CSc 4350: Software Engineering

Fall 2016

Team Phoenix

Fergus Kelley

Andrew Truong

Ryan Ocampo

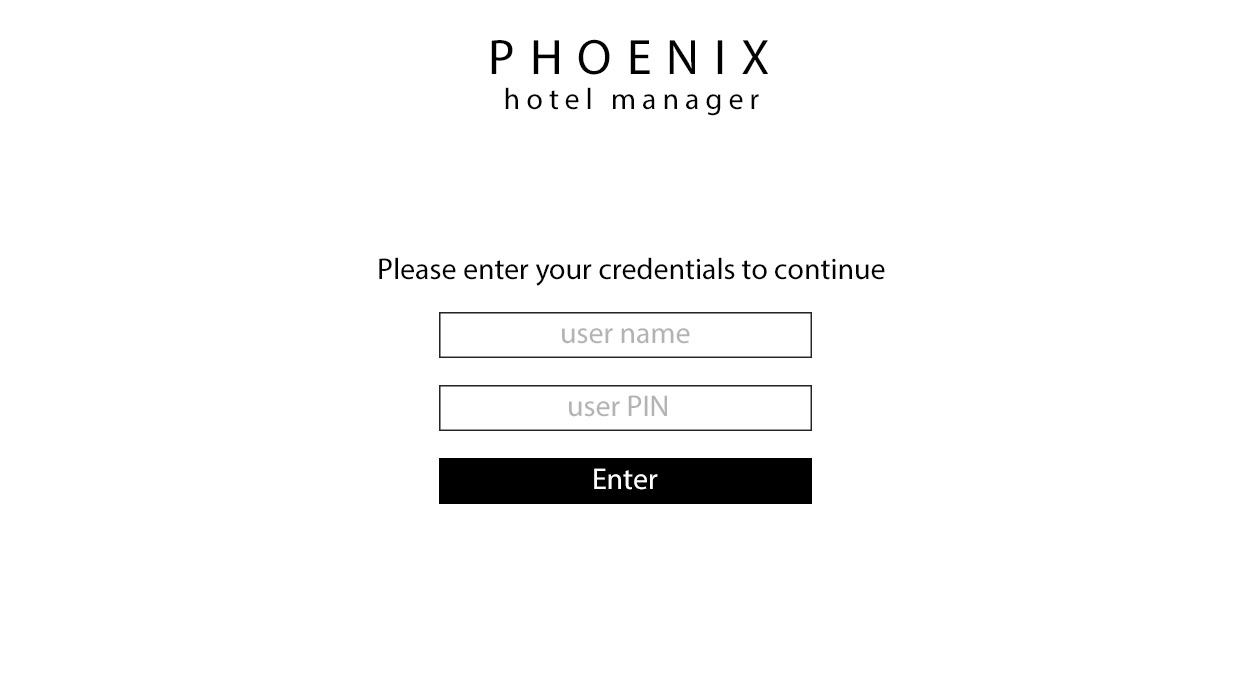
Deividas Rutkauskas

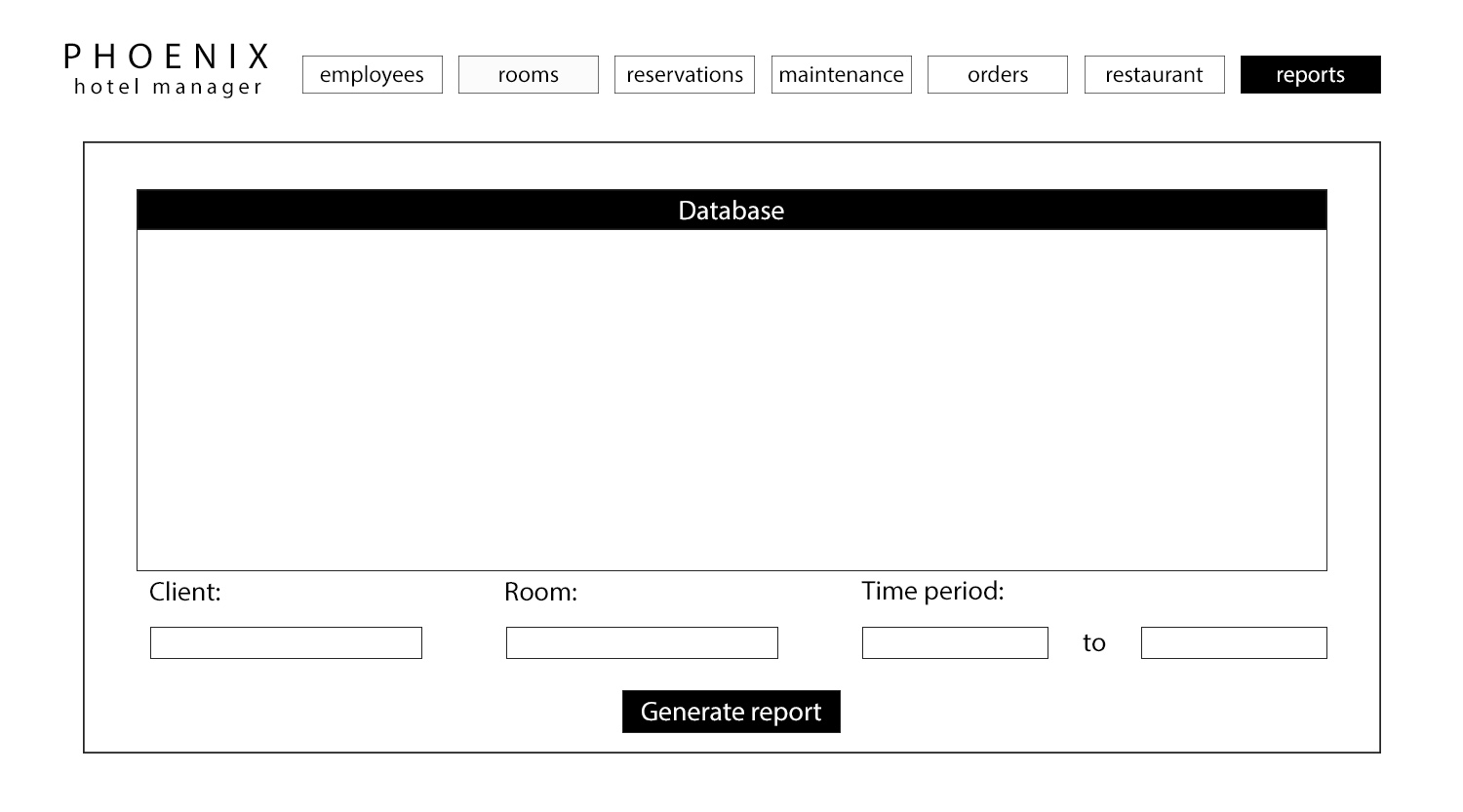
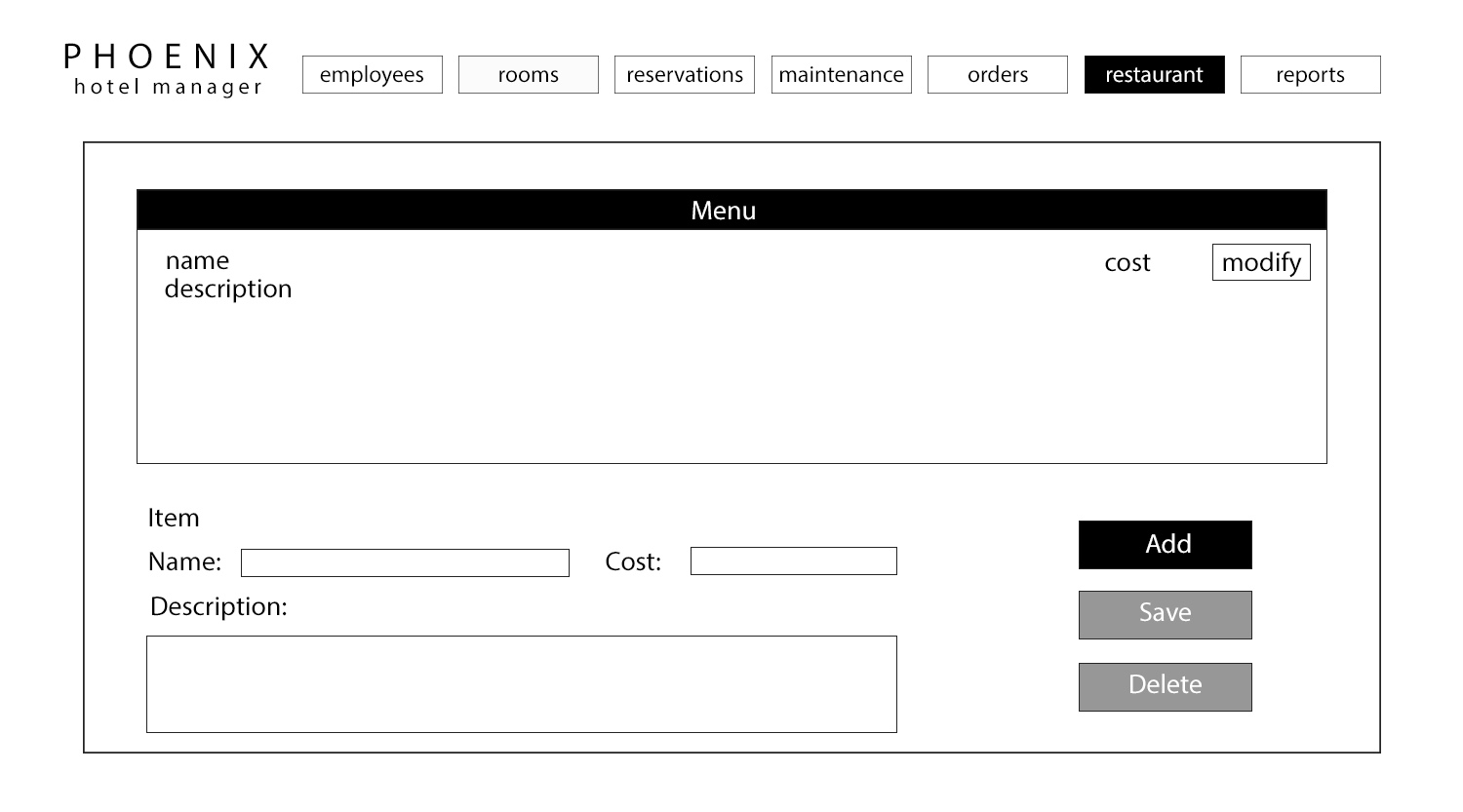
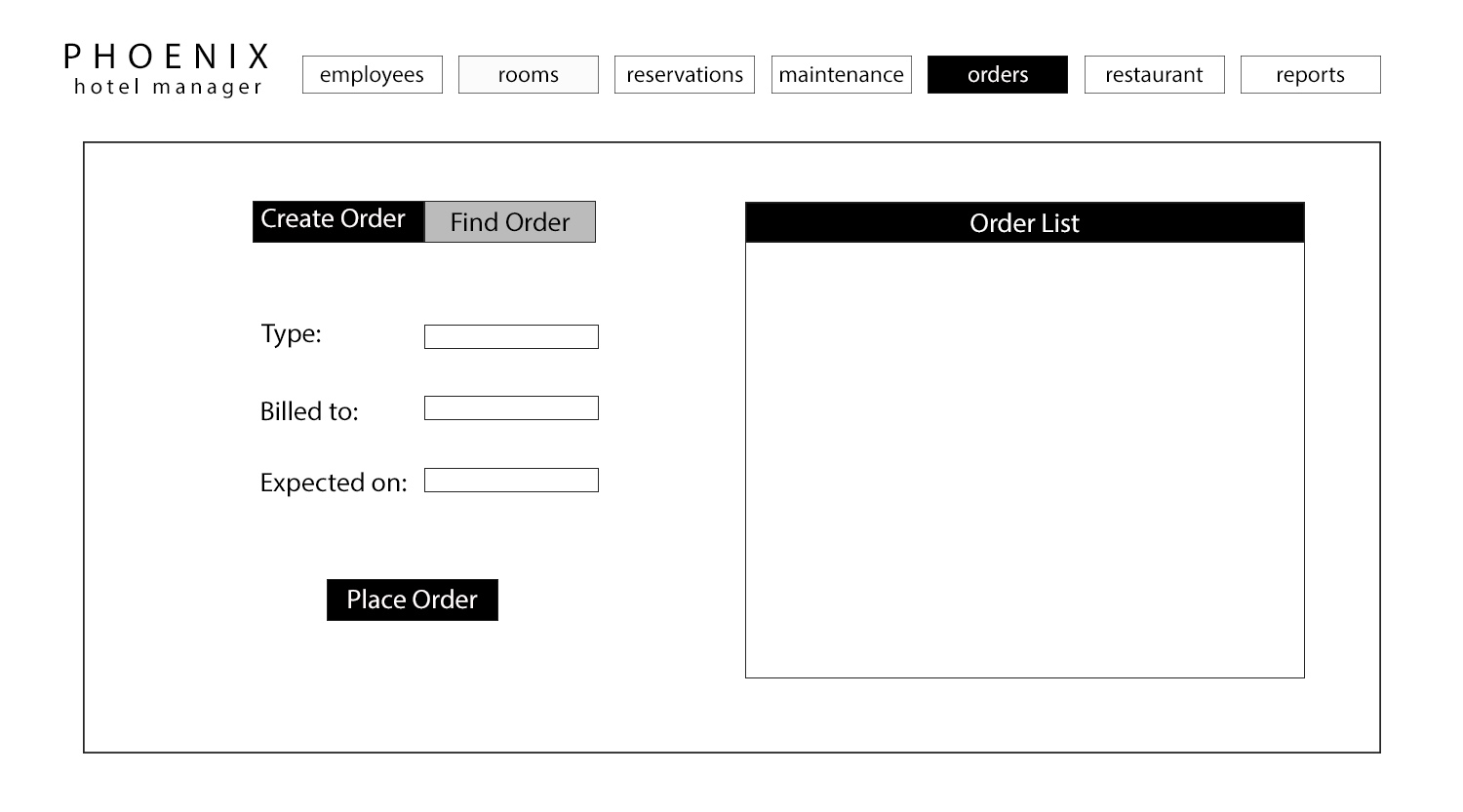
