Room Reservation System Documentation

Release 1.0.0

R2E2 Team of Developers

June 04, 2015

Contents

CHAPTER 1

Introduction

The DCS Room Reservation Service (RRS) is an online platform for reserving rooms in the Department of Computer Science (DCS). It aims to provide an efficient and convenient way of reserving rooms for all the individuals involved. By hosting the service online, the entire flow of the reservation system will not be greatly affected by the absence of important individuals. (Example: Suppose that the building administrator or the Department Chair of the building, whose approval is needed for the room reservation, is out of the country then they can still fulfil their duties online.)

It also aims to to prevent issues caused by human error can easily be avoided by the RRS, like preventing conflicting reservations.

This project also aims to be extended to a bigger scope being deployed throughout the College of Engineering and providing a uniform reservation system for the entire college.

Contents:

1.1 System Features

1.1.1 Managing Users

Add User

- Description and Priority

  This is a high priority feature which allows the system administrators to create new user of a particular type.

- **Stimulus/Response Sequences**

Step	User Input	Response
1	System Admin opens **Add User Form**	RRS shows Add User Form for A
2	System Admin enters necessary fields including name, email address, and roles (Building Admin, Faculty, Orgs, etc) and submits them.	RRS checks if email is unused an valid up mail account and sends a confirmation email to the account
3	User opens email and activates account using the link provided in the email. The user also provides a password for his account.	RRS creates new account.

Edit User

- Description and Priority

This is a medium priority feature which allows the system administrators or users to edit user information.

- **Stimulus/Response Sequences**

Step	User Input	Response
1	System Admin opens **Edit User Form**	RRS shows Edit User Form for Admins
2	System Admin presses **Delete User**	RRS asks for confirmation and informs that the System Admin that t cannot be undone.
3	System Admin confirms	RRS marks that user deleted and can no longer be used. Also deletes future room reservations that the user has made. Note: User account since past reservations are associated with the user.

Search Users

- Description and Priority

  This is a medium priority feature which allows the system administrators to search users based on some chosen filters.

- **Stimulus/Response Sequences**

Step	User Input	Response
1	System Admin/User opens **List of Reservations Form**	RRS shows all users in a table forma
2	System Admin/User edits search fields and submits	RRS filters users and present users in format.

1.1.2 Making Reservation Requests

Make Request for Room Reservation

- Description and Priority

  This is a high priority feature which allows the users to create a request for a single room reservation.

- **Stimulus/Response Sequences**

Step	User Input	Response
1	User opens **Reserve Rooms Form**	RRS shows *Add Reservation Form*
2	User fills up the necessary fields and submits	RRS checks for any conflicts with any approved roo reservation and adds new request to the database, no the faculty adviser, building admin and the departme
3	(Needed only if the one making the request is an org) Advisor approves/disapproves Requests	RRS marks the new request as “approved by adviser RRS marks the request as disapproved and notifies t student.
4	Building Admin approves/disapproves new request	RRS marks the new request as “ approved by buildin admin” Or RRS marks the request as disapproved an notifies the student.
5	Department Chair approves/disapproves new Request If all needed approval are received	RRS marks the new request as “approved by departm chair” Or RRS marks the request as disapproved and notifie student. RRS approves request and includes it in the reservations.

1.1.3 Viewing Room Reservation Requests

Tracking Requests Made

- Description and Priority

  This is a medium priority feature which allows the users to find their requests and to see the program of its approval.

- **Stimulus/Response Sequences**

Step	User Input	Response
1	User opens **List of Reservations** page	RRS gets the requests made by that user and displaying it on a table. requests have tags saying whether or not they have been approved by advisor, building admin and/or dept. chair.
2	User fills up search fields and submits	RRS filters and displays only the requests that match the given search

1.1.4 Managing Rooms

Add Room

- Description and Priority

  This is a high priority feature which allows the building administrators to create new room of a particular type.

- **Stimulus/Response Sequences**

Step	User Input	Response
1	System Admin opens **Add Room** form	RRS shows Add Room form
2	System Admin enters necessary fields and submits	RRS adds room to the database.

Edit Room

- Description and Priority

  This is a low priority feature which allows the building administrators to edit room information.

- **Stimulus/Response Sequences**

Step	User Input	Response
1	System Admin opens **Edit Room** form	RRS shows Edit Room form along with the informatio associated with that room.
2	System Admin edits fields and submits	RRS updates database.

Remove Room

- Description and Priority

  This is a low priority feature which allows the building administrators to remove the room from the database.

- **Stimulus/Response Sequences**

Step	User Input	Response
1	System Admin opens **Edit Room** form	RRS shows Edit Room form along with the information associated w room.
2	System Admin presses "Delete Room"	RRS asks for confirmation and informs that this cannot be undone.
3	System Admin confirms	RRS marks that room deleted and can no longer be used. Also delete future room reservations to that room. Note: Room information is ke since past reservations are associated with the room.

Search Room

- Description and Priority

  This is a low priority feature which allows the building administrators to search for a based on some chosen filters.

