for a recurring event

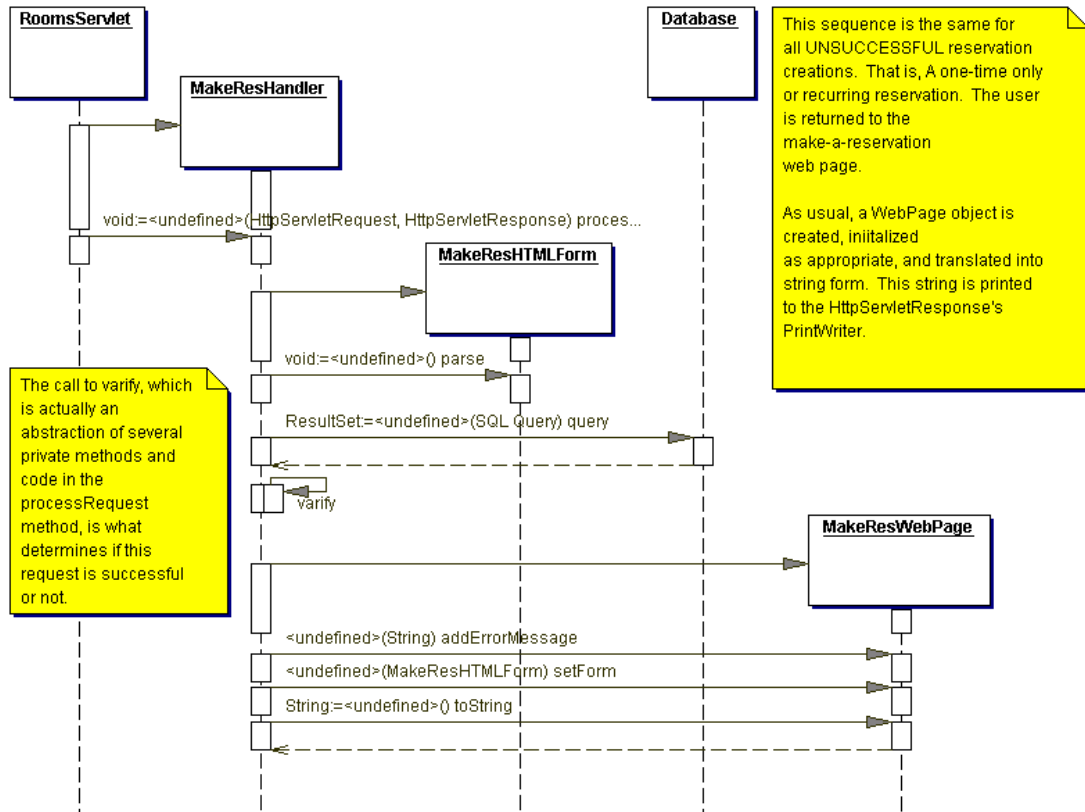


Figure 9 Sequence Diagram: Reserve a Room, Reserve a Room for a Recurring Event, Cancel a Reservation for UnSuccessful.

The Rooms and Colloquium System

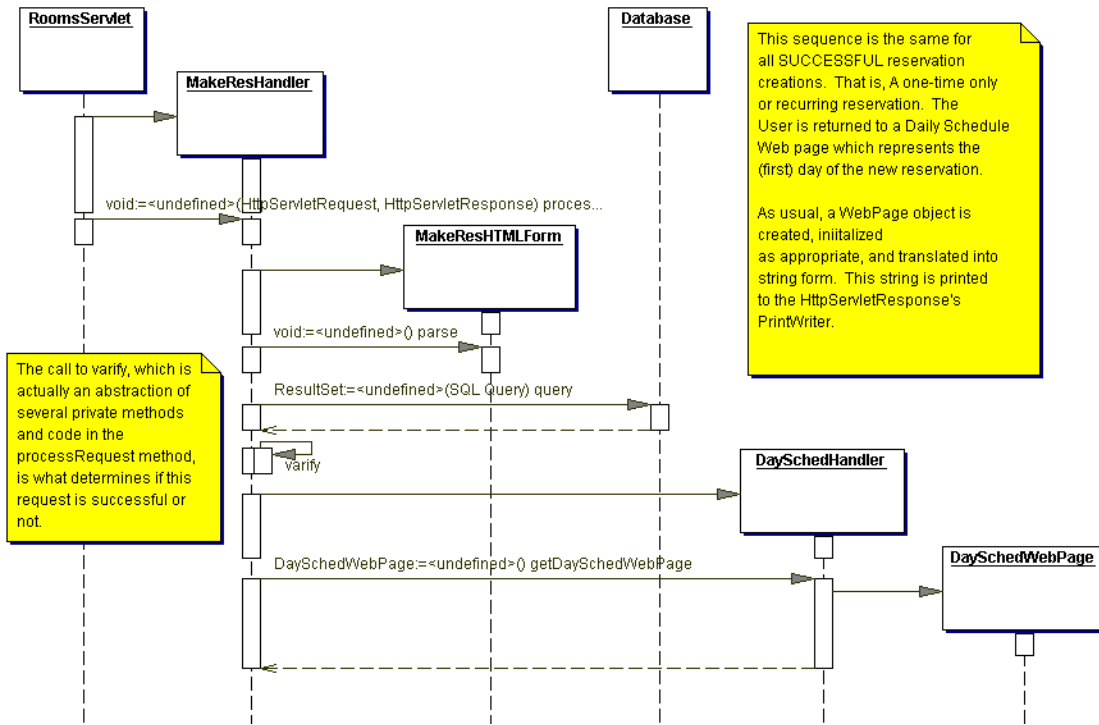


Figure 10 Figure 6 Sequence Diagram: Reserve a Room, Reserve a Room for a Recurring Event, Cancel a Reservation for Successful.

