

# RoSys Program Manual

## Introduction

This manual is a guide that should help clarify some of the ideas we used throughout the program. It will detail how some parts of the program works.

An important thing to keep in mind about our program is that some parts, such as the user interface, were made purely to demonstrate the programs functionality. The main part of the program was the “Core” project, which is meant to serve as a backend server to any/multiple user exposed interfaces.

Our focus with the Core is the ability to integrate into other systems, so we made sure to use interfaces for some of the main resources with the program.

This focus on being able to integrate the program, as well as wishing to demonstrate the programs functions on its own, was a greater challenge than expected.

## The User Interface

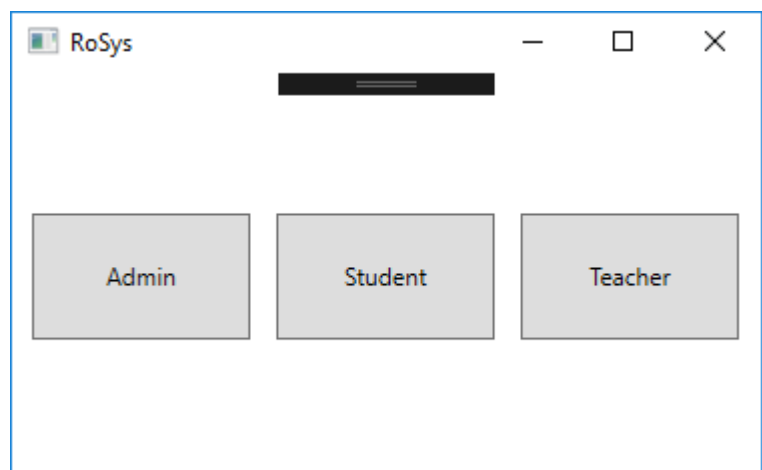
**Remember: The User interface was made purely to demonstrate features of our program**

### Initial Run

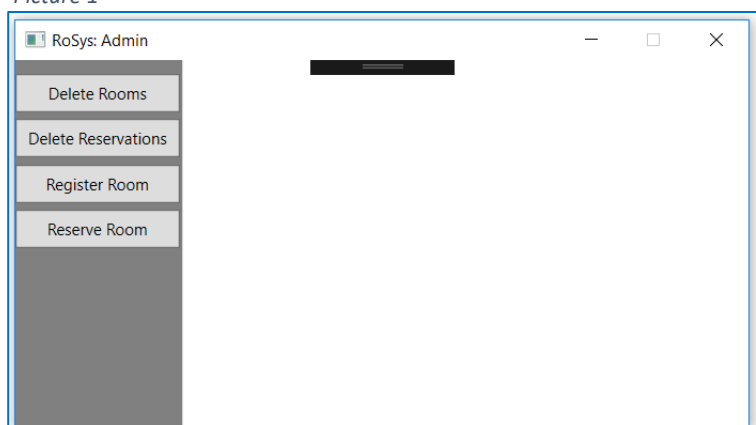
When running the program, there are, initially, 3 options, as seen in Picture 1. This is essentially our “login”, where selection is made of which type of user will interact with the program.

The different types of users have different functionality. Student and Teacher generally have the same features, but Teacher has access to rooms that Student does not.

Admin has some management features, like deleting rooms and reservations, registering new rooms, selecting specific rooms for a reservation and access to a wider variety of rooms to reserve.



