1. Insert Deliverables Here

B.

Our system was programed in C# Using the windows forms in visual studio. The system currently only runs on windows desktop computers. The information is currently stored in three text documents: reservables.txt, reservations.txt, and users.txt. The system currently only supports the main functionality and has a rudimentary GUI.

The functionality that the system currently supports includes administrators being able to manage reservables and users. It allows all users to check if a reservable is available for a specific time and duration, view and delete their current reservations. When managing reservables the administrator can create, delete, or edit reservables. When creating a computer reservable the admin can select what room to put it in. the admin can also move computers from one room to another. When managing users, the admin can create, edit, and delete users. The admin must supply the user type, name and email. The admin can also edit any of these fields for any user. When searching for reservations, only the reservables for that specific time appear (no browsing reservations).

The GUI currently is very basic. The reservables and users are only displayed by there ID number with no other information to signify what the object is. Reservations are displayed with basic information but not formatted well for user reference. The GUI is not resized or customized for different screens resolutions or window sizes.

Partially complete: (future improvements if we continued working on it)

Future improvement ideas or features that are yet to be implemented include a check-in/check-out of reservables so currently checked-out reservables can be viewed and if a reservation is ended early, the reservable can be checked out by someone else earlier. This would also include “walk-in” reservations for quick access. Another not yet implemented feature that was planned was displaying the type of reservable and location and information when selecting a reservable to reserve. This would also extend to users when the administrator is editing users in the database. Speaking of databases, the entire system would use databases to store the users, reservables, and reservations instead of using text files and reading from and writing to them on startup and shutdown respectively. A third feature would have included removing all reservations from the database when the reservable they were made for had been deleted. This would also send a message to the holder of any reservation of that reservable to notify them that the reservation that had been cancelled so they could reserve a different room or computer. A potential improvement on our current design could include adding laptops to the reservable items on a daily or an hourly rental schedule.

Some ideas: check-in and checkout of reservables, students only being able to reserve computers, deleting reservables deletes reservations, browsing reservations, admin being able to manage all reservations, users being able to see currently in use reservables, GUI resizing, displaying more info besides just the id number for reservables and users , more feedback for user.

C.

|  |  |  |
| --- | --- | --- |
| Team Member | Total Time | Activities |
| Alex Bisbach |  |  |
| Ben Mehn |  |  |
| Evan Gjerde |  |  |
| Jun Jeong |  |  |
| Levon Swenson |  |  |

D. Some ideas: finding times to meet, organizing workloads, people not knowing C# before starting this project

One major difficulty was learning how to use C#. Some of us had never used C# before, so learning the differences between C# and C++ was sometimes a challenge, especially when it came to public and private constructors giving us problems with not being able to create things and us not knowing why for a bit.

Another difficulty was the inevitable finding time to meet. Matching up our schedules was difficult at times due to other class projects, work, and other obligations. This was solved by meeting to get started and distribute work to do when each of us had a chance. This allowed us to communicate the ideas and things we wanted to implement while still being able to work on the program when each of us did not feel rushed or like we should be working on something that was more urgent at that time.

This also allowed us to distribute the workload more evenly. This was a problem at the beginning of the project and sometimes lead to one person feeling overwhelmed or too busy, but the problem was more or less solved by the time the final 2 milestones were worked on and turned in.