Make colloqium information available

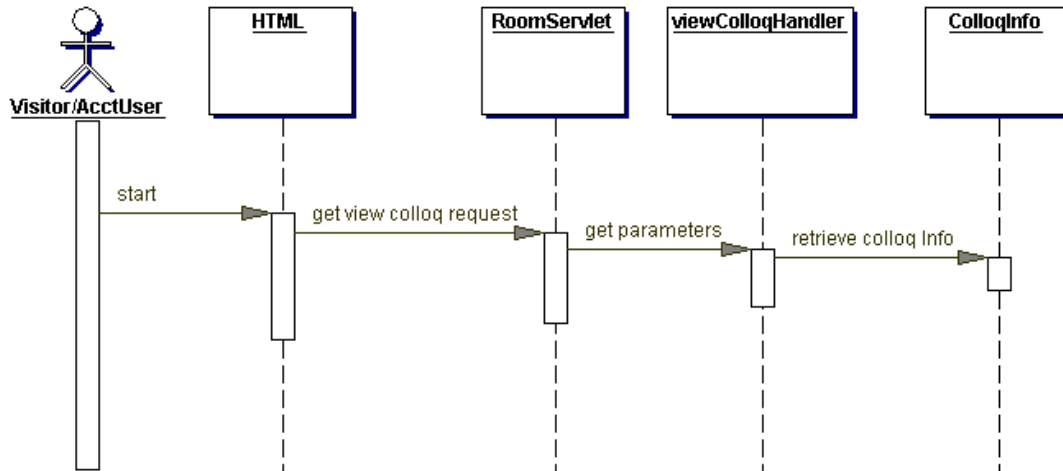


Figure 11 Sequence Diagram Make colloqium information available

Asociate colloqium information with a reservation

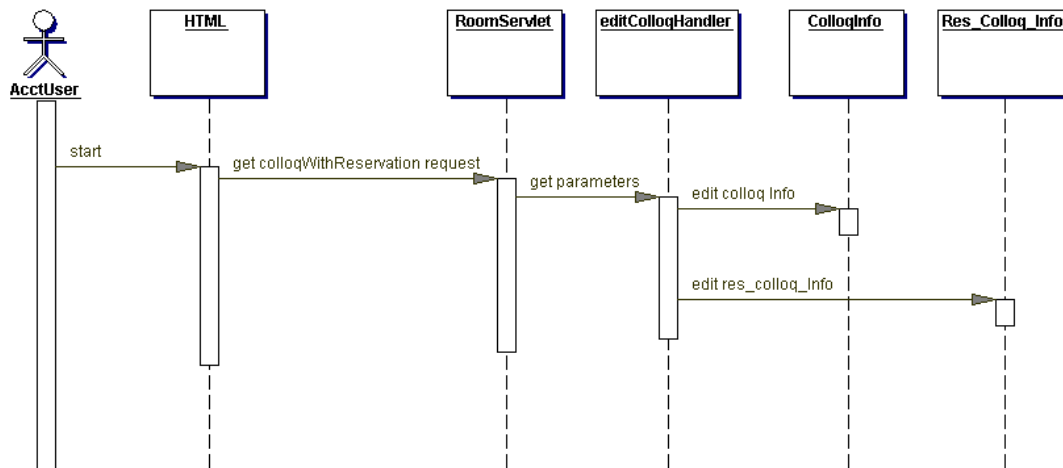


Figure 12 Sequence Diagram Asociate colloqium information with a reservation

Reserve a room for colloquium

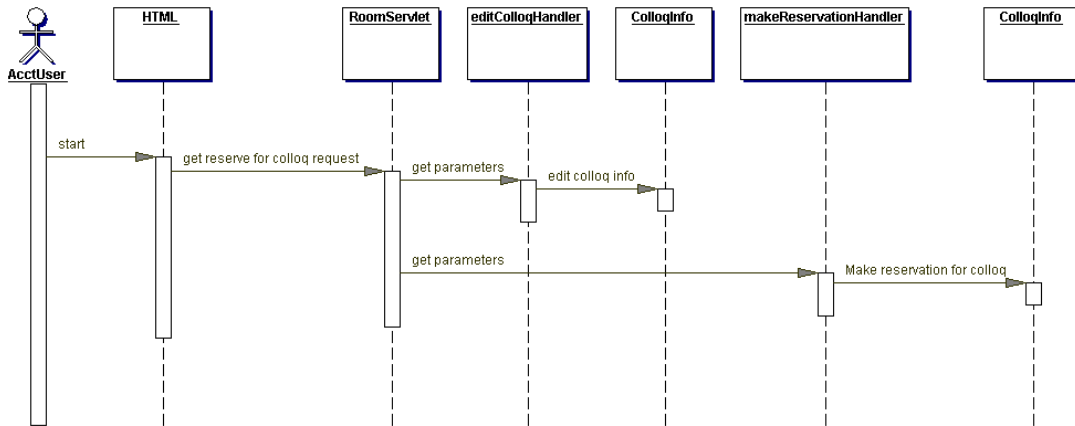


Figure 13 Sequence Diagram Reserve a room for colloquium

Modify a reservation

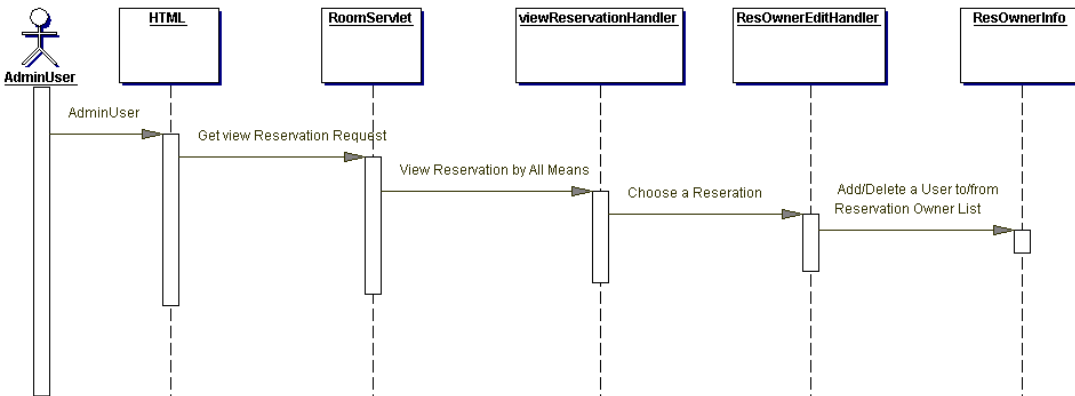


Figure 14 Sequence Diagram: Modify a reservation

Cancel a Reservation, Cancel part of a recurring reservation, Cancel the whole recurring reservation

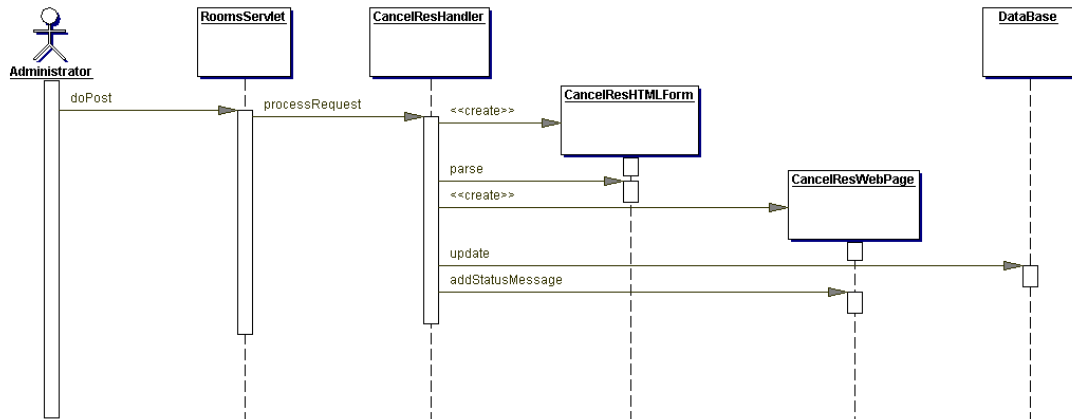


Figure 15 Sequence Diagram Cancel a Reservation, Cancel part of a recurring reservation, Cancel the whole recurring reservation