Picture 1



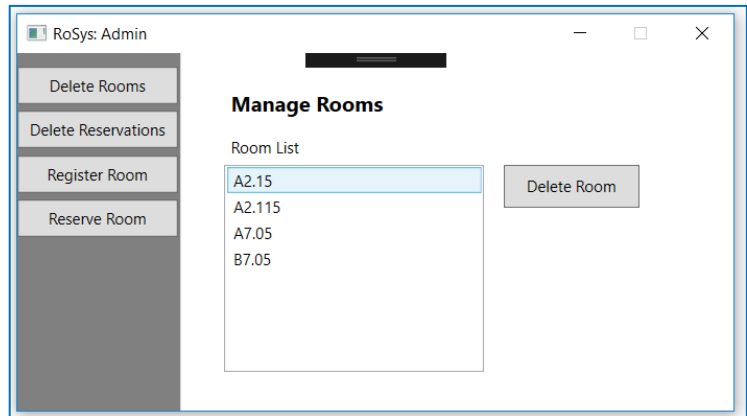
Picture 2

By pressing the “Delete Rooms” button, in Picture 2, a list of rooms appears, from which a selection can be made, and then can be deleted, Picture 3. This will delete Rooms from the repository and database. Also, it will delete all reservations for that room from the program and database.

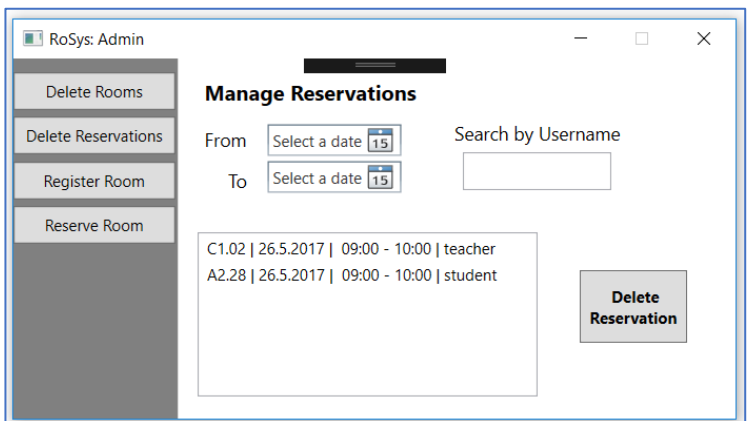
When the “Delete Reservations” button is pressed, a page, “Manage Reservations”, appears, Picture 4. If both select dates are empty as well as “Search by Username” the list shows all reservations. If only “Search by Username” is filled, then the list shows all reservations for that specific user. If only one “Select a date” field is filled with a specific date, then the list shows either all reservations from that date or until that date, depending on which date was filled in – from or to. If both “Select a date” fields are filled, then the list shows all reservations for that period. If only one “Select a date” field is filled with a specific date and “Search by Username” is filled, then the list shows either that users reservations from that date or until that date, depending on which date was filled in, the same as previously. And if all fields are filled in, then the list displays all reservations for that specific user in that specific period. If an admin selects a reservation and presses the “Delete Reservation” button that reservation will be removed from the repository and database and a message will appear, Picture 5. To close the message box, the admin needs to press the “OK” button.

To register a new room in RoSys, an admin needs to press the “Register Room” button, Picture 6. Before pressing the “Register Room” button in the “Register a new room” page, all fields must be filled, otherwise the program will crash. In the “Building” field, only one letter can be inputted. If all fields are filled and the “Register Room” button is pressed, a message will popup, Picture 7. To close the message box, the admin needs to press the “OK” button.

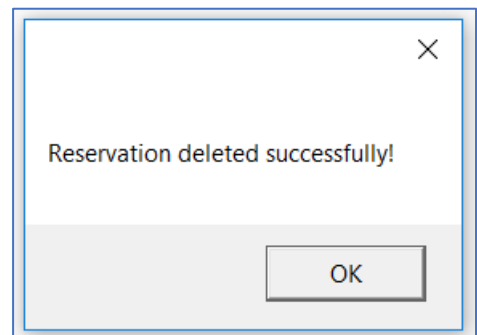
Admins can also reserve rooms. All fields must be filled. Date and times are automatically set to today’s date and 9:00-10:00, respectively. The user needs to exist in the database and repository and the text, “Enter a username”, must be cleared before the admin can input the username. The admin also needs to select which room he wants the