- **Stimulus/Response Sequences**

Step	User Input	Response
1	User opens **View Rooms** page	RRS shows all rooms in a calendar format
2	User edits search fields and submits	RRS filters rooms and present rooms in a table

View Room Availability

- Description and Priority

  This is a high priority feature which allows the users to look at the availability of rooms for a particular week.

- **Stimulus/Response Sequences**

Step	User Input	Response
1	User opens **View Rooms** page	RRS shows all rooms in a calendar format
2	User edits search fields and submits	RRS filters reservations and displays all available time slots for particular room in the week of the given date.

1.2 User Manual

1.2.1 Introduction

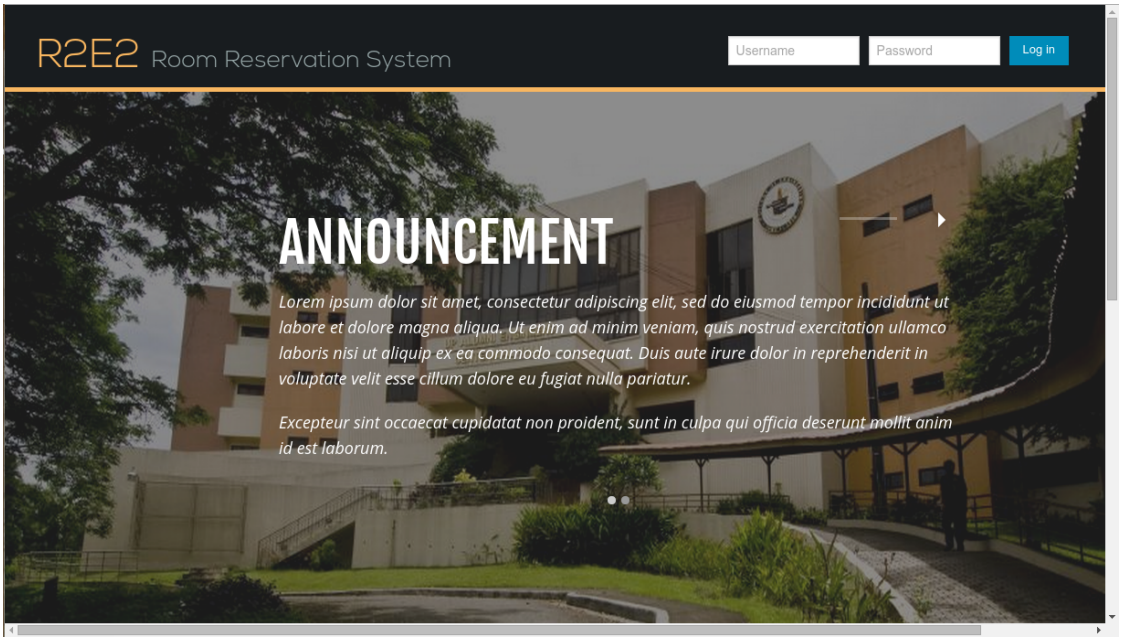
This user manual is the how-to guide for guests, student organizations, advisers, faculty, and department chairpersons.

1.2.2 How to Make an Account

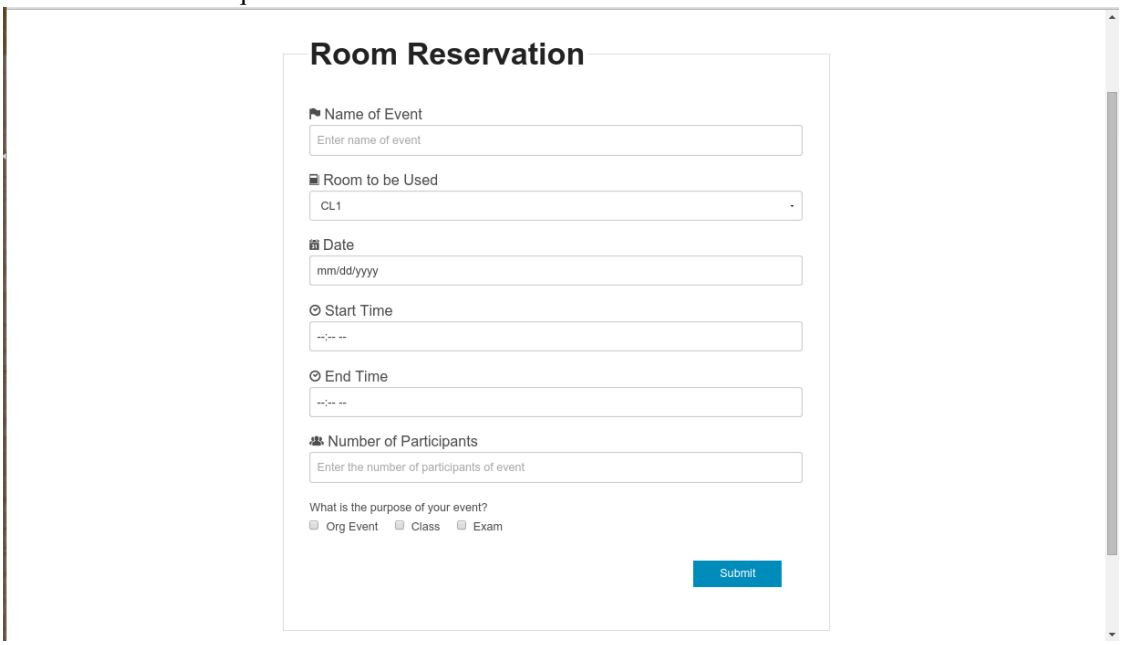
1. Contact the System Administrator and request for an account.
2. When approved, wait for email notification. (UP Webmail)