Change a user's authorization list

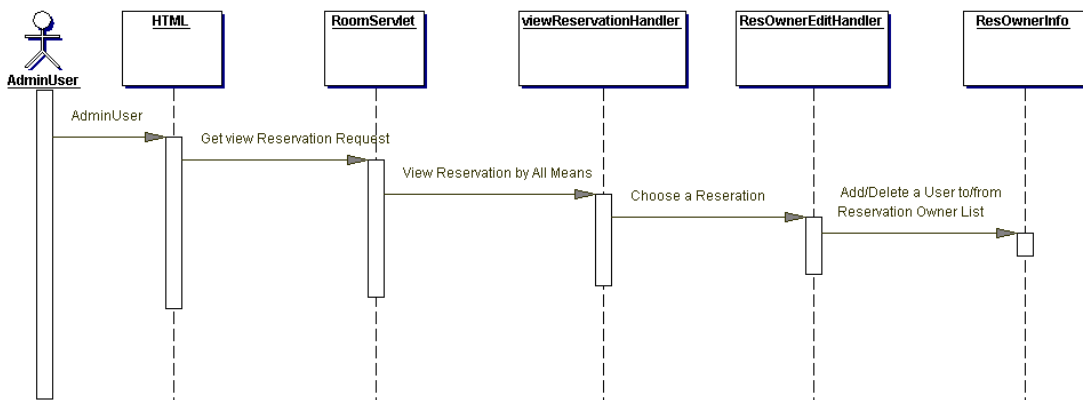


Figure 16 Sequence Diagram Change a user's authorization list

Change a user's authorization level

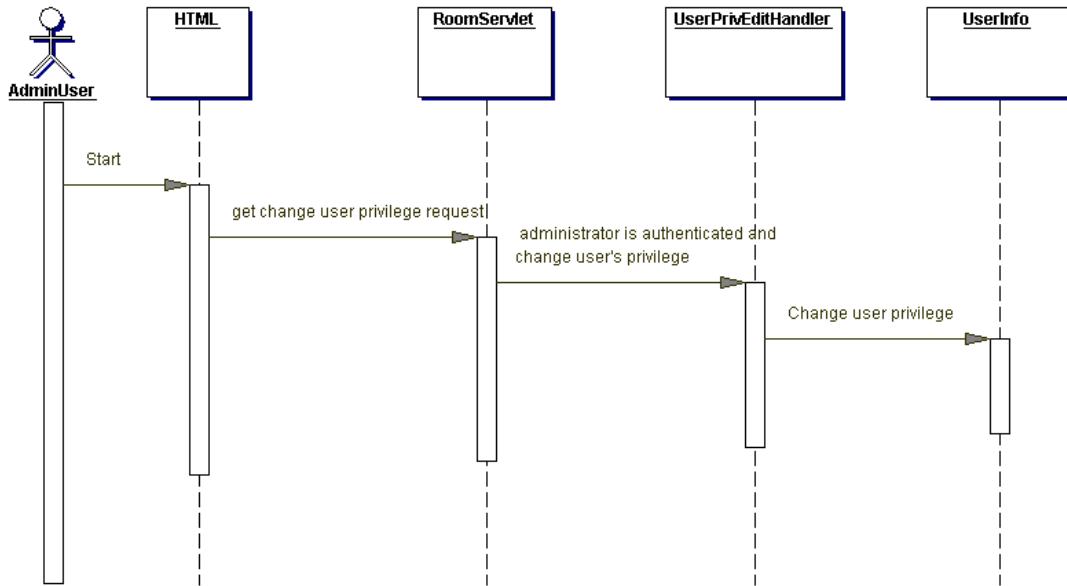


Figure 17 Sequence Diagram Change a user's authorization level

Make a room available for reservations

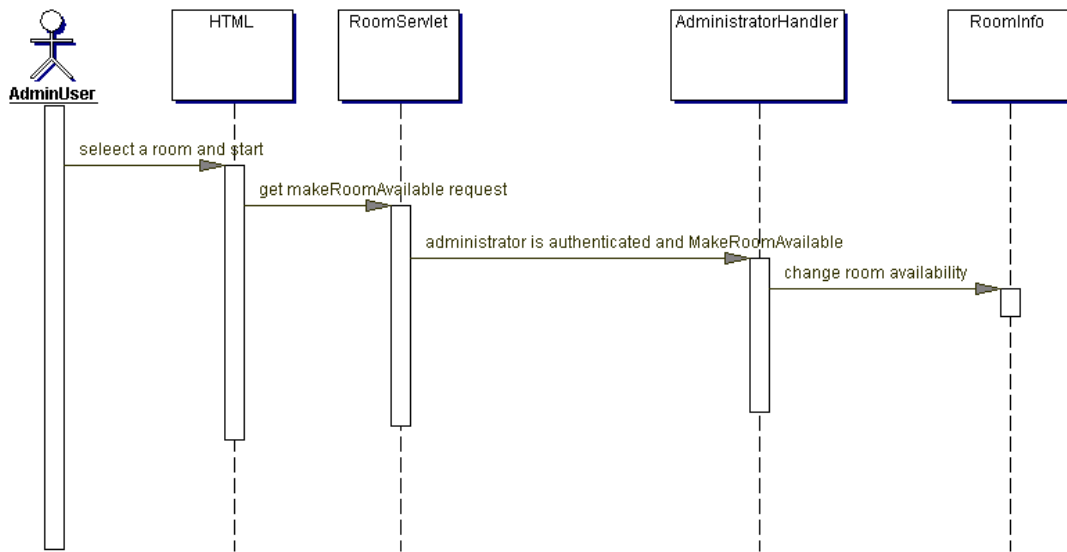


Figure 18 Sequence Diagram Make a room available for reservations

Make a room unavailable for reservations

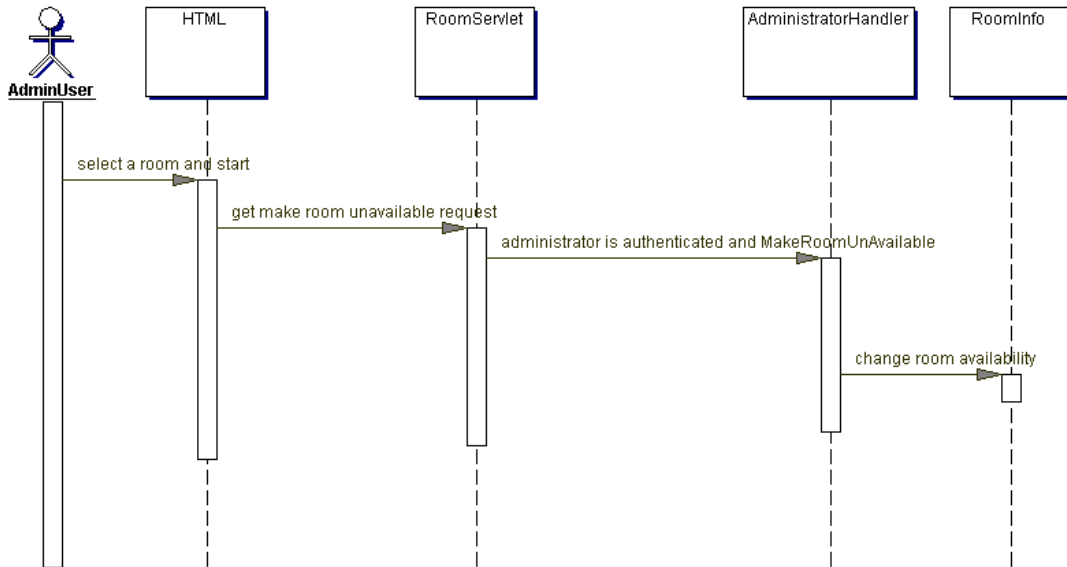


Figure 19 Sequence Diagram Make a room unavailable for reservations

Cancel any reservation

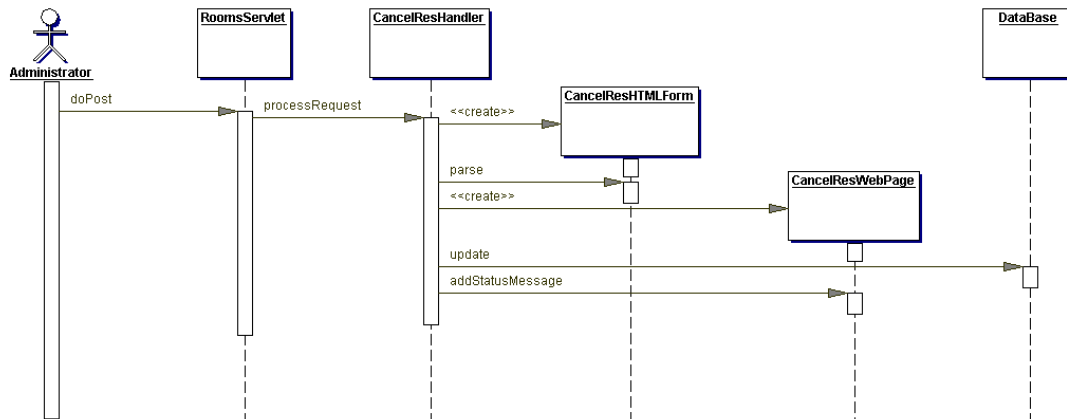


Figure 20 Sequence Diagram: Cancel any reservation

Modify any reservation

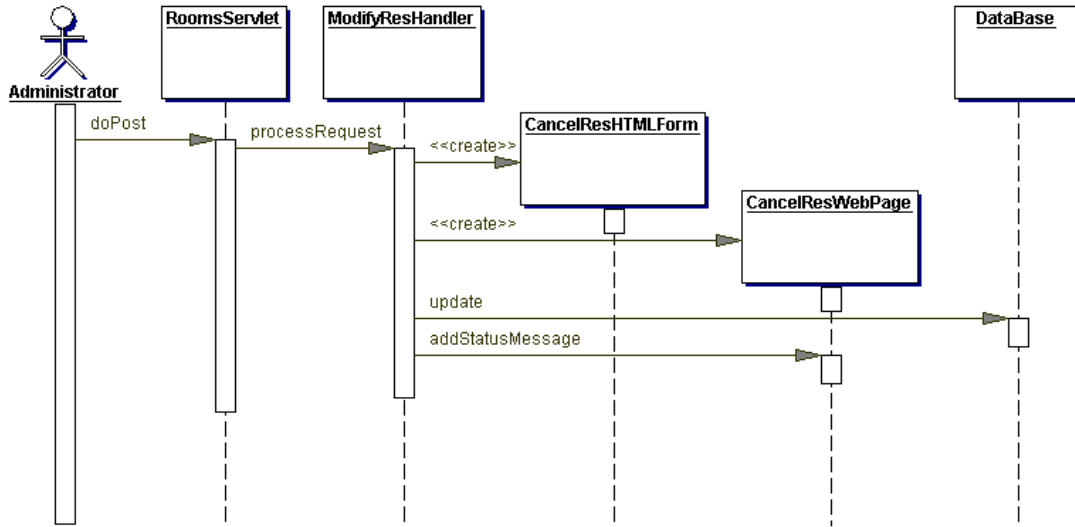


Figure 21 Sequence Diagram: Modify any reservation

Purge old records (remove "old" reservations)

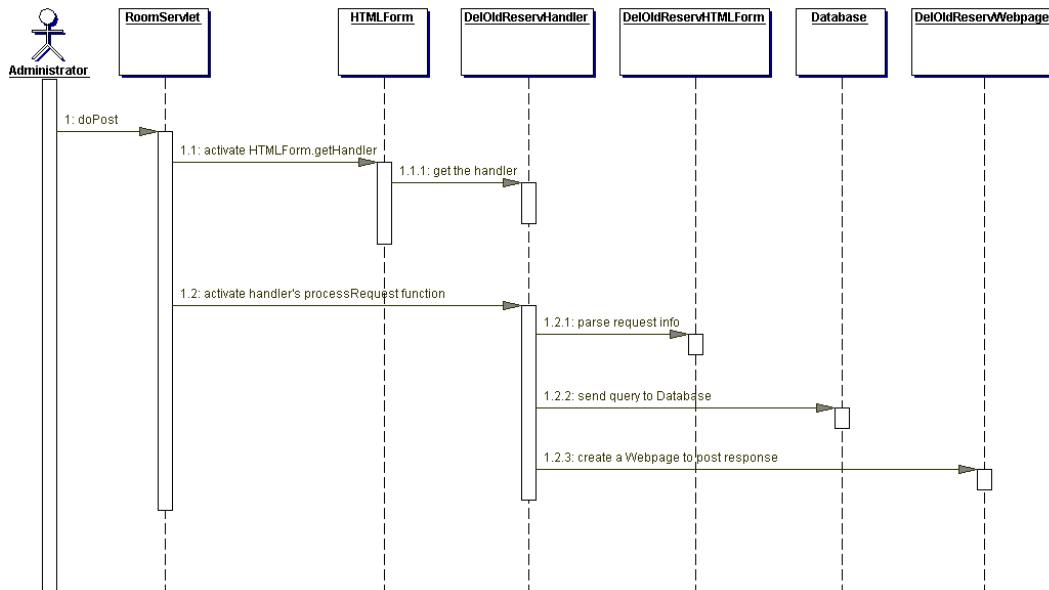


Figure 22 Sequence Diagram Purge old records (remove "old" reservations)

