|  |  |  |
| --- | --- | --- |
| **Text cases** | **Excepted Result** | **Actual Result** |
| The system on load must initialize Hotel Objects and RoomType object, and perform action on them. | 1. On load the system should create an Hotel Array of size 4 and RoomType object of size 11, then load Hotel Array with data from hotel text file provided. 2. Load RoomType objects with data from RoomType text files provided. 3. Assign corresponding RoomType objects to their corresponding Hotel Objects Array. | 1. On load, system creates an Hotel Array of size 4 and RoomType object of size 11, then loads Hotel Array with data from hotel text file provided. 2. Loads RoomType objects with data from RoomType text files provided. 3. Assigns corresponding RoomType objects to their corresponding Hotel Objects Array.   **[Successful]** |
| The system must display menu and prompt user to input. | After initialization system should display menu and ask user to input their choice from the menu. Must validate that user’s input, if not display message and ask again to enter the choice. | The system calls for menu class constructor and display menu. Then prompts user to enter the option from menu (1 to 5). If they enter option beyond the range then, a message is displayed with “not valid menu option, please try again “. This does loops until user enters valid menu option. Once valid menu is chosen, then switch conditional operator navigates system to the corresponding task that needs to be done.  **[Successful]** |
| The user opts to display all hotel details. | A report should be displayed that, for each hotel, lists that hotel’s details (name, address, star rating), followed by the details of each of its room types (name, max occupancy, regular price and also the sale price if it differs from the regular price).  Return to main menu | Displays the report for each hotel with its details (name, address, star rating), followed by details of each of its room types and their details (name, max occupancy, regular price, sale price if different from regular).  Returns to main menu  **[Successful]** |
| The user opts to find cheapest room | 1. A report should be displayed with hotel’s details (name, address, star rating), followed by details of the room types (name, max occupancy, regular price and also the sale price if it differs from the regular price), that is the cheapest among other. 2. Should compare each hotel and its room to find cheapest 3. Return to main menu | 1. A report is displayed with hotel’s details (name, address, star rating), followed by details of the room types (name, max occupancy, regular price and also the sale price if it differs from the regular price), that is the cheapest among other. 2. Compares each hotel and its room types to get the cheapest among them. 3. Returns to main menu   **[Successful]** |
| The user opts to set a sale price for a particular room of a particular hotel | Should match the entered hotel name with the one in Hotel Array.   1. If not found, message should be displayed saying no match found, please try again. Should ask user to enter again until validated. 2. Once validate, name of room types should be displayed at that hotel as numbered list and prompt user to select one of them. This must be valid again else ask user until its valid input. 3. Then details of selected room (regular price and current sale price) must be displayed. Ask user for new sale price. 4. Sale price must be between 50% and 100% of regular price. Based on this valid user input new sale price and loop until its satisfies validation.   Then should display sale price updated message. | The entered hotel name with the one in Hotel Array is tried to be matched.   1. Message displayed saying no match found, please try again. Ask user to enter again until validated. 2. Problem with user hotel name with space (e.g. El Grandor) as sc.next( ) command escapes after space and match is not found even if there is hotel with that name.   **[Failed]** |
| The user opts to set a sale price for a particular room of a particular hotel | Should match the entered hotel name with the one in Hotel Array.   1. If not found, message should be displayed saying no match found, please try again. Should ask user to enter again until validated. 2. Once validate, name of room types should be displayed at that hotel as numbered list and prompt user to select one of them. This must be valid again else ask user until its valid input. 3. Then details of selected room (regular price and current sale price) must be displayed. Ask user for new sale price. 4. Sale price must be between 50% and 100% of regular price. Based on this valid user input new sale price and loop until its satisfies validation. 5. Then should display sale price updated message. 6. Return to main menu | [Fixed using sc.nextline() command]  The entered hotel name with the one in Hotel Array is tried to be matched.   1. Message displayed saying no match found, please try again. Ask user to enter again until validated. 2. Then name of room type is displayed of that hotel as numbered list and user is prompted to select one of them. This is again validated. If not then user is prompt to enter until its valid data. 3. Then details of selected room (regular price and current sale price) is displayed. User is asked for new sale price. 4. Sale price is validated based on condition. If it satisfies validation message is displayed as sale updated. 5. Else again asked to input the new sale price until condition is met. 6. Returns to main menu.   **[Successful]** |
| The user opts to find the result based on matching criteria | User should enter valid values for min Occupancy, min star rating and maximum price willing to pay.  System should search each Hotel and each room of hotel that matches the user criteria and have at least one vacancy.  Once found, these room should be grouped under each hotel they belong to with their details (name, max occupancy, regular price and also the sale price if it differs from the regular price) and hotel details (name, address and star rating) displayed.  Should return to main menu | User enters values for min Occupancy, min star rating and maximum price willing to pay.  Validation is done for each input as user inputs them. If any input doesn’t met condition, the user is again prompt to enter valid input for that input until the condition is met.  Then it searches the Hotel and each room of the Hotel that matches the user criteria and have at least one vacancy.  Once found, the matching room are grouped under the hotel they belong to with their details (name, max occupancy, regular price and also the sale price if it differs from the regular price) and hotel details (name, address and star rating) is printed.  Then, return to main menu  **[Successful]** |
| The user opts to exit the system | The system should exist and used would not be able to use system. | The system exists displaying message system existing.  User is unable to use the system.  **[Successful]** |