1.2.3 How to Make an Application Requests

1. Log-in using your account.



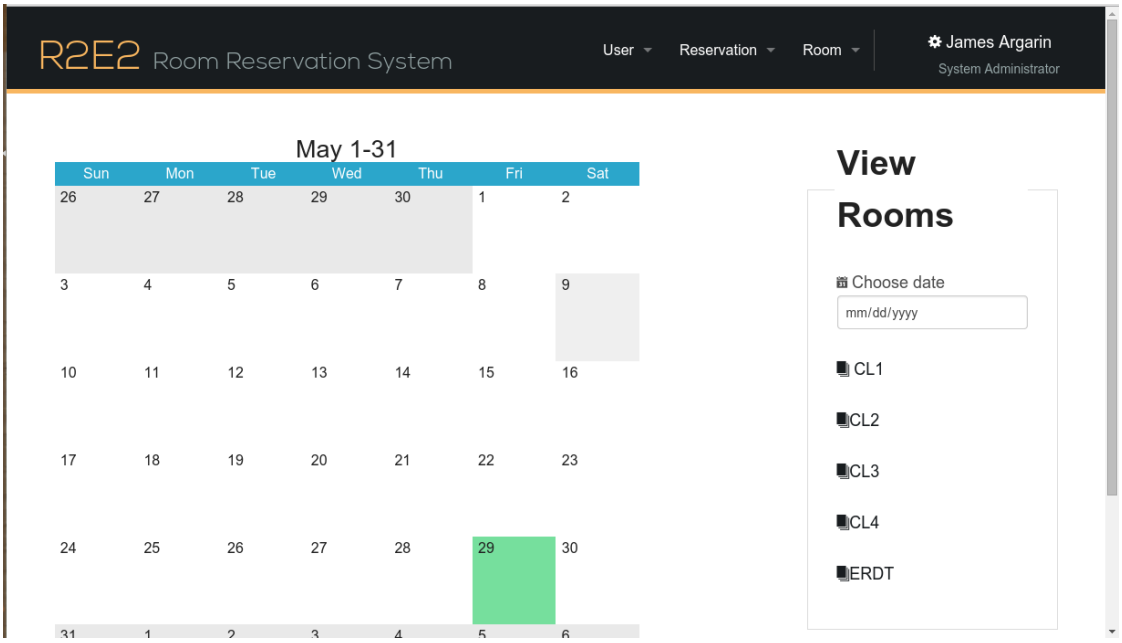
2. Click “Request Room” in the left sidebar.



3. Fill-up necessary information.
4. Click Submit.
5. Wait for your application to be approved.

1.2.4 How to View Room Availability

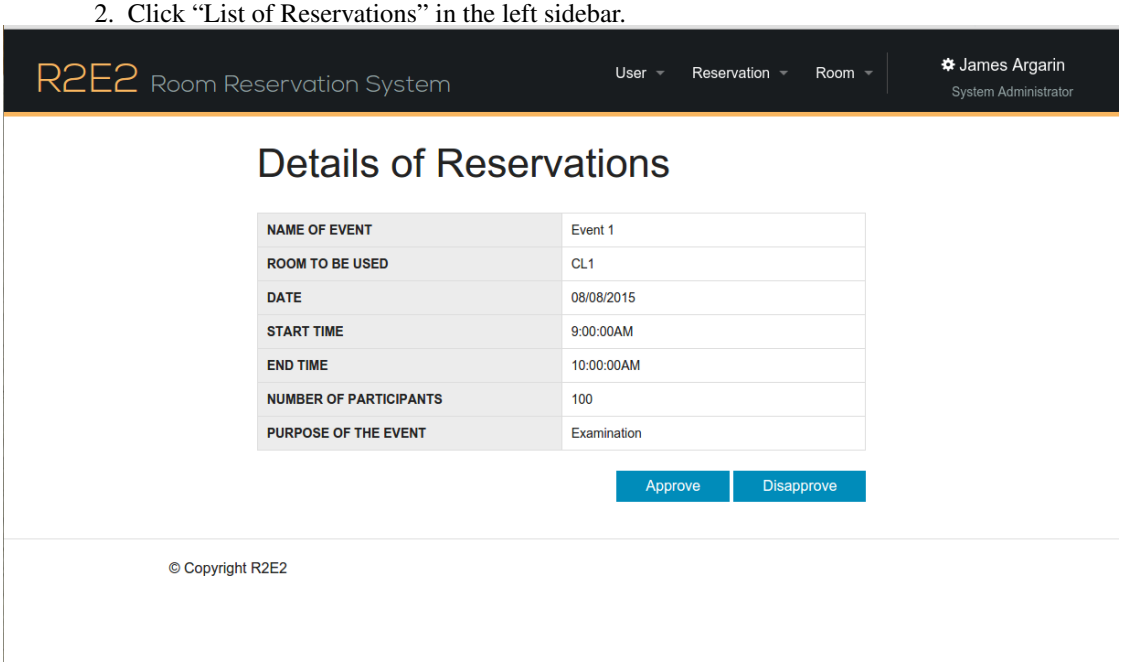
1. Log-in using your account.
2. Click “View Rooms” in the left sidebar.