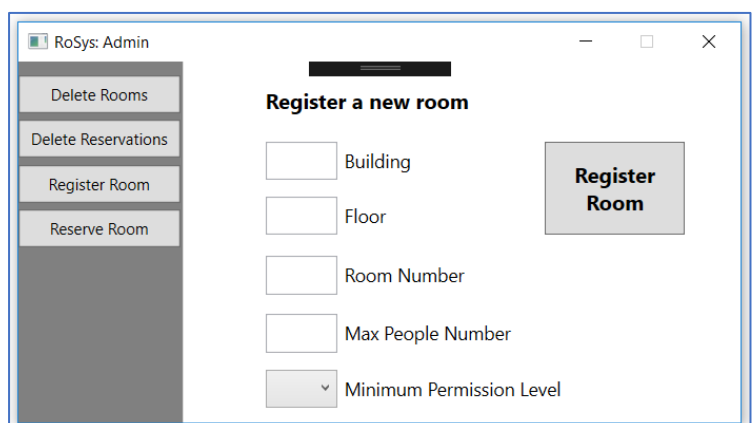
Picture 3



Picture 4

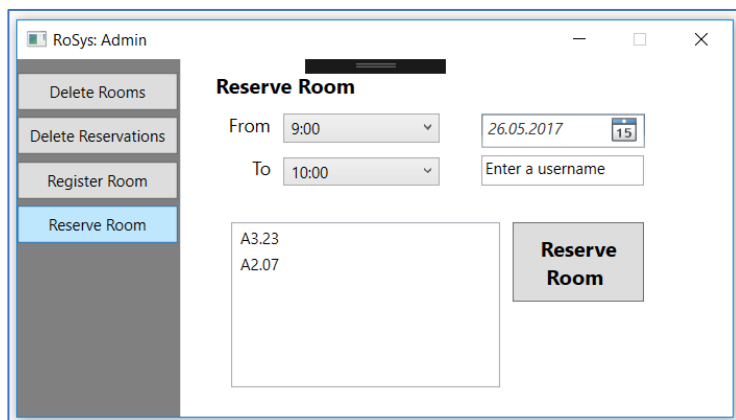


Picture 5



Picture 6

reservation to be in. If everything is done right, then a message will appear, like in Picture 9.



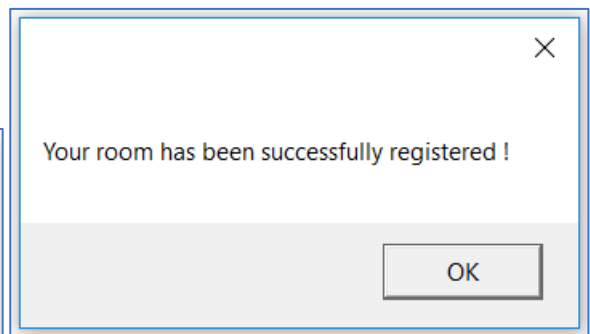
Picture 8

By pressing the “Student” button in the menu, Picture 1, the “RoSys: Student” window pops up, Picture 10.

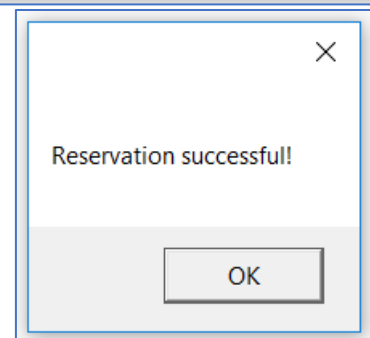
The “Reserve Room” button opens a page with four parameters to

reserve a room: “Number of people”, “Select a date”, “From” and “To” (Picture 11). The number of people that will be in the room needs to be typed in the textbox. The “Select a date” box is automatically set to today’s date if not changed manually by clicking on calendar icon and selecting a specific date. The “From” and “To” drop downs are made to select the time that the reservation will start and the time it will end. If any of the fields are still empty, then a message should be shown, but it has not been implemented yet. If a student user tries to reserve another room at the same time and date, the room will not be reserved and a message will be displayed, Picture 12. If there are available rooms with matching parameters as inputted, then a message saying that the user has been assigned to a room, as shown at the bottom of Picture 11, if there are no available rooms, then a message saying that there are no available rooms will appear, Picture 13.