Change the set of room attributes

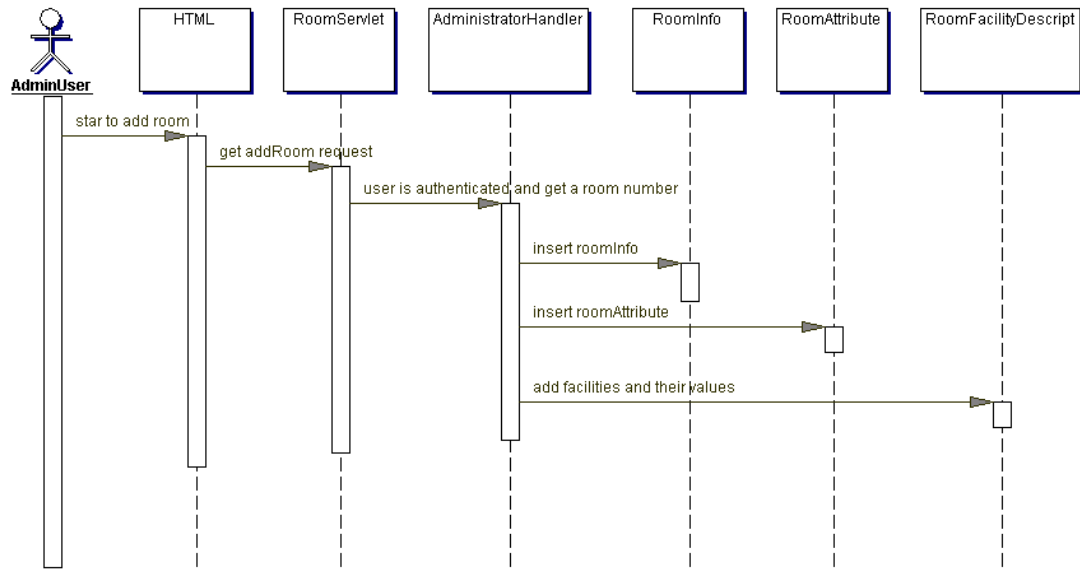
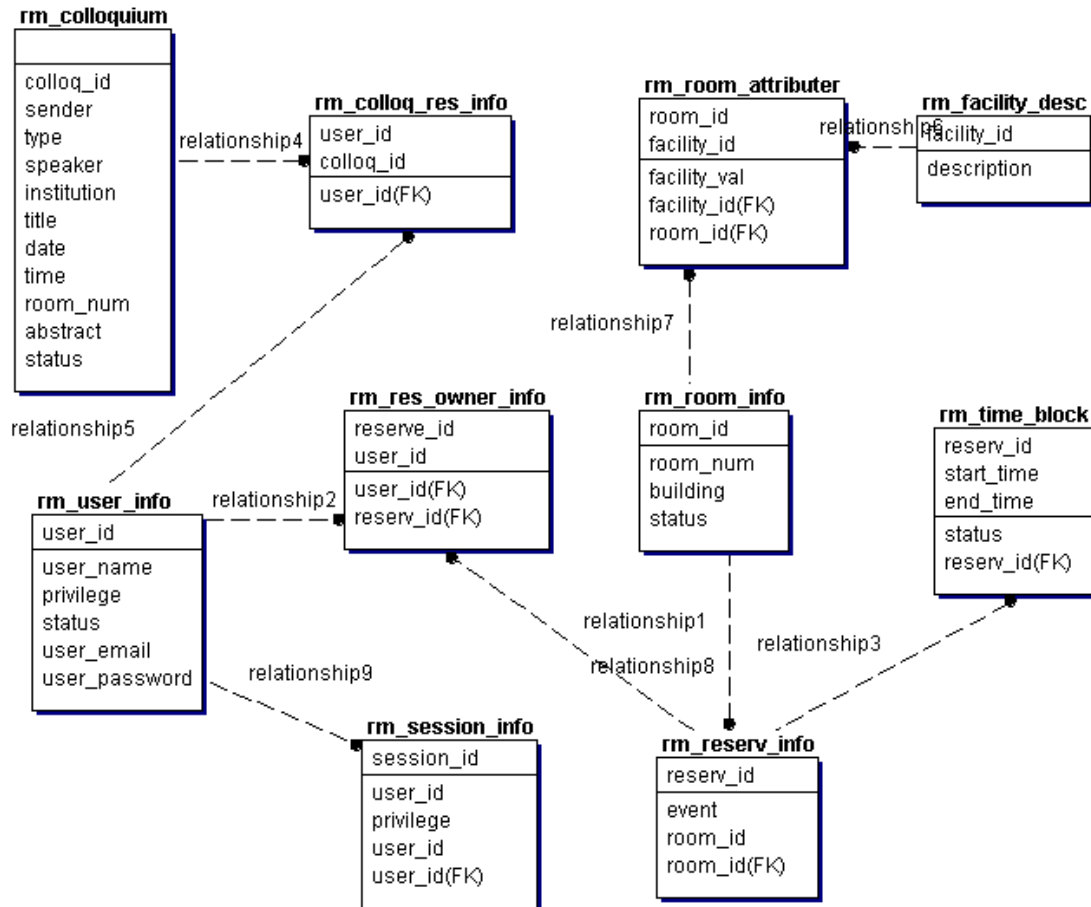


Figure 23 Sequence Diagram Change the set of room attributes

Database Schema

This is the Database Schema diagram for the persistent layer of the system.



User Interface Screen Shot

The screenshot shows a web browser window displaying the 'Room Reservation System' for the University of Wisconsin, Madison. The page features a header with the department name and navigation links. Below the header is a navigation menu with options like 'Log In', 'Today', 'This Week', 'Reserve A Room', 'Colloquium', 'Find A Room', 'Find A Reservation', 'Help', and 'Admin'. The main content area is titled 'Room Reservations for Thursday, December 20, 2001 (text version)'. It contains a table with columns for time slots (8:00 AM to 6:00 PM) and rows for various room numbers (e.g., CS1325, CS2310, CS2312, CS2340, CS3215, CS332, CS3331, CS4444, CS4567, CS3331, CS3332, CS333, CS7331, CS777, CS999, and CS7700). The table is currently empty. Below the table is a 'View Daily Reservation Schedule' button and a date selection tool set to 'December 20th 2001'.

	8:00 AM	9:00 AM	10:00 AM	11:00 AM	12:00 PM	1:00 PM	2:00 PM	3:00 PM	4:00 PM	5:00 PM	6:00 PM
CS1325											
CS2310											
CS2312											
CS2340											
CS3215											
CS332											
CS3331											
CS4444											
CS4567											
CS3331											
CS3332											
CS333											
CS7331											
CS777											
CS999											
CS7700											

Test Plan

Code Review and System Testing

1. Introduction to Code Review
2. Introduction to System Testing
3. Test Cases for Room Reservation System: Login/User Authentication
4. Test Cases for Room Reservation System: View Reservation
5. Test Cases for Room Reservation System: View Room(s)
6. Test Cases for Room Reservation System: Make/Edit a Reservation
7. Test Cases for Room Reservation System: Colloquium
8. Test Cases for Room Reservation System: Administration
9. Test Cases for Room Reservation System: Security Requirements
10. Room Reservation System Testing Report Form
11. Description of the system testing report form

Introduction to Code Review

(from Shari Lawrence Pflefer, *Software Engineering: Theory and Practice*, Prentice Hall, 2001)

It is helpful to ask an objective group of experts to review both of the code and documentation for misunderstandings, inconsistencies, and other faults. There are two types of code review: a walkthrough and an inspection.

Code Walkthrough

In a walkthrough, you present your code and accompanying documentation to the review team, and the team comments on their correctness. You only need to find faults in the codes, but not necessary fix them.

Code Inspection

The code inspection is similar to a code walkthrough but is more formal. The review team checks the code and documentation against a prepared list of concerns. The team can review algorithms and computations for their correctness and efficiency. The team may even estimate the code's performance characteristics in terms of memory usage or processing speed, in preparation for assessing compliance with performance requirements.

Steps

1. The team meets as a group for an overview of the code and a description of the inspection of the inspection goals.
2. Team members prepare individually for a second group meeting. Each inspector studies the code and its related documents, noting faults found.
3. In a group meeting, team members report what they have found, recording additional faults discovered in the process of discussing individual's findings.

4. For the room reservation system, the code walkthrough is carried out.

Introduction to System Testing

(from Shari Lawrence Pflefer, Software Engineering: Theory and Practice, Prentice Hall, 2001)

After the unit testing and integration testing is finished, system testing is carried out to test the integrated system.

Steps in System Testing

1. Function testing: checking that the integrated system performs its functions as specified in the requirements. This step is run by the developers.
2. Performance testing: comparing the integrated components with the nonfunctional system requirements. The requirements include security, accuracy, speed and reliability. This step is run by the developers.
3. Acceptance testing: assuring that the system requested by customers is the system that was built for the customers.
4. Installation testing: allowing user to exercise system functions and to document additional problems that result from being at the actual environment.

Test documentation

1. Test plan: describing the system itself and the plan for exercising all functions and characteristics
2. Testing specification and evaluation: detailing each test and defining the criteria for evaluating each feature addressed by the test
3. Test description: presenting the test data and procedures for individual tests.
4. Test analysis report: describing the results of each test.