3. Search the name of the room you want to view.

1.2.5 (Adviser, Building Administartor, Department Chairperson) How to Approve Requests

1. Log-in using your account.
2. Click “List of Reservations” in the left sidebar.



3. Click the reservation you want to view.
4. Click Approve button if desired.

1.3 System Administrator's Guide

1.3.1 Introduction

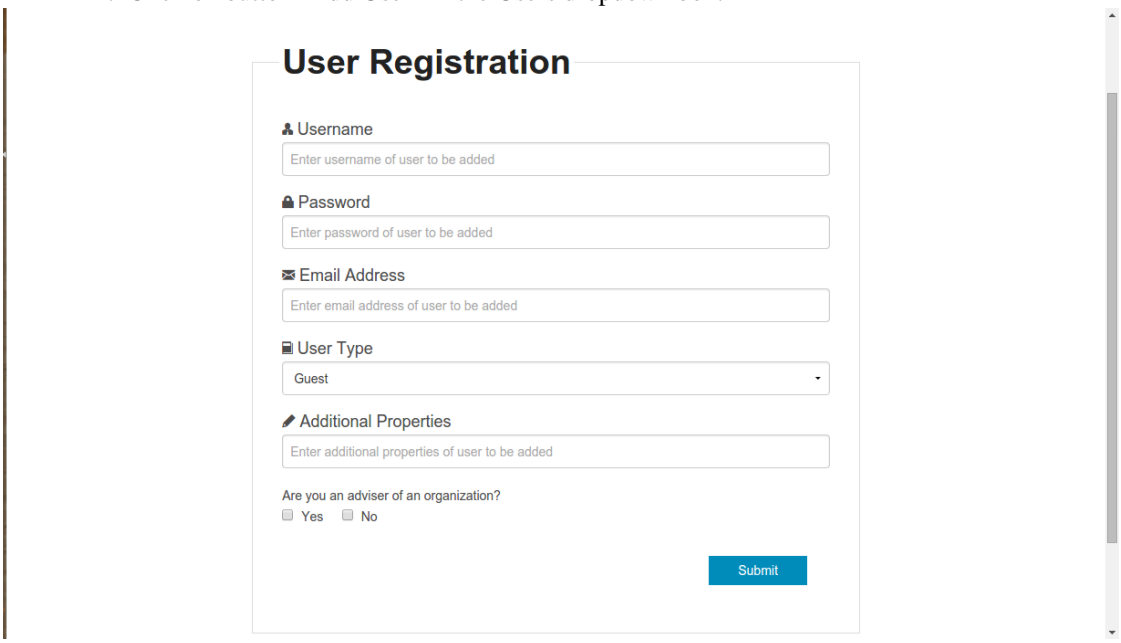
The system admin has the ability to make new users, edit their information and to delete user accounts.

1.3.2 How to make a new System Admin

1. Log-in using the Django Admin. "link/admin"
2. Click "+Add" button in the Users table.
3. Fill up necessary information.
4. Go back to the Home page and click "+Add" button in the User Profiles table.
5. Choose the newly created user and choose "System Admin" in the user type.
6. Click Save.

1.3.3 How to Make a User

1. Click on button "Add User" in the Users dropdown box.



2. Provide the necessary information along with choosing the type of user.
3. Submit information to create new user.

1.3.4 How to Edit User

1. Log-in using the Django Admin. "link/admin"
2. Click "Change" button in the Users table.
3. Select the user you want to edit.

