# Proposal

Goal: Create a light, free, and open-source, user-friendly interface java program that allows customers to purchase items and administrators to manage their hotel. It will be posted on GitHub and fit any system that contains JDK version 22 and above. Customers should easily check the room's status and status. They will have access to add or remove items and rooms to shopping cart and check out. They can check in or check out through this program. If needed, they can upgrade their rooms, check their own rooms’ status, or refund. If they have a limited budget, they can use a program that will maximize the value of items they purchase. Administrators will be able to log into admin mode with specialized passcode. They can add or remove any rooms and properties associated with these. They will also be able to check daily income through this window. They will have access to managing staff, including wages, working hours, and information. Mainly focusing on providing customers with a good purchasing experience.

Costumer Section:

1, Rooms: To make it simple, use small squares with different colors to represent its status on GUI. It will be great to use hash map to store their information (Similar to unordered map in C++ or dictionary in Python). Users might be able to sort and filter them with a research bar. When they click on it, it will open a window, or dialogue box and bring out their details and features.

2, Purchase: It will be easier to use array list to create a shopping cart. Add remove button to remove item and add button to add items to the cart. When user clicks on the extras, it will lead them to extra items (food, accessories, movies etc.). When users purchase the room, they will enter the days they will live in, estimated check in and check out date and time, and their information. The checkout button will automatically calculate the final cost and show their items. (This page will be extensive, so administrators can add any payment method they like). A search bar or tags selection may be included to allow users to search for what they need. If a refund is requested, a lower percentage can be refunded as the time passes.

2.1: Additional budget method: To maximize the value, use dynamic programming or greedy to list and calculate best solution. (this is for the extra) User can add it to the shopping cart.

3, Upgrade rooms, check in, check out: User can log in with the information they provided to check in and out. It will jump to another window that allows them to click buttons to check in and out. If they want to upgrade the room, they can search for other rooms and an additional fee will be required. It will be added to the shopping cart.

Administrator Section

1, Passcode: With a special input window, administrator will be able to input the special passcode to log into the administrator mode.

2, Modify rooms: Administrator can directly modify the key-value pairs stored in HashMap. This button will open the file where HashMap is stored. Pop-up windows will appear to make adding and deleting easier.

3, Check income: Pop up a window that will show daily income. (Additional: Use graph to represent it)

4, Manage Staffs: As administrators log in, they can modify staffs’ properties. Frequently changed values (e.g. wages and working times) will be stored in HashMap. The same way as rooms, adding and removing features will be added to make hiring and firing easier.