System Testing Reports

Tester Name: Ming Li	Testing Date: 12/12/01
Name of the Module: Create Reservation	
Function of the Module: Make Reservation	
Testing Procedure: Click the link for "Create Reservation" and fill in the form And hit the submit button	
Input Data: username, Event description, Room, Date, start time, end time	
Output: Schedule table of that date, each color bar has a link with the specified reservation	
Test Result: Ok	
Security Checking: Ok, you must be a user at least	
Performance Evaluation: OK. The start time only have :00, :15, :30, :45 choices, if some people only need from 8:55, that cannot be satisfied, this is only minor point.	
User Interface Evaluation: OK. However, if there is too many different event description, there will be too many color bar links and they will make the output messy. Also right now, the table of schedule seems too big and only shows from 8am to 6pm?	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Room Reservation System Testing Report Form

Tester Name: Ming Li	Testing Date: 12/12/01
Name of the Module: Create Reservation	
Function of the Module: Make Recurring Reservation	
Testing Procedure: Click the link for "Create Reservation" and fill in the form And hit the submit button	
Input Data: username, Event description, Room, Date, start time, end time Recurrent end date, recurrent pattern	
Output: Schedule table of that date, each color bar has a link with the specified reservation	
Test Result: Ok	
Security Checking: Ok, you must be a user at least	
Performance Evaluation: OK. For the recurrent pattern, it might confuse people with both frequency of recurrence (in days) and recurrent applies on weekdays. What if people want to make reservation every Monday and said frequency of recurrence in 8 days	
User Interface Evaluation: OK. However, if there is too many different event description, there will be too many color bar links and they will make the output messy. Also right now, the table of schedule seems too big and only shows from 8am to 6pm?	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Room Reservation System Testing Report Form

Tester Name: Ming Li	Testing Date: 12/12/01
Name of the Module: Create Reservation	
Function of the Module: Modify(update) reservation	
Testing Procedure: Click the link in the daily schedule and fill in the form and hit the submit button	
Input Data: username, event description, room	
Output: Schedule table of that date, each color bar has a link with the specified reservation	
Test Result: Ok	
Security Checking: Not good enough. Someone else can update your reservation	
Performance Evaluation: OK.	
User Interface Evaluation: OK. However, if there is too many different event description, there will be too many color bar links and they will make the output messy. Also right now, the table of schedule seems too big and only shows from 8am to 6pm?	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments: However, security need to be improved	

Room Reservation System Testing Report Form

Tester Name: Ming Li	Testing Date: 12/12/01
Name of the Module: Create Reservation	
Function of the Module: Cancel a Reservation	
Testing Procedure: Click the link in daily schedule and fill in the form. And hit the submit button	
Input Data: no input Data, just hit the delete reservation button	
Output: Schedule table without the specified reservation, each color bar has a link with the specified reservation	
Test Result: Ok	
Security Checking: Not good enough. Someone else can delete your reservation	
Performance Evaluation: OK. Except the security problem	
User Interface Evaluation: OK. However, if there is too many different event description, there will be too many color bar links and they will make the output messy.	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments: Security need to be improved	

Room Reservation System Testing Report Form

Tester Name: Najun Zhou	Testing Date: 12/10/2001
Name of the Module: view today's schedule	
Function of the Module: view today's reservations	
Testing Procedure: login as a visitor, today's schedule is shown. For each reservation, click and get into a page for reservation view/edit. Also test text version.	
Input Data: N/A	
Output: all today's reservation.	
Test Result: correct.	
Security Checking: OK	
Performance Evaluation: good	
User Interface Evaluation: good	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Note: Please read the Description of the System Testing Report Form.

Room Reservation System Testing Report Form

Tester Name: Naijun Zhou	Testing Date: 12/10/2001
Name of the Module: view schedule for a specific day	
Function of the Module: view all reservations in a specific day	
Testing Procedure: select a day, month and year to see the reservations of that date.	
Input Data: 12/18/2001	
Output: reservations for that day.	
Test Result: correct	
Security Checking: OK	
Performance Evaluation: OK	
User Interface Evaluation: OK	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Note: Please read the Description of the System Testing Report Form.

Room Reservation System Testing Report Form

Tester Name: Najun Zhou	Testing Date: 12/10/2001
Name of the Module: View schedule for this week	
Function of the Module: view all reservations of this week.	
Testing Procedure: click button of "This week's schedule" to see reservations	
Input Data: N/A	
Output: a table representing all reservations of this week	
Test Result: correct	
Security Checking: OK	
Performance Evaluation: OK	
User Interface Evaluation: OK. The label might be too small.	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Note: Please read the Description of the System Testing Report Form.

Room Reservation System Testing Report Form

Tester Name: Najun Zhou	Testing Date: 12/10/2001
Name of the Module: view schedule for a specific week	
Function of the Module: view reservation for a specific week	
Testing Procedure: select a date to view all reservations of the week containing the specified date.	
Input Data: December 19, 2001	
Output: reservations for 12/16/2001 to 12/22/2001	
Test Result: correct	
Security Checking: OK	
Performance Evaluation: OK	
User Interface Evaluation: OK	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Note: Please read the Description of the System Testing Report Form.

Room Reservation System Testing Report Form

Tester Name: Najun Zhou	Testing Date: 12/10/2001
Name of the Module: find a room	
Function of the Module: view information of all rooms	
Testing Procedure: click the button "Find A Room"	
Input Data: N/A	
Output: all rooms together with their facilities and facility values	
Test Result: correct	
Security Checking: OK	
Performance Evaluation: OK	
User Interface Evaluation: OK	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Note: Please read the Description of the System Testing Report Form.

Room Reservation/Colloquium System Testing Report Form

Tester Name: Hongwei Zhu	Testing Date: Dec., 11, 2001
Name of the Module: AddUserHandler	
Function of the Module: Add a new user	
Testing Procedure: 1. Click "Need an Account or Lost Password Click Here" 2. Fill the username, passwd, e-mail and click "Submit query"	
Input Data: user: hzhu; passwd: zhw917; e-mail: hongweizhu@students.wisc.edu	
Output: No output. User added successfully.	
Test Result: OK	
Security Checking: OK. Currently, user is not restricted to CS people.	
Performance Evaluation: Good.	
User Interface Evaluation: Nice.	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Note: Please read the Description of the System Testing Report Form.

Room Reservation/Colloquium System Testing Report Form

Tester Name: Hongwei Zhu	Testing Date: Dec., 11, 2001
Name of the Module: ChangePasswordHandler	
Function of the Module: Change user's password	
Testing Procedure: 1. Click "To Change Your Password Click Here" 2. Fill the username, new passwd, e-mail and click "Submit query"	
Input Data: user name: hzhu; new passwd: zhw917a	
Output: No output. User's password changed successfully.	
Test Result: OK	
Security Checking: OK	
Performance Evaluation: Good.	
User Interface Evaluation: Nice.	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Note: Please read the Description of the System Testing Report Form.

Room Reservation/Colloquium System Testing Report Form

Tester Name: Hongwei Zhu	Testing Date: Dec., 11, 2001
Name of the Module: LoginHandler	
Function of the Module: Let user log in	
Testing Procedure: 1. input user name, passwd; 2. click "submit query"	
Input Data: user name : hzhu; passwd: zhw917a	
Output: No output. User's logged in successfully.	
Test Result: OK	
Security Checking: OK	
Performance Evaluation: Good.	
User Interface Evaluation: Nice.	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Note: Please read the Description of the System Testing Report Form.

Room Reservation/Colloquium System Testing Report Form

Tester Name: Hongwei Zhu	Testing Date: Dec., 11, 2001
Name of the Module: AddUserHandler, RM_PgAuthenticate	
Function of the Module: Generate a new password for user who forget password.	
Testing Procedure: 1. Click "Need an Account or Lost Password Click Here" 2. Fill the username, e-mail and click "Lost Password"	
Input Data: user name : hzhu	
Output: Get an e-mail which has a password consistent of random characters.	
Test Result: OK	
Security Checking: OK	
Performance Evaluation: Good.	
User Interface Evaluation: Nice.	
Does this module implement the requirements? (please underline one) <u>Yes</u> No If no, your comments:	

Note: Please read the Description of the System Testing Report Form.

Room Reservation/Colloquium System Testing Report Form

Tester Name: Minyi Xu	Testing Date: Dec., 11, 2001
Name of the Module: change user's authorization level	
Function of the Module: change a user's authorization level (visitor, user, administrator)	
Testing Procedure:	
Input Data: change a user's authorization level from 50 to 99.	
Output: that user's authorization level is now 99	
Test Result: OK	
Security Checking: OK, only administrators can access that administrator page and perform those functions.	
Performance Evaluation: OK	
User Interface Evaluation: Pleasant user interface	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Room Reservation/Colloquium System Testing Report Form

Tester Name: Minyi Xu	Testing Date: Dec., 11, 2001
Name of the Module: specify room's attributes	
Function of the Module: add a room's facilities and their values	
Testing Procedure:	
Input Data: specify a newly-added room attributes by adding 50 chairs, 12 desks, 3 projectors.	
Output: chairs, projectors and desks are added as expected	
Test Result: OK	
Security Checking: OK, only administrators can access that administrator page and perform those functions.	
Performance Evaluation:	
User Interface Evaluation: Pleasant user interface	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Room Reservation/Colloquium System Testing Report Form

Tester Name: Minyi Xu	Testing Date: Dec., 11, 2001
Name of the Module: make a room available for reservation	
Function of the Module: make a room available for reservation	
Testing Procedure:	
Input Data: Add a room 7070	
Output: I can make reservation using cs7070.	
Test Result: OK	
Security Checking: OK, only administrators can access that administrator page and perform those functions.	
Performance Evaluation: OK	
User Interface Evaluation: Pleasant user interface	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Room Reservation/Colloquium System Testing Report Form