4. Edit fields with the new information.
5. Click Save.

1.3.5 How to Filter User

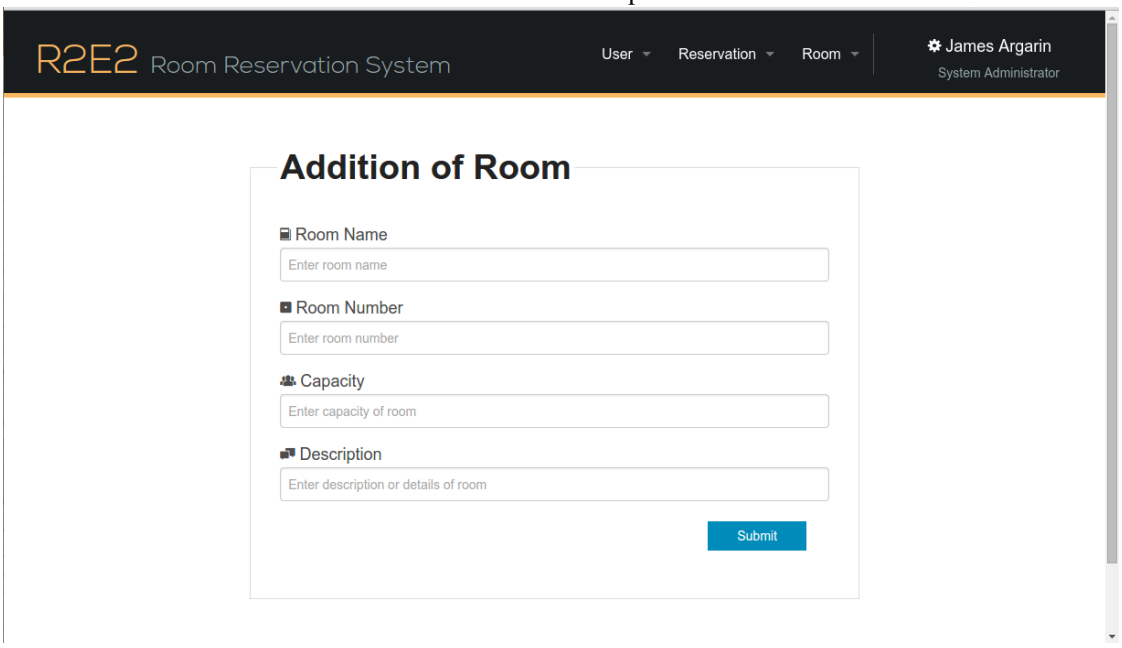
1. Click on the button “List of Users” in the Users dorpdown box.
2. Edit the search bar on the top of the screen.
3. Press Filter.

1.3.6 How to Remove User

1. Log-in using the Django Admin. “link/admin”
2. Open the Users table.
3. Click the name of the user you want to remove.
4. Click the Delete button.
5. Confirm deletion.

1.3.7 How to Add a Room

1. Click on button “Add Room” in the Rooms dropdown box.



2. Provide the necessary information along with choosing the type of user.
3. Submit information to create new room.

1.4 Technical Document

1.4.1 Introduction

The room reservation system is composed of different modules, classes, and tools for the software itself to become fully functional. Its components, their relationships, and their features are described in this portion as well as the dependencies used for optimal end-product utilization.

1.4.2 Database Schema

The database schema of the software presents the relationships between the classes (relations) involved in the system's backend part. It also includes the fields of each classes and how they are used to interact with other classes.

The following figures show the initial and future database schemas of the room reservation system.