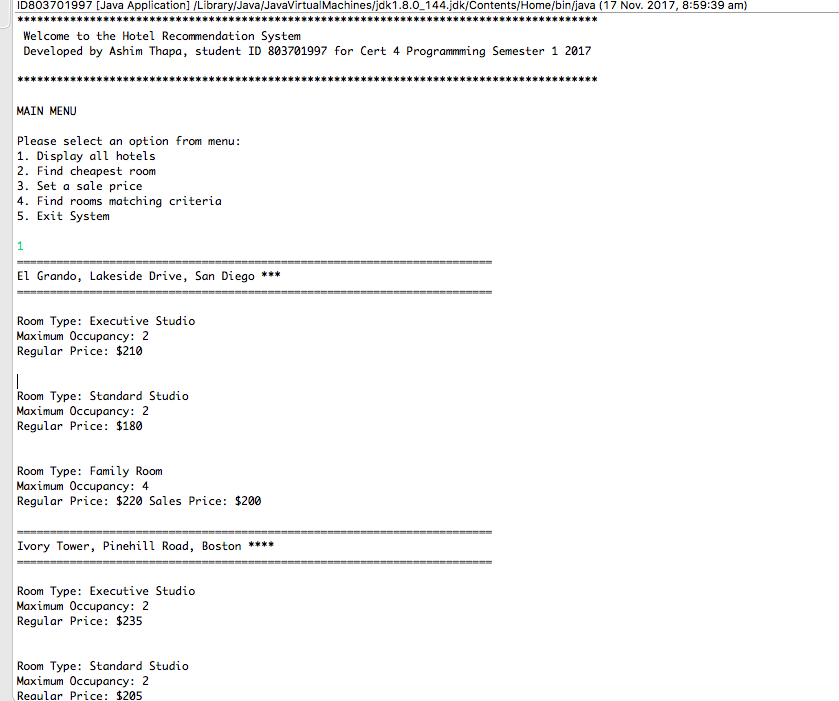


Fig1: Menu Displayed and user asked to prompt from the menu. User selects 1, all hotel is displayed

