

Author of all use cases is George Costas

## Hostel21 Use Cases, Version I:

- UserAdd
  - Goal – Read in user input and create a user based on the required fields.
  - Actors- Administrators are the only users
  - PreCondtions- program Compilation
  - Triggers – If user hits the enter key after desired input is requested.
  - Course of Events- First enter in the desired user information in the order specified in the readme file. The user is then given a user id, which will never be used for other users. Finally the user information will be placed and added to a list of users.
  - Alternative Paths- The user may enter (first name, last name and email,) and/or (cc#, expiration date, security code, and phone number). This allows flexibility with giving away financial information.
  - PostCondtions- Increment user id so it is unique.
  - BusinessRules- a user must enter first name, last name and email at least.
  - Notes- this use case is implemented
- UserChange
  - Goal – Read in user input and change a user based on the required fields.
  - Actors- Administrators are the only users
  - PreCondtions- program Compilation, and the list of users must contain at least 1 user with a matching user id
  - Triggers – If user hits the enter key after desired input is requested.
  - Course of Events- First enter in the desired user information in the order specified in the readme file. If the user id matches one in the list of users, then the input information will overwrite the information for each relevant field.
  - Alternative Paths- no alternative paths.
  - PostCondtions-None
  - Notes- this use case is implemented. Users should be sure to rewrite the correct information into each field; otherwise these fields will be overwritten.
- UserView
  - Goal – Read in user input to display a users information.
  - Actors- Administrators are the only users
  - PreCondtions- program Compilation, and the list of users must contain at least 1 user with a matching user id
  - Triggers – If user hits the enter key after desired input is requested.
  - Course of Events- First enter in the desired user information in the order specified in the readme file. If the user id matches one in the list of users, then the correct user and their information will be displayed.
  - Alternative Paths- no alternative paths.
  - PostCondtions-None
  - Notes- this use case is implemented.

- AdminLoad
  - Goal – Read in user input and extract relevant hostel and bed information from a xml File.
  - Actors- Administrators are the only users
  - PreCondtions- program Compilation, presence of the file at inputted file path.
  - Triggers – If user hits the enter key after desired input is entered.
  - Course of Events- First enter in the desired command in the order specified in the readme file. If this list of commands contains a valid file path then the program will read each line of the file and parse it's value into the appropriate object. Before storing the data each bed must be given a bed id and each hostel a hostel id. Finally these objects will be added to either a list of hostels (if the object is a hostel) and a list of bed (if the object is a bed)
  - Alternative Paths- None
  - PostCondtions- increment the hostel and bed ID values so the IDs are unique.
  - BusinessRules- must capture information on the hostels smoking, alcohol, deadline, and penalty values.
- AdminRevenue
  - Goal –view the revenue of a specified date range. The revenue is based off of money earned in a hostel and is not altered beyond adding more revenue.
  - Actors - Administrators are the only users
  - PreCondtions- program Compilation, Bookings must exist in the list of current or past bookings.
  - Triggers – If user hits the enter key after desired input is requested.
  - Course of Events- enter the correct command prompt and the list of pasty and current bookings will be searched. If the list contains bookings that have bed/dates matching the dates requested, then add the revenue for those bed/dates. Finally display this revenue.
  - Alternative Paths- The user may enter one date or a star and end date.
  - BusinessRules-
  - Notes- this use case is implemented, but it could be improved upon by controlling the property of revenue in more complex ways. For example have revenue for profits, and a revenue for total money earned.
- AdminOccupancy
  - Goal – look through the bookings made and display the percentage of rooms booked.
  - Actors- Administrators are the only users
  - PreCondtions- program Compilation
  - Triggers – the list of bookings must have at least one booking.
  - Course of Events- First enter in the desired user information in the order specified in the readme file. This command will then iterate through the

list of current bookings and divide that number by the beds available in all hostels. The result of this arithmetic is displayed.

- Notes- this use case is implemented. But could be significantly improved via occupancy per hostel, or date.
- SearchByBed
  - Goal-Ideally this search goes through a list of hostels and for each hostel it goes through their rooms and if the rooms contain the same amount or more beds than the user requested it displays those rooms and gives them search id's.
  - Actors- Administrators are the only users
  - PreCondtions- program Compilation, the list of hostels and rooms must be loaded.
  - Triggers – If user hits the enter key after desired input is given.
  - Course of Events- First enter in the desired city to search in, then give a start date, an end date and a number of beds desired. The search will go through the lists and see if any rooms containing the required amount of beds, in the date range exist. If these beds do exist they will be given search ID's and then displayed.
  - PostCondtions- Increment search id so it is unique.
  - BusinessRules- show the cheapest combinations first
  - Notes: this use case is not currently implemented, I had trouble adding the rooms as an object, and so my idea of how to perform the search could not be implemented. A hackneyed version is included in the code.
- SearchByDate
  - Goal-this search goes through a list of hostels, then their respective beds. If the date range specified in the command is greater than 1 day, it will display a list of the beds that are available on the dates requested in the proper city.
  - Actors- Administrators are the only users
  - PreCondtions- program Compilation, the list of hostels and rooms must be loaded.
  - Triggers – If user hits the enter key after desired input is given.
  - Course of Events- if the range of dates is greater than one, then search the list of hostels for hostels matching on city. Sort the list of beds by date. Then search the list of beds for all beds in that date range. If the beds match the date requested then add a search id to a list for the bed.. If the range of days is 1 then iterate over the list of hostels and then the list of beds if any of the beds in the list match on day then assign them a search id and display the beds.
  - PostCondtions- Increment search id so it is unique.
- BookAdd
  - Goal – assign a bed to a user and charge the user for the bed.
  - Actors- Administrators are the only users
  - PreCondtions- program Compilation, proper input
  - Triggers – If user hits the enter key after desired input is requested.

- Course of Events- the list of users will be searched for the matching user ids. A list of beds and the bed's list of search ids will be iterated through until the desired search id is found. Once the search id and user id are found, booking will be created, given an id and added to the list of current bookings. The bed for the booking will be removed from the list of beds in the hostel. The variable booked will be set on the bed and booking. Finally the user will be charged the price of the booked room and that charge will be added to the total revenue of hostel 21's records.
- PostConditions- Increment user id so it is unique.
- BusinessRules- make sure credit card information is available for the user making the booking. Make sure to charge the user.
- BookCancel
  - Goal –remove a booking and make the previously booked room available.
  - Actors- Administrators are the only users
  - PreConditions- program Compilation, proper input
  - Triggers – If user hits the enter key after desired input is requested.
  - Course of Events- the program will search through the list of current bookings to find the proper booking. If the id's match then the proper booking, will check that it is not after the deadline date (currently a poor implementation). If it is after the deadline then 75% of the price is retained. If it is before the deadline then the total revenue of the hostel21 company is reduced by the price. Either way the booking is set to not booked and it is then added back to the list of available hostels.
  - BusinessRules- must implement deadline policy and penalty
  - Notes- this use case is implemented, but poorly the functionality of the booking is correct but the functionality of the deadline, and penalty are reduced due to troubles figuring how to handle the date.
- BookView
  - Goal – search through a list of bookings to view a booking.
  - Actors- Administrators are the only users
  - PreConditions- program Compilation, proper input.
  - Course of Events- the list of current and past booking is iterated through, if the booking id in the command matches one of the booking id's in these lists then, display the booking
  - Notes- this use case is implemented