Tester Name: Minyi Xu	Testing Date: Dec., 11, 2001
Name of the Module: make a room unavailable for reservation	
Function of the Module: make a room unavailable for reservation	
Testing Procedure:	
Input Data: make the room 0001 unavailable for reservation	
Output: I can not make reservation using 0001.	
Test Result: OK	
Security Checking: OK, only administrators can access that administrator page and perform those functions.	
Performance Evaluation: OK	
User Interface Evaluation: Pleasant user interface	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Room Reservation/Colloquium System Testing Report Form

Tester Name: Minyi Xu	Testing Date: Dec., 11, 2001
Name of the Module: change the set of room attributes	
Function of the Module: change a room's facility values	
Testing Procedure:	
Input Data: change the attributes of room cs7070: add a projector, delete 10 chairs, change the number of desks from 10 to 20	
Output: The attributes of cs7070 are changed as expected	
Test Result: OK	
Security Checking: OK, only administrators can access that administrator page and perform those functions.	
Performance Evaluation: OK	
User Interface Evaluation: Pleasant user interface	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Room Reservation/Colloquium System Testing Report Form

Tester Name: Minyi Xu	Testing Date: Dec., 11, 2001
Name of the Module: purge old records	
Function of the Module: delete old reservations from the database	
Testing Procedure:	
Input Data: purge all reservation records before 02/01/2001	
Output: All reservation records before 02/01/2001 are removed.	
Test Result: OK	
Security Checking: OK, only administrators can access that administrator page and perform those functions.	
Performance Evaluation: OK	
User Interface Evaluation: Pleasant user interface	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Room Reservation/Colloquium System Testing Report Form

Tester Name: Minyi	Testing Date: Dec., 11, 2001
Name of the Module: cancel any reservation: cancel a regular reservation	
Function of the Module: cancel a regular reservation	
Testing Procedure:	
Input Data: cancel a reservation made on 12/01/2001	
Output: that non-recurring reservation is cancelled	
Test Result: OK	
Security Checking: OK, only administrators can access that administrator page and perform those functions.	
Performance Evaluation: OK	
User Interface Evaluation: Pleasant user interface	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Room Reservation/Colloquium System Testing Report Form

Tester Name: Minyi	Testing Date: Dec., 11, 2001
Name of the Module: cancel a recurring reservation	
Function of the Module: cancel a recurring reservation	
Testing Procedure:	
Input Data: 12/01/20001 to 20/01/2001 (once every two days)	
Output: that recurring reservation is cancelled	
Test Result: OK	
Security Checking: OK, only administrators can access that administrator page and perform those functions.	
Performance Evaluation: OK	
User Interface Evaluation: Pleasant user interface	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Room Reservation/Colloquium System Testing Report Form

Tester Name: Minyi	Testing Date: Dec., 11, 2001
Name of the Module: cancel a part of recurring reservation	
Function of the Module: cancel some times in a recurring reservation	
Testing Procedure:	
Input Data: recurring time: 12/01/2001-- 12/20/2001), canceled time: 12/10/2001 -- 12/20/2001	
Output: that part of part of recurring reservation is cancelled	
Test Result: OK	
Security Checking: OK, only administrators can access that administrator page and perform those functions.	
Performance Evaluation: OK	
User Interface Evaluation: Pleasant user interface	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Room Reservation/Colloquium System Testing Report Form

Tester Name: Minyi	Testing Date: Dec., 11, 2001
Name of the Module: cancel a day of a recurring reservation	
Function of the Module: cancel one reservation in a date from a recurring reservation	
Testing Procedure:	
Input Data: for reservation of 12/01/2001-- 12/20/2001, cancel 12/10/01	
Output: reservation on 12/10/01 among the recurring reservations is cancelled	
Test Result: OK	
Security Checking: OK, only administrators can access that administrator page and perform those functions.	
Performance Evaluation: OK	
User Interface Evaluation: Pleasant user interface	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Room Reservation/Colloquium System Testing Report Form

Tester Name: Minyi	Testing Date: Dec., 11, 2001
Name of the Module: cancel one recurring pattern of recurring reservation, e.g. for 12/01/2001-- 12/20/2001, every Tuesday, Thursday, to cancel the Tuesdays reservations	
Function of the Module: cancel a day of a recurring reservation	
Testing Procedure:	
Input Data: for 12/01/2001-- 12/20/2001, every Tuesday, Thursday, cancel the Tuesdays reservations	
Output: all Tuesday reservations among the recurring reservations are cancelled	
Test Result: OK	
Security Checking: OK, only administrators can access that administrator page and perform those functions.	
Performance Evaluation: OK	
User Interface Evaluation: Pleasant user interface	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Room Reservation/Colloquium System Testing Report Form

Tester Name: Minyi	Testing Date: Dec., 11, 2001
Name of the Module: Modify a reservation	
Function of the Module: modify a reservation including the topic of a reservation, and its purpose, its room	
Testing Procedure:	
Input Data:	
Output: The topics, purpose and room number is changed as expected	
Test Result: OK	
Security Checking: OK, only administrators can access that administrator page and perform those functions.	
Performance Evaluation: OK	
User Interface Evaluation: Pleasant user interface	
Does this module implement the requirements? (please underline one) <u>Yes</u> No	
If no, your comments:	

Package Documentation

Class User Interface

Package	Class	Tree	Deprecated	Index	Help
-------------------------	-----------------------	----------------------	----------------------------	-----------------------	----------------------

Package edu.wisc.cs.rooms

Interface Summary	
Authenticate	
Handler	The basic method(s) for all Handler objects.

Class Summary	
AddColloquiaHandler	Handle requests to view the daily schedule.
AddColloquiaHandlerWithReserve	Handle requests to view the daily schedule.
AddColloquiaHTMLForm	An AddColloquiaHTMLForm is a HTMLForm with following parts: (1) date: a java.util.Date which represents the date for which this colloquia represents.
AddColloquiaHTMLFormWithReserve	An AddColloquiaHTMLForm is a HTMLForm with following parts: (1) date: a java.util.Date which represents the date for which this colloquia represents.
AddColloquiaWebPage	
AddColloquiaWebPageWithReserve	
AddRoomFacilityHandler	
AddRoomFacilityHTMLForm	
AddRoomHandler	
AddRoomHTMLForm	
AddUserHandler	Handle attempts to add a user to the system.
AddUserHTMLForm	An HTMLForm with the following parts:
AddUserWebPage	Web page for add a user to the ROOMS system.
AdministratorHandler	
AdministratorHTMLForm	A AdministratorHTMLForm is an HTMLForm with no attributes.
AdministratorWebPage	FirstViewWebPage.java Created on November 25, 2001, 3:51 PM
ChangeFacilityValueHandler	
ChangeFacilityValueHTMLForm	
ChangePasswordHandler	Handle attempts to add a user to the system.
ChangePasswordHTMLForm	An HTMLForm with the following parts:
ChangePasswordWebPage	Web page for add a user to the ROOMS system.
CollogHandler	Handle requests to view or edit a reservation.
CollogHTMLForm	An HTMLForm which allows the user to edit, delete (whole or in part) or view a certain reservation instance.
CollogWebPage	Displays the necessary form to make a reservation.
Database	Methods common to the Rooms Database
DaySchedHandler	Handle requests to view the daily schedule.
DaySchedHTMLForm	A DaySchedHTMLForm is a HTMLForm with two parts: (1) date: a java.util.Date which represents the date for which this schedule represents.
DaySchedRoom	Data structure used by DaySchedWebPage
DaySchedTimeBlock	Data structure used by DaySchedWebPage
DaySchedWebPage	A DaySchedWebPage displays the reservation schedule for a given day, and a given format (graphical, which is default, or text-only).
DebuggingHandler	Handle requests in a very limited way, echoing back request information.
DeleteFacilityHandler	
DeleteFacilityHTMLForm	
DelOldReservHandler	Handle requests to remove old reservations.
DelOldReservHTMLForm	A DelOldReservHTMLForm is a HTMLForm with two parts: (1) date: a java.util.Date which represents the date.

The Rooms and Colloquium System

DelOldReservWebPage	Displays the necessary form to remove old reservations.
EditRoomAttributeHandler	
EditRoomAttributeHTMLForm	
ErrorWebPage	
FindCollogHTMLForm	
HandlerFactory	A factory to create handler objects by name.
HTMLForm	A HTMLForm represents the < FORM > part of an HTML web page.
HTMLTag	Represents a tag in a HTML document.
HTMLText	HTMLText is a special kind of HTMLTag that is not associated with open, close tags; just text.
InputTable	An InputTable is an HTMLTag of type "table" for use with the HTMLForm's subclasses' input tables.
InputTableRow	An InputTableRow is an HTMLTag of type "tr" which represents a row in an InputTable which displays an input selection for an HTMLForm's input.
LoginHandler	
LoginHTMLForm	An HTMLForm with the following parts:
LoginWebPage	Web page for logging into system.
MakeResHandler	Handle requests to make a reservation.
MakeResHTMLForm	An HTMLForm with the following parts:
MakeResWebPage	Displays the necessary form to make a reservation.
MakeRoomAvailableHandler	
MakeRoomAvailableHTMLForm	
MakeRoomAvailableResultWebPage	
MakeRoomUnavailableHandler	
MakeRoomUnavailableHTMLForm	
MakeRoomUnavailableResultWebPage	
MD5	Implementation of RSA's MD5 hash generator
MD5State	Contains internal state of the MD5 class
NameValuePair	Represents a { name, value } pair to be used with HTMLTag objects.
ResInfoHandler	Handle requests to view or edit a reservation.
ResInfoHTMLForm	An HTMLForm which allows the user to edit, delete (whole or in part) or view a certain reservation instance.
ResInfoWebPage	Displays the necessary form to make a reservation.
ResOwnerAddHTMLForm	
ResOwnerDeleteHTMLForm	
ResOwnerEditHandler	
ResOwnerEditHTMLForm	
ResOwnerEditWebPage	
ResOwnerViewHandler	
ResOwnerViewHTMLForm	
ResOwnerViewWebPage	
RM_PgAuthenticate	Concrete authentication module.
RM_SessionInfo	
RoomEditorWebPage	
RoundTripHandler	Handle requests in a very limited way, echoing back request information.
SimpleWebPage	A simple edu.wisc.cs.rooms.WebPage which displays (1) a title, (2) a list of zero or more "status" messages, usu.
SpecificReserv	Data structure used by SpecificReservWebPage
SpecificReservHandler	Handle requests to view the weekly schedule.
SpecificReservHTMLForm	A SpecificReservHTMLForm is a HTMLForm
SpecificReservWebPage	A SpecificReservWebPage displays the reservation schedule for a given week, and a given format (graphical, which is default, or text-only).
SpecificRoom	Data structure used by SpecificRoomWebPage
SpecificRoomAttrib	
SpecificRoomFacility	

