ECE-2524
Introduction to Unix for Engineers
Project: Room reservation application

Group: *Zeid Ayssa,Kaleb Dufera, Michael Surratt, Alazar Hailemariam, Brian Chen*

Date: 9/25/12

Summary

Our application is a booking manager for rooms on the Virginia Tech campus. A drop-down menu reveals a listing for all buildings on grounds, and a second drop-down menu allows users to select a room number therein. After specifying date and time, a simple click of a button reserves that room for that specified time slot. The application does check for errors and parameters, rejecting a reservation due to invalid time frames and previous bookings. After a successful booking, the application assigns that booking a number that we can use to reference it later. This is mainly to keep track of our bookings, and allows us to cancel reservations by entering it into a box we provide below. The entire system is simple, yet practical.

Unix Philosophy

Several rules from the Unix philosophy served as general guidelines during the development of this application. Once the fundamental tasks of the program were identified, they were broken into individual functions per the rule of modularity. Further, all functions were implemented independent of the GUI following the rules of extensibility, composition, and separation. Additionally, the rule of least surprise is evident in the GUI which is straightforward and easy to use.

Implementation

The work is divided between the group members. Zeid and Michael were responsible for debugging and GUI interfacing, Alazar and Kaleb were responsible for code implementation, and Brian was responsible for collectively making the GUI work with the actual code implementation. The program uses various Qt classes such as QtDate, QtTime, QtString, and QtStringlist. The GUI is designed in a way that the user must specify the date, time, building, and room number. All this information must be provided in order to reserve the room. After the user completes all the required fields and presses the “Reserve room” button, all the information is written to a “reservation.dat” output file and after each user input is checked for any date, time and room conflicts. If a conflict exists, the application prevents the user from reserving a room and thus the user will not get a reservation number. The final stage of the application is the confirmation number. This indicates that the user has successfully reserved a room. The application also lets the user delete any reservations that previously made. This is done by entering the reservation number and pressing the “Delete reservation” button.

Conclusion

The application performed according to the design specification. The most difficult aspect of this project was combining the design approach of each team member into the final product. Each team member had different implementation methods which required the group to develop code that was well encapsulated and modular. Overall, the result was successful and breaking the project into different parts and assigning it to different group members reduced errors and increased the functionality of the application.