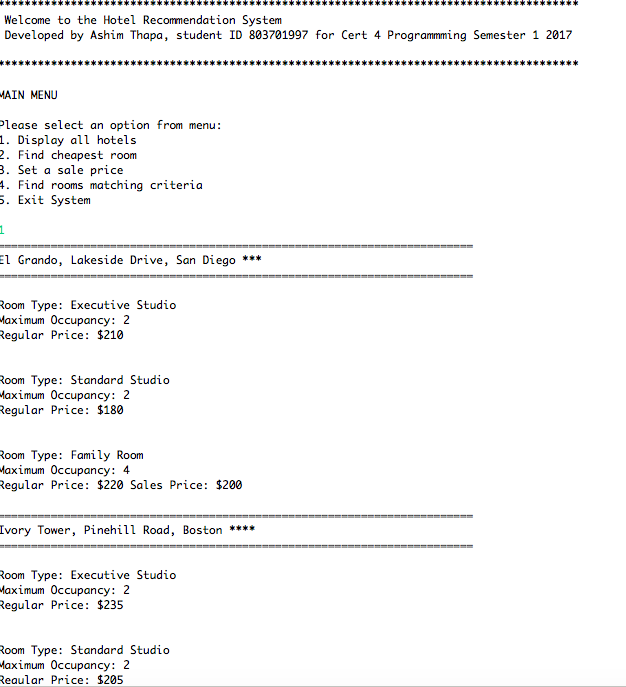


Fig2: continuation from figure1

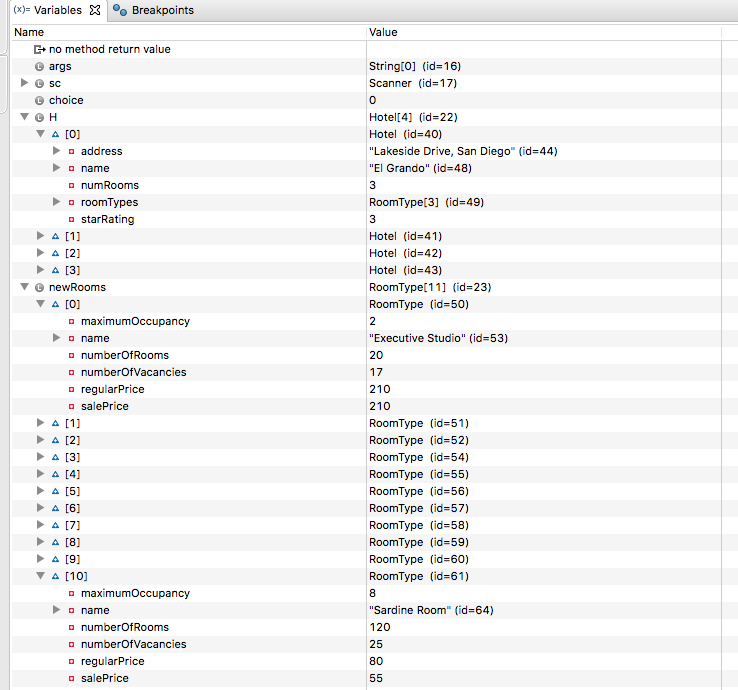


Fig3: Data loaded to Hotel Array (H) and Roomtype Array(newRooms) from text files.

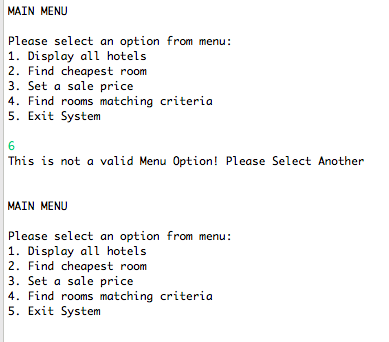


Fig4: Menu Validation

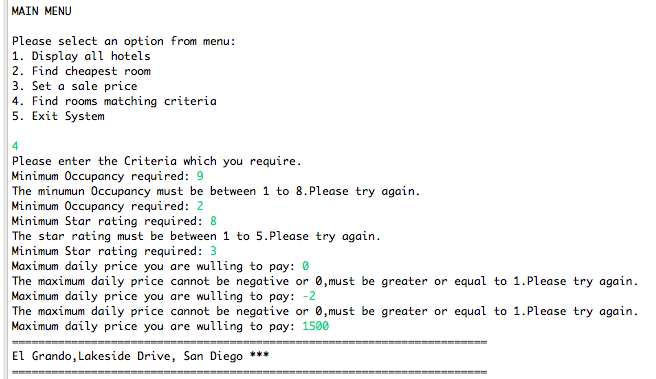


Fig5: Matching Criteria user input validation for each input

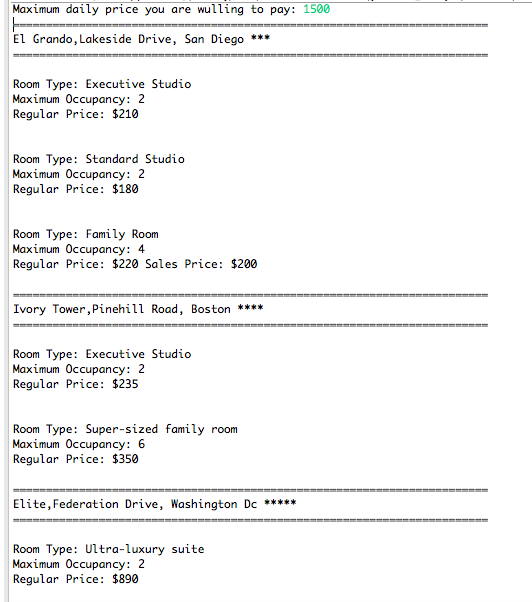


Fig6: Continuation from figure 5; after validation Hotel and rooms are displayed.

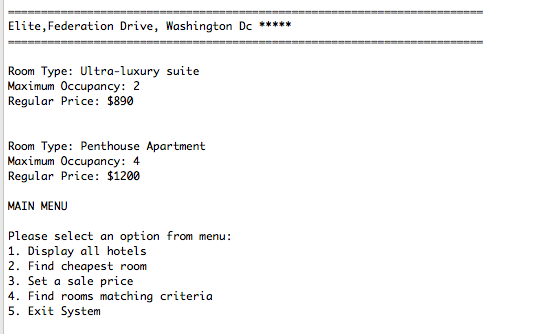


Fig7: Continuation for Fig6; that shows the remaining displays. (Control returns to main menu after successful task completion).