The Rooms and Colloquium System

SpecificRoomHandler	Handle requests to view the weekly schedule.
SpecificRoomHTMLForm	A SpecificRoomHTMLForm is a HTMLForm with two parts: (1) roomNum: an String representing the room number.
SpecificRoomWebPage	A SpecificRoomWebPage displays the specificRooms and their specificRoomAttributes in a given format (graphical, which is default, or text-only).
StartAdmPageHTMLForm	
TestHandler	Handle requests to present the test page
TestHTMLForm	A TestHTMLForm is an HTMLForm with no attributes.
TestWebPage	For testing purposes
UserPrivEditHandler	
UserPrivEditHTMLForm	
ViewColloquiumHandler	Handle requests to view the daily schedule.
ViewColloquiumHTMLForm	An ViewColloquiumHTMLForm is a HTMLForm with following parts: (1) startDate: a java.util.Date which represents the start Date (2) endDate: a java.util.Date which represents the end Date (3) Type: a string which represents the type (4) Speaker: a string (5) Univeristy or company: a string (6) time: a java.util.Time (7) location: a string (8) tabstract: a string
ViewColloquiumWebPage	
ViewRoomHandler	
ViewRoomHTMLForm	
ViewRoomWebPage	
WebPage	A WebPage represents an HTML web page which conforms with the specifications of a Rooms web page (and, by extension, a UW web page).
WeekSchedHandler	Handle requests to view the weekly schedule.
WeekSchedHTMLForm	A WeekSchedHTMLForm is a HTMLForm with two parts: (1) date: the date for which this schedule begins.
WeekSchedRoom	Data structure used by WeekSchedWebPage
WeekSchedTimeBlock	Data structure used by WeekSchedWebPage
WeekSchedWebPage	A WeekSchedWebPage displays the reservation schedule for a given week, and a given format (graphical, which is default, or text-only).

Exception Summary	
HTMLFormParseException	A rooms exception thrown by HTMLForm objects when they are unable to <code>parse()</code> a request.
RM_SessionNotFoundException	
RM_UserExistsException	
RoomsException	An exception specific to the rooms package

[Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV PACKAGE](#) [NEXT PACKAGE](#)

[FRAMES](#) [NO FRAMES](#)

User Manual

Log In	Today's Schedule	This Week's Schedule	Create Reservation	Find A Room	Help	Admin
------------------------	----------------------------------	--------------------------------------	------------------------------------	-----------------------------	----------------------	-----------------------

Rooms Help

Basic Functionality

- [Authenticate](#) (identify yourself to the rooms program)
- [Retrieve a forgotten password](#)
- [View current reservations](#)
- [View the attributes of a room](#)
- [Look up an existing reservation](#)
- [View Colloquia](#)
- [Make Colloquium Information Available](#)
- [Create a room reservation](#)
- [Create a recurring reservation](#)
- [Edit a reservation](#)
- [Cancel a reservation](#) (for recurring reservations, this might be a single instance, or the entire recurrence)

Help With Specific Pages

- [Log in](#)
- [Today's Schedule](#)
- [This Week's Schedule](#)
- [Create Reservation](#)
- [View, Edit, Add or Delete Colloquia](#)
- [Find Colloquia](#)
- [Find a Room](#)
- [Find a Reservation](#)
- [View/Edit Reservation Information](#)
- [Admin](#)

Basic Functionality

Authenticate:

- If you are a new user:
 1. From the main menu at the top and bottom of each page, click "Log in". This will bring up the [log in](#) page.
 2. Enter your desired username and password.

The Rooms and Colloquium System

3. Click the "X" button. This will bring you back to the [log in](#) page which will display your user level. If your username or password is invalid, you will be advised to retype them.
- If you already have a username and password:
 1. From the main menu at the top and bottom of each page, click "Log in". This will bring up the [log in](#) page.
 2. Enter your username and password.
 3. Click the "Y" button. This will bring you back to the [log in](#) page which will display your user level. If your username or password is incorrect, you will be advised to retype them.

Retrieve a lost password:

1. From the main menu at the top and bottom of each page, click "Log in". This will bring up the [log in](#) page.

View current reservations:

- To view reservations for a given day:
 1. From the main menu at the top and bottom of each screen, click "Today's Schedule". This will bring up a table of today's reservations organized by room number and time of day. Below this table is a list of the reservations for today. Below this list are drop-down lists which will allow you to select another day.
 2. Click on any event description to see the [reservation information](#) associated with each reservation.
 3. Set the date and click on the "View Room Reservations" button to view the schedule for another day.
- To view reservations for a given week:
 1. From the main menu at the top and bottom of each screen, click "This Week's Schedule". This will bring up a table of this week's reservations organized by room number and weekday. Below this table is a list of the reservations for this week. Below this list are drop-down lists which will allow you to select another week.
 2. Click on any event description to see the [reservation information](#) associated with each reservation.
 3. Set the date and click on the "View Room Reservations" button to view the schedule for another week.

View Colloquia: There are multiple ways to view colloquia.

- If you know some of the colloquia information:
 1. From the main menu at the top and bottom of each screen, click "Colloquia". This will take you to the [find colloquia](#) page.

The Rooms and Colloquium System

2. Select and enter the colloquia information that you know, and press the "Find Colloquia" button. This will bring up the [colloquia](#) page, which will display all of the colloquium information.
- If you know the room reservation with which the colloquia is associated:
 1. Go to the [reservation information](#) page for the reservation you are interested in. (See the section on [finding a reservation](#), and the section on [viewing, editing, and cancelling a reservation](#)).
 2. Click the colloquia link which is in the "Colloquia Information" section of this page. (Toward the bottom of the page, there is a colloquia section which lists the colloquia titles associated with this reservation. Each title is a hyperlink to the [colloquium page](#) associated with each colloquium.)

Add Colloquia: There are multiple ways to add colloquia.

- If you wish to make available colloquium information that is associated with a room reservation:
 1. Go to the [reservation information](#) page for the reservation you are interested in. (See the section on [finding a reservation](#), and the section on [viewing, editing, and cancelling a reservation](#)). NOTE: If you need to make a room reservation for this, simply create a new reservation and click on its link in the [daily schedule](#) page to take you here. (See the section on [creating room reservations](#).)
 2. Click on the "add colloquium" link in the "Colloquium Information" section of this page. This will take you to the [view, add, edit or delete colloquia](#) page.
- If you wish to make available colloquium information that is not associated with any room reservation:
 1. From the main menu at the top and bottom of each screen, click "Colloquia". This will take you to the [view colloquia](#) page.
 2. Near the top of the page, there is a link to add new colloquium information.

View the attributes of a room:

1. From the main menu at the top and bottom of each page, click "Find A Room". This will bring you to the [find a room](#) page, which allows you to select the room you are interested in.
2. Click the submit button to bring up the attributes for the given room.

Look up an existing reservation:

1. From the main menu at the top and bottom of each page, click "Find A Reservation". This will bring you to the [find a reservation](#) page, which allows you to specify the reservation you are interested in.
2. With the month/date/year drop-down lists, Specify the earliest and latest reservation date of the reservation(s) you are interested in.

The Rooms and Colloquium System

3. **Optional:** enter the username associated with the reservation(s) you are interested in.
4. **Optional:** enter the description of the reservation(s) you are interested in.
5. Click the "View Specific Reservations" button. This will bring up a new page which contains the reservations which match your specification.