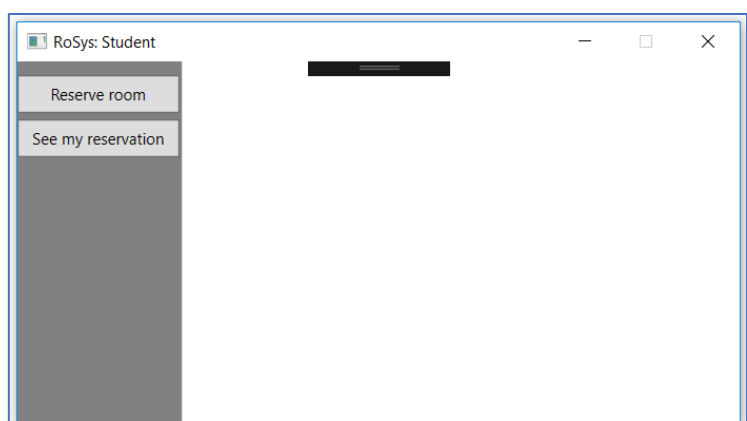
To see the reservation the student user has made, he or she needs to press the button, “See My Reservation”, Picture 14. One list item shows one reservation. A reservation is shown as the room it is in, the date, the time and the user



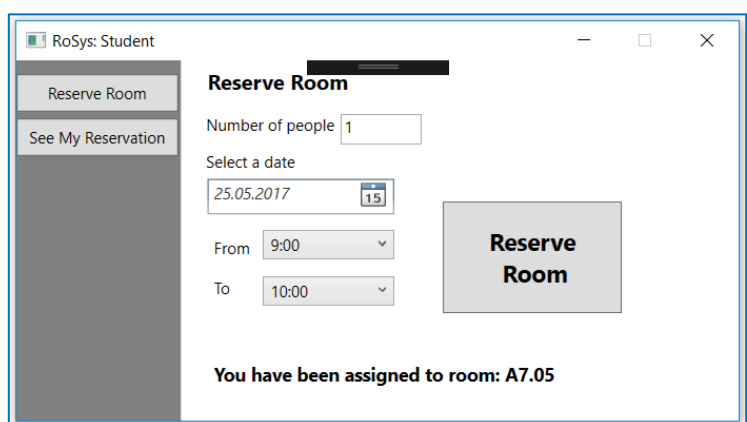
Picture 7



Picture 9



Picture 10



Picture 11

**You already have a room booked, at this time.**

Picture 12

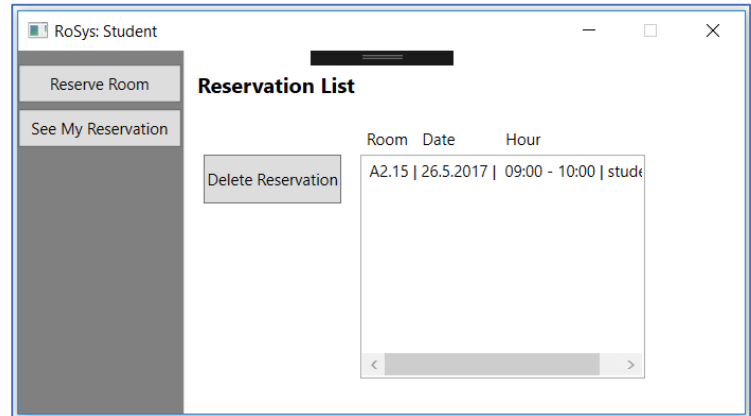
who made the reservation. If the user no longer needs the reservation, then he or she can select a reservation and press the button, “Delete Reservation”. This will delete the reservation from the repository and database.

