Hotel Management System

Stephen Gung

**User Manual**

As any user:

* Change Password/Password – The user may change his/her password for this application
* Log Out/Log – When the user is done interacting with the application he/she may log out (although not necessary, it is recommended)
* File – User can open or save a file. Note: Guest users are denied this action; nothing will happen

As a Guest:

* Editing Information – The guest may edit his/her information such as the username he/she uses for this application
* Services
  + Room Service – user may order room service if desired. Once the user has chosen his/her desired meal the application will display a confirmation ( showing the type of foods the user has chosen, the price for each individual food, and then the total cost)
  + Maintenance – user may request maintenance service if a piece of equipment needs to be replace (for reasons such as being damaged).
  + Room Change - guest may request a room change, however it needs to be approved by (either) the employee or manager
  + Credit Card Change – guest may decide to use a different credit card number for purchasing use and such; this needs to be approved by (either) the employee or manager

As an Employee:

* Implementations as a Guest – The user is capable of performing all functions the guest can, with the exception of changing rooms and changing the credit card number. These functions however will be targeted for a guest
* Check In
  + New Guest – user can check in a new guest, the application will require the user to fill out all information necessary in order to add this guest into the system. Note: Usernames for this application are unique, the application will check for this (as well as other things)
  + Returning Guest – When a guest is checked out the application will store their information in case if the guest decides to come back. This save time; requiring only for the user to select the correct username ( first and last name are mentioned in case) and a room to check into
* Check Out – The user may check a guest out; the application will display a list of all the current guests for the user to select
* Requests – This function will appear ONLY if a guest has made a (room change and/or credit card change) request. Once a request has been accepted or denied the corresponding guest will be notified the next time they log in. Note: User does not have to immediately accept or deny Requests and may complete it later

As the Manager:

* Implementations as an Employee – The manager is capable of performing all functions the employee can
* Editing Information – The manager may edit an employees’ information. This function is identical to editing a guests’ information
* Employee
  + Add – The manager can hire a new employee; the application will require the user to fill out all information necessary in order to add this employee into the system. Note: Usernames are unique; the application will check for this
  + Remove – The manager can decide to fire an employee; the application will display a list of all the current employees for the manager to select
* Menu
  + Edit – The manager may edit/remove items from the current food menu
  + Add – The manager may add a new item onto the food menu

**User Interface Design**

As with many applications risks are always possible. Many of the functionality in this application requires the user to complete it (rather than having its’ own independent window) before proceeding. This prevents any possible overlap of data.

A scroll pane is added onto the main panel; since not all screens have the same size. (However in reality, If a hotel were to have this kind of application they would most likely have provide computers, these computers would most likely be roughly the same size)

This application uses colors to denote specific things in the hotel. For example, open rooms are white, while an occupied room will be colored in purple.

Shortcuts are added for some key functions (such as logging out; Ctrl L). These shortcuts are designed to save time. Note: Those with shortcuts are mentioned within the application one way or another (example: hovering over the tabbed panes)

All data regarding editing information is updated on the fly; there is no linear sequence. This is designed to save time

Many key functions are located at the corners; this is designed purposefully because it is easier to locate something at the edges of the screen rather than somewhere in the middle

When logging in as different types of users many of the implementations are located in generally the same area. For example: when logging in as a guest, the ‘Guest Info’ box is located at top of the right panel. When logging in as an employee or manager, the ‘Guest Info’ box will be located at the same place. This is designed for the purpose of users being able to be familiar with an application; If each layout is different for every user then it forces the users to spend time figuring out where each functionality is located. (However, in reality a person would never really see the three types of layouts; a guest would never sign in as an employee, and so on. So this design principle is really just for testing (the application) purposes)

Right clicking features (within the hotel’s layout) are also added. These features are designed to save time. Right clicking features also depend on where the user right clicks; right clicking within the Café will display options different from right clicking the Front Desk. Right clicking features also depend on what type of user. Note: these features have functions that are identical with the buttons with only one difference; for example: if the user wanted to check in a new guest, checking in a new guest via button click will require the user to select a room, while right clicking within an empty room will set the room location to the room that was right clicked.

**Note: This is for when testing the application**

When the application starts there are already some guests and employees added. These can be deleted/changed when running the application. They are added so the user/tester doesn’t have to keep adding people initially

Guests:

* Username, Password
  + jdoe, pass1
  + jsmith, pass1
  + mfenix, pass1

Employee:

* Username, Password
  + bjamin, emp
  + tclaire, emp

Manager:

* Username, Password
  + manager, 123

Possible extra credit:

* Shortcuts, also note that when in the Log in screen the user can press enter (rather than pressing the ‘Log In’ button)
* Returning Guest implementation
* Guest and Employee tables (for viewing purposes)
* Open and save individual file(s) implementations
* The Request and Notifications implementation, as they require signing in and out to see results; it is actually responsive in a real practical environment