Create a reservation:

1. From the main menu at the top and bottom of each page, click "Create Reservation". This will bring you to the [create a new reservation](#) page.
2. Enter the necessary information, including your user name, a description of the event, the room you mean to reserve, and the time your reservation begins and ends.
3. Click the "Make Reservation" button to make the reservation.
4. If you wish to make a recurring reservation, enter in the appropriate recurrence information, and click "Make Recurring Reservation"

View, edit, or cancel a reservation:

1. Choose one of the following methods to bring you to the [reservation information](#) page:
 - o From the daily schedule:
 1. From the main menu at the top and bottom of each page, click "Today's Schedule". This will bring up the [daily schedule](#) page.
 2. At the bottom of this page, set the date on which the reservation you're interested in occurs. Click the "View Room Reservations" button to bring up the reservations for that day.
 3. Click the description of the reservation you're interested in. (This appears in a list of reservations which is located under the table of reservations for the date you selected.)
 - o From the weekly schedule:
 1. From the main menu at the top and bottom of each page, click "This Week's Schedule". This will bring up the [weekly schedule](#) page.
 2. At the bottom of this page, set the date on which the reservation you're interested in occurs. Click the "View Room Reservations" button to bring up the reservations for that week.
 3. Click the description of the reservation you're interested in. (This appears in a list of reservations which is located under the table of reservations for the date you selected.)
 - o By entering the reservation information:
 1. From the main menu which appears on the top and bottom of each page, click "Find a reservation". This will bring up the [find a reservation](#) page.
 2. Enter the reservation information

3. Click the submit button to take you to the reservation information page.
 2. If you wish to edit the reservation information, enter it in to the "Reservation Information" section and click the "Update Reservation" button.
 3. If you wish to cancel the reservation, click the "Delete Reservation" button.
 4. If this is a recurring reservation, and you wish to cancel *just this instance*, click the "Delete Reservation" button (the same as above).
 5. If this is a recurring reservation, and you wish to cancel *the entire recurrence*, click the "Delete Entire Recurrence" button.
 6. If this is a recurring reservation, and you wish to cancel *a different instance* of the recurrence, take the following steps:
 1. Select the date on which this reservation occurs.
 2. Click the "Go To Selected Occurrence" button. This will bring up the [reservation information](#) page for the selected occurrence.
 3. Click the "Delete Reservation" button.
-

Help With Specific Pages

Log in Page:

1. The page will display your user level.
2. Enter your username.
3. Enter your password.
4. Click the "Submit" button.
5. This will return you to the log in page, which will display your new user level. If your username or password is invalid, the log in page will advise you of the problem.

Daily Schedule Page:

1. You will see a table of reservations for a given day organized by room number and time-of-day. The reservations appear as colored bars within the table cells indicating which room they are in, and what time they begin and end.
 - o Click the colored bar to bring up the [reservation information](#) page.
2. You will see a list of the reservations which appear in the table. The reservations in this list are associated with a color, indicating which reservations are shown in the table.
 - o Click on the reservation event description to bring up the [reservation information](#) page.
3. A form which allows you to view the schedule for another day.
 1. Select the month, date, and year for the daily schedule you wish to view next.
 2. Click the "View Room Reservations" button to view the schedule for that day.

Weekly Schedule Page:

1. You will see a table of reservations for a given week organized by room number and date. The reservations appear as colored bars within the table cells indicating which room they are in, and what time they begin and end.
 - Click the colored bar to bring up the [reservation information](#) page.
2. You will see a list of the reservations which appear in the table. The reservations in this list are associated with a color, indicating which reservations are shown in the table.
 - Click on the reservation event description to bring up the [reservation information](#) page.
3. A form which allows you to view the schedule for another week.
 1. Select the month, date, and year for the weekly schedule you wish to view next.
 2. Click the "View Room Reservations" button to view the schedule for that week.

View, Add, Edit or Delete Colloquia Page:

1. User information
 - Lists the user information
2. Colloquium information
 1. Enter the speaker
 2. Enter the institution
 3. Enter the type
 4. Enter the title
 5. Enter the abstract
3. Reservation information
 - If this colloquium is associated with a room reservation, the room reservation information will be displayed here.
 - If this colloquium is not associated with any room reservation, you will be able to update the following information:
 1. Enter the date (via drop-down lists for the month, date, and year)
 2. Enter the time (via drop-down lists for the hour, minute, and AM/PM)
 3. Enter the room number (via a drop-down list of all rooms)
4. Add colloquium button
 - If this is new colloquium information, press this button to make the colloquium information available. This button will not appear if your user authentication level does not allow it.
5. Update colloquium button
 - If this is existing colloquium information, press this button to update the colloquium information as specified on the page. This button will not appear if your user authentication level does not allow it.
6. Delete colloquium button

The Rooms and Colloquium System

- If this is existing colloquium information, press this button to remove the colloquium information. This button will not appear if your user authentication level does not allow it.

Find Colloquia Page:

1. Link to add colloquia
 - If you wish to add colloquia that is not associated with a room reservation, click this link. It will take you to the [view, add, edit or delete colloquia](#) page, which will allow you to make colloquia available.
2. Find Colloquia
 1. Specify the information you know about the colloquium information you are interested in.
 2. Click the submit button. You will be returned to this page, with your results displayed at the top of the screen.

Create Reservation Page:

1. Reservation information
 1. Enter the username for which this reservation is scheduled.
 2. Enter the event description
 3. Enter the room number (via a drop-down list of all room numbers)
2. Date/time information
 1. Enter the date on which this reservation occurs (via drop-down lists for the month, date, and year)
 2. Enter the time which this reservation begins (via drop-down lists for the hour, minute, and AM/PM)
 3. Enter time which this reservation ends (via drop-down lists for the hour, minute, and AM/PM)
3. Click the "Make Reservation" button to create a reservation which occurs once. If the reservation cannot be created, you will be returned to this page, and a list of reasons will be displayed at the top.
4. Recurrence information (Note: For recurring reservations, the begin date/time of this recurrence is the date/time listed above)
 1. Choose the method for defining the end date for this recurrence with one of the following:
 - If you wish to choose the recurrence end date yourself:
 1. Click the option button by "User defined"
 2. Enter the recurrence end date (via drop-down lists for the month, date, and year)
 - If you want the recurrence to end at the end of the current session:
 1. Click the option button by "End of current session"
 2. The recurrence will end at the date indicated
 2. Choose the recurrence frequency. This represents the number of days between each instance. If you wish to schedule a reservation which occurs only on certain days of the week, skip this step and go to the next one.

The Rooms and Colloquium System

You may change this value, for instance, if you need the reservation to take place once every other week--you would enter 14. The default value is 1, which should take care of most recurring reservations.

3. Choose the weekdays on which this recurrence applies. For instance, if you wish to schedule a reservation which occurs only on Mondays, Wednesdays, and Fridays, select these, and only these, check-boxes. The recurring reservation will not be scheduled on the weekdays which are not selected.
5. Click the "Make Recurring Reservation" button to create a reservation which occurs on the days specified. If the reservation cannot be created, you will be returned to this page, and a list of reasons will be displayed at the top.

Find a Room Page: No information is available about this page.

View/Edit Reservation Information Page:

1. Edit the reservation information. Note: If this is a recurring reservation, this information applies to *all* instances, not just this one.
 1. Edit the username
 2. Edit the event description
 3. Change the selected room (via a drop-down list of all rooms)
2. View the date/time information.
 1. The date on which this reservation occurs
 2. The time this reservation begins
 3. The time this reservation ends
3. Click the "Delete Reservation" button to delete this reservation. If this is a recurring reservation, clicking this button will delete only the instance indicated by the date/time information above.
4. For recurring reservations only, view the recurrence information.
 1. Select the instance of this recurrence that you are interested in (via a drop-down list of date/time information)
5. For recurring reservations only, click the "Go To Selected Instance" button to go to the reservation information page for the instance selected in the previous step.
6. For recurring reservations only, click the "Delete Entire Recurrence" button to cancel all instances of this recurrence.

Admin Page: (Only administrator users will be allowed to access this page. Other users will be denied.)

1. Make a room available
2. Make a room unavailable
3. Create A Room
4. Edit Room Attribute
5. Edit the user's privilege
6. Remove Old Reservations

The Rooms and Colloquium System

1. Choose minimum date (via a drop-down list of month, date, year) **Be careful!** The default date is the current date, so you must choose the minimum date carefully.
2. Click the "Remove" button. This will delete all reservations from the database that occur before the given date. You will be directed to a page indicating "SUCCESS" (even if no reservations were deleted because none qualify).

Log In	Today's Schedule	This Week's Schedule	Create Reservation	Find A Room	Help	Admin
------------------------	----------------------------------	--------------------------------------	------------------------------------	-----------------------------	----------------------	-----------------------

[Computer Sciences Department](#)
[University of Wisconsin - Madison](#)

5355a Computer Sciences and Statistics ~ 1210 West Dayton Street, Madison, WI 53706
cs@cs.wisc.edu ~ voice: 608-262-1204 ~ fax: 608-262-9777

Maintenance Plan

1. Once the project is finished, burn the lot onto CD-ROM, and deliver it to the lab (this includes all documentation as well as source code).
2. Work with the lab setting up the permanent location for that rooms program. This may require various changes to the source code and/or database.
3. Continuing students will be available for contact from the lab and/or users regarding questions, minor enhancements and bug fixes.

Index

Acctuser, 7

Administration, 8, 32

Administrator, 7, 11

Architecture, 13, 14, 15

Class Diagram, 16

Class Professor, 6

Code Review, 32

Database, 14, 30

Glossary, 7

Group Mailing List, 5

Group Members, 6

Group Mentors, 6

Maintenance Plan, 71

Reservation, 7, 8, 9, 21, 32

Room Attributes, 7

Schedule, 17

Schema, 30

Sequence Diagrams, 17

System Testing, 32, 33

Targets, 14

Team, 6

Test Plan, 32

User, 7, 8, 9, 11, 12, 31, 32

Users, 5, 8, 11

Visitor, 7, 11

Web Page, 5