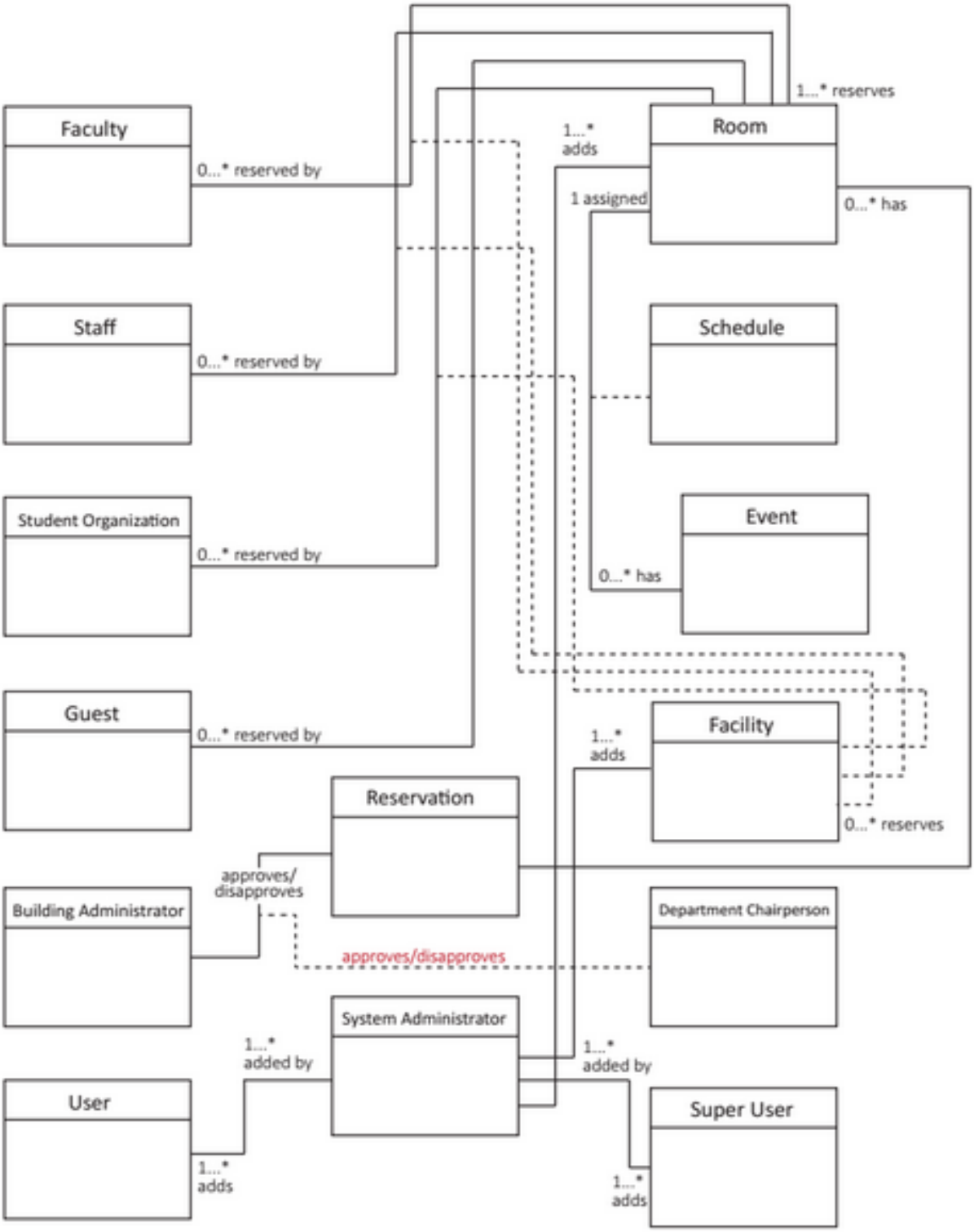
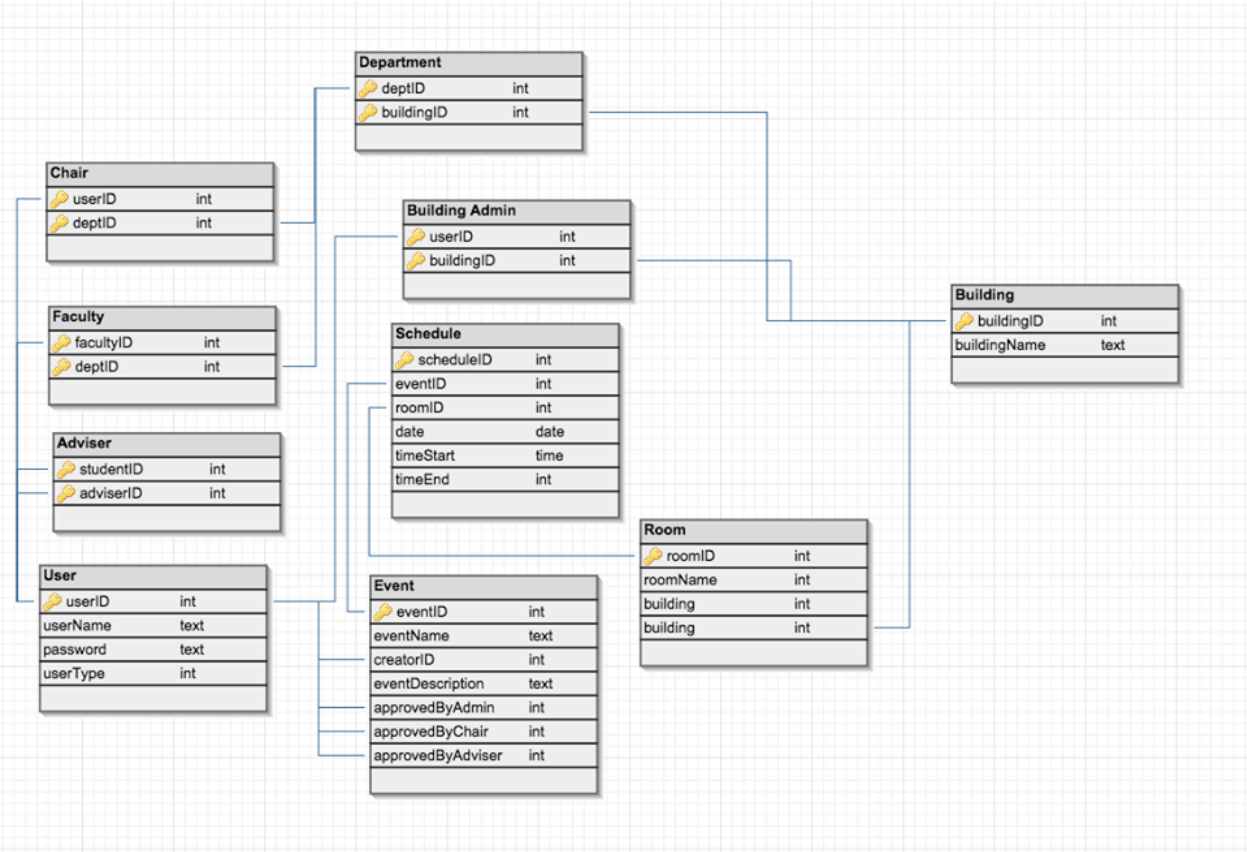
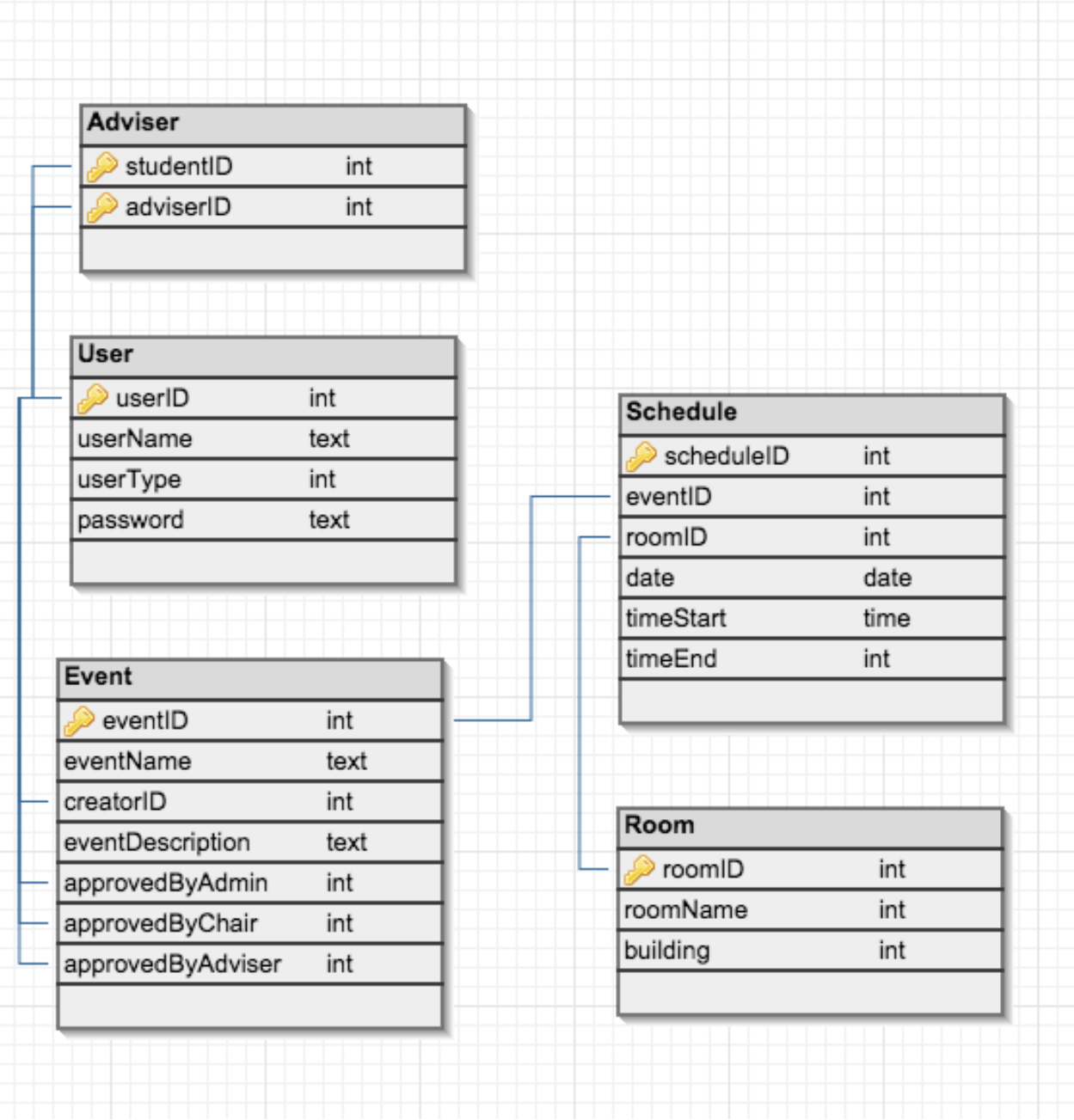