**No rooms available**

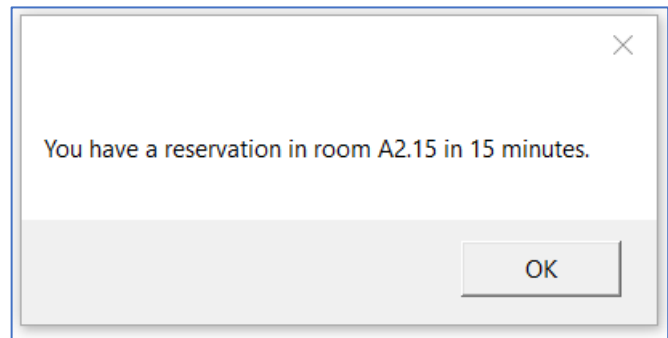
Picture 13

As we mentioned earlier, the teacher user has the same functions as the student user, but teachers can reserve rooms with teacher permission level, which student users are not able to reserve.

The system also notifies the user if they have a reservation in 15 minutes, Picture 15. By pressing button “OK” message box will be closed.



Picture 14



Picture 15

## Code

This section will cover some of the features of the code.

### Core: DAL Facade

This class is like a bridge between our DAL and Core. It also converts things from database strings into objects and other way around.

#### Delete All Users, Delete All Rooms

Sends method in DAL users to delete all users.

#### Insert Room, Insert User, Pass Reservation To DAL

Sends method in DAL rooms, users or reservations to insert this room, user or reservation in database.

#### Get User, Get Room, Get Reservation

All three methods are same type, they just work with different types of objects. These methods is the ones that collects all needed data from other methods in DAL Facade and then returns user, room or reservation object to its caller.

#### Get All Users, Get All Reservations, Get All Rooms

These three methods similar like previously are just combining information and then returning lists of user, room or reservation objects from database and DAL.

#### Convert From Strings To User Objects

This method creates user objects from string dictionaries. It also before creating object converts string into integer number so in Helper Functions later this number could be converted into permission level. And adds them to list, which will be returned to caller.

#### Convert From Strings To Room Objects

This method similar than previous also creates objects from strings, stores them in list and returns. As room parameters are mostly integers, they are needed to be converted. Also building names in Lillebaelt Academy are called by only one letter, so conversion is needed to be made from string to char.

#### Convert From Strings To Reservation Objects

This method gets string of foreign keys from database – username and building, floor and number. As it doesn't have any user or room object but reservations need them, we have created dummy room and dummy user. By checking foreign key strings with usernames and building Ids in repositorys (Equals methods) we can find the real room and user and together with other parameters add them to reservations object and then to list.

#### Delete Reservation, Delete Room, Delete User

Sends method in DAL users, rooms or reservations to delete user, room or reservation from database.

## Convert From Reservation Object To Strings

### Core: Helper Functions

Used to keep various generic functions that are used throughout the core project.

#### Convert Int To Permission

ConvertIntToPermission method converts with switch integers from database into permission level enums – Student, Teacher, Admin.

#### Time Collides

Method checks if time from reservation collides with start or end time.

### Core: Initialize

#### Core: Reservation

#### Core: ReservationRepository

#### Core: ReservationsObserver

#### Core: Room

#### Core: RoomRepository

#### Core: RosysThreads

#### Core: System Settings

The System Settings class, is currently used to define the “environment” the program runs in.

We set up 3 levels of environment, but only uses 2: Development and Production currently has no differences, but Test affects which database the connection runs to.

#### Update System Environment

This method updates the Environment Variable in the DAL. It is run when the Env property is set, and under Initialize.

### Core: User

#### Core: UserRepository

## UI.GUI: LoggedIn

Is the class that stores which user is currently logged in. It is used by the GUI to keep track of which user was initially selected.