RoSys Program Manual

# Introduction

This manual is a guide, that should help clarify some of the ideas we used throughout the program. It should 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, where 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.

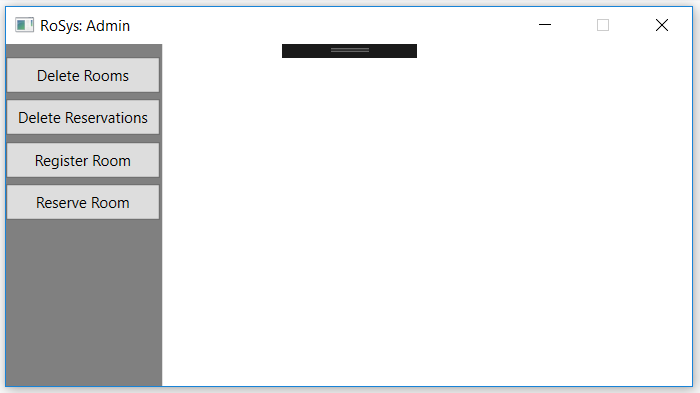
This focus on being able to integrate the program, as well as wishing to demonstrate the programs functions, on its own, was a greater challenge than expected.

# The User Interface

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

## Initial Run

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

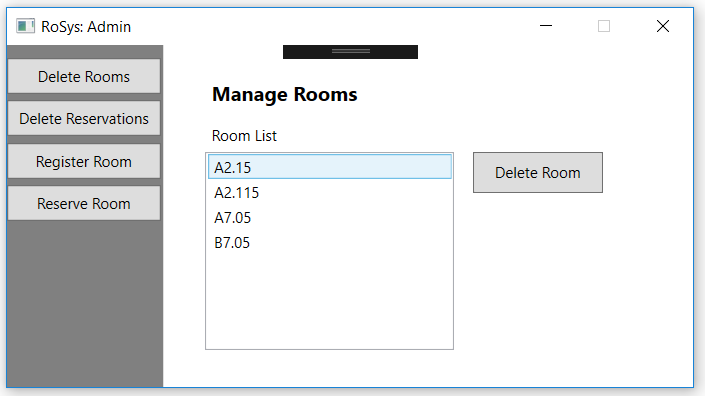
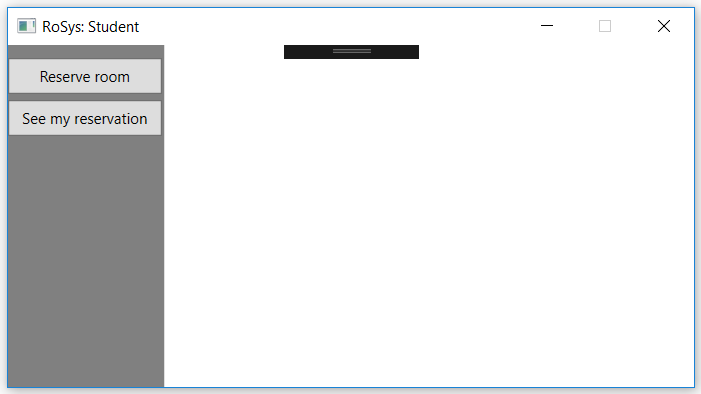
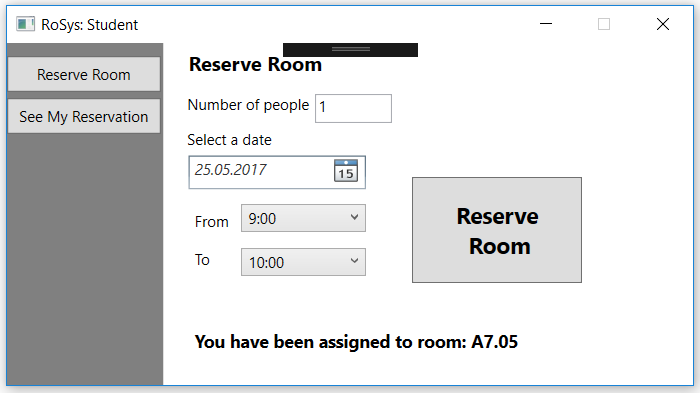
The different types of users have some different functionality. Student and Teacher generally have the same features, but Teacher have access to more rooms.

Picture

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

By pressing “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 repository and database.

Picture

From menu in Picture 1 by pressing “Student” button pops up window RoSys: Student, Picture .

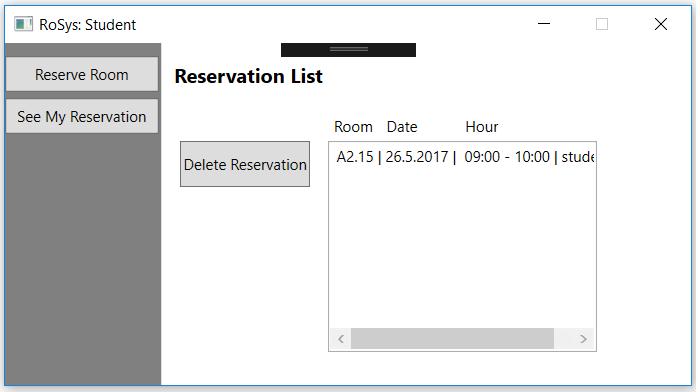
Picture

“Reserve Room” button opens page with four parameters to reserve room: Number of people, Select a date, From, To. The number of people that will be in the room are needed to be typed in the textbox. The “Select a date” automatically is set to be today if not changed manually by clicking on calendar icon and selecting a specific date. “From” and “To” are made for selecting time from which reservation will start and to what time it will last. If one of empty fields are still empty an exception will be thrown and will show a message like in Picture. If a student user tries to reserve one more room at the same time and date, other exception will be thrown, Picture. If in the system there is free room with matching parameters as inputted, a message will be displayed, saying that user has been assigned to room as shown in Picture, if not, a message will appear, that there are no free rooms.

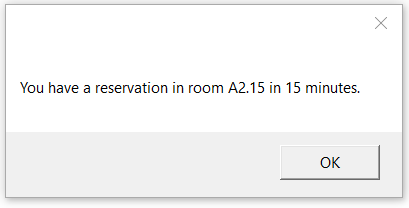
To see the reservation that student user has made he or she needs to press button “See My Reservation”, Picture. One list item shows one reservation starting with room that’s reserved followed by date, hours and the username of the user who reserved room. If user doesn’t need reservation, he or she can select specific reservation and press button “Delete Reservation”, the reservation will be deleted from repository and database.

Teacher user as we mentioned earlier has the same buttons and messages, only difference is that teachers can reserve rooms with teacher permission level, which users cant as their level is smaller.

Every time there is reservation in 15 minutes a message box will appear with message saying that reservation will start after 15 minutes, Picture. By pressing button “Ok” message box will be closed.

­­

# Code

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

## 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: HelperFunctions

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

# 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.