Figure 1 basically shows the allowable instances of users to reserve a room. It also shows how a reservation is created and how it is approved by the Building Administrator and later by the Department Chair. For a reservation to be instantiated, the reservation class needs the list of pre-added rooms, events, and schedules by the users. System Administrator creates the user profiles according to their types and hierarchy. This is the current database schema of

the room reservation system.





1.4.3 Modules

Current:

users

forms.py

- user registration form

views.py

- add user(s)

- view user(s)

models.py

- user attributes reflected in database

urls.py

- reference for user navigation on the website

reservations

forms.py

- reservation form

views.py

- add reservation(s)
- view reservation(s)
- show reservation(s) status
- pending
- approved
- disapproved
- show reservation(s) timestamp
- change reservation(s) status
- approved
- disapproved

models.py

- reservation attributes reflected in database

urls.py

- reference for user navigation on the website

rooms

forms.py

- room (registration) form

views.py

- add room(s)

models.py

- room attribute reflected in database

urls.py

- reference for user navigation on the website

To follow:

users

- edit user settings
- delete user

reservations

- Adviser for "organization(s)"
- filter reservations
- reservation history

rooms

- delete room(s)

1.4.4 Dependencies

Database

PostgreSQL (Postgres) - An object-relational database management system (ORDBMS) used to store and retrieve data securely. - The extensibility, reliability, stability, and compatibility of PostgreSQL are the key reasons why we choose it as our project database server.

Django-crispy-forms

- A Django application that let developers easily customize forms using a CSS framework without writing custom form templates
- It provides a (|crispy) filter and ({%crispy%}) tag that controls the rendering behavior of Django forms.

Django-braces

- A Django application similar to Django decorators (e.g., login_required, user_passes_test)
- It offers a lot of view restrictions that can be applied in different user access

1.5 Testing

1.5.1 Use Case Testing

Use Case Testing is a functional black box testing technique that helps testers to identify test scenarios that exercise the whole system on each transaction basis from start to finish.

The RRS has the following use cases:

- login
- home page/dashboard
- logout
- addition of users
- addition of rooms
- viewing of rooms
- addition of reservations
- viewing of reservations

- approval of reservations
- status of reservation

Test Tools

Selenium WebDriver

The primary new feature in Selenium 2.0 is the integration of the WebDriver API. WebDriver is designed to provide a simpler, more concise programming interface in addition to addressing some limitations in the Selenium-RC API. Selenium-WebDriver was developed to better support dynamic web pages where elements of a page may change without the page itself being reloaded. WebDriver's goal is to supply a well-designed object-oriented API that provides improved support for modern advanced web-app testing problems.

Test Method

In the main folder of the project, run the following code:

```
python manage.py test
```

1.5.2 Usability Testing

The alpha version of the RRS was deployed in Heroku and can be seen here: http://r2e2.herokuapp.com/users/login

Test Method

RRS was officially introduced to potential users using Social Media accounts (i.e Facebook). They were asked to try the system for the first time using accounts the System Admin had initially created. The following instructions were given to the users:

1. Visit this page: http://r2e2.herokuapp.com/users/login
2. Login using: Username: <*username*>, PW: <*password*>
3. Click reserve rooms.
4. Fill up the form.
5. Wait for your request to be approved.

After trying the system, the users were asked to answer a survey powered by GoogleDocs, through this link: http://bit.ly/r2e2_survey

Test and Survey Results

A total of 15 users were able to use the system properly and were able to reserve rooms. The following table displays the questions asked in the survey and the average score of the participants. A score of 1 means the user Agrees with the statement, and a score of 5 means the user Disagrees with the statement.

Statements/Questions	Average Score
I found the design of the website appealing	2.3
I think I would like to use this system	1.7
I found this system unnecessarily complex	3.8
I thought the system was easy to use	1.6
I think that I would need the support of a technical person to be able to	4.1
use this system.	
I found the various functions in the systems were well integrated	2.1
I thought there was too much inconsistency in this system	3.7
I would imagine that most people would learn to use this system very quickly	1.6
I found the system very cumbersome to use	3.7
I felt confident using the system	1.7
I needed to learn a lot of things before I could get going with this system	4.5

Users were also asked for their comments and suggestions about the system. They include:

- more rooms for reservations
- a calendar view for the rooms
- improve User Interface

The RRS deployment is a success based on the results of the survey.

CHAPTER 2

Overall Description

2.1 Project Scope

There are no information systems that are currently being implemented and the RRS is a project will be replacing the reservation system of the DCS, which is currently being implemented manually by people, building admins.

The RRS will aim to be self-contained, meaning that it will have no dependencies to other existing information system.

It is expected that more functionalities will be included, or that this system might be extended. The RRS will be implemented and tested in DCS however, if successful, it will be done be deployed in the entire college.

2.2 Project Features

The RRS will have two general roles, the users and the administrators, and the basic functionalities that each of these roles have are shown below.

Feature	Description
View Room Availability	Users can check the availability of a particular room for some chosen date. Admins can check the availability of a particular room for some chosen date along with all the names of the events that will occur.
Request for Room Reservation	Users can make requests for reservations where they provide necessary information including the date and time that they want to use the room.
Track Room Reservation Requests	Users can check the status of their requests to know whether or not their requests have been approved.
Manage Users	Admins can create and delete the users, and also edit their privileges.
Manage Rooms	Admins can create, edit, and delete the rooms that are available for reservation.
Approve / Disapprove Room Requests	Admins can approve or disapprove room requests and also keep track of all approved room reservations.
Make Room Reservations	Admins can reserve rooms without the need of making a request.

These are the main functionalities that must be implemented. However, there are some functionalities that may be implemented but are not as important as the ones listed above: a billing system for the fees and an activity approval system.

2.3 User Classes and Characteristics

As mentioned in the Product Features, there are two main roles in the RRS, the users and the admins. The table below described the specific roles.

Role	Description
Student Organizations	An org is a type of user. For a request made by an org to be approved, it must be approved by the org's adviser, the building admin and also the Department Chair of the building.
Faculty	A faculty is type of user. For a request made by a faculty, it must be approved by the building admin and also the Department Chair. Faculties generally have a higher priority over orgs and they can also have roles as an advisor for multiple orgs.
Guests	A guest is a temporary user that is given to individuals or organisations that are from outside UP. For a request made by a guest, it must be approved by the building admin and the Department Chair of the building. Guests will have a special tag on their requests to indicate and remind the building admin and the Department Chair that these are special users.
Advisor	An advisor approves or disapproves requests made by the orgs they handle.
Building Administrators	Building administrators have the ability manage the rooms of their respective buildings and all requests for reservations for rooms in their building has to be approved by them.
Department Chair	The Department Chair or the dean of a building has the final say on whether or not an event will be approved. The dean is also considered to be a special type of faculty.
System Administrators	System administrators has the ability to manage the users. Building administrators will usually be also system administrators.

2.4 Operating Environment

The current operating environment is Heroku.

CHAPTER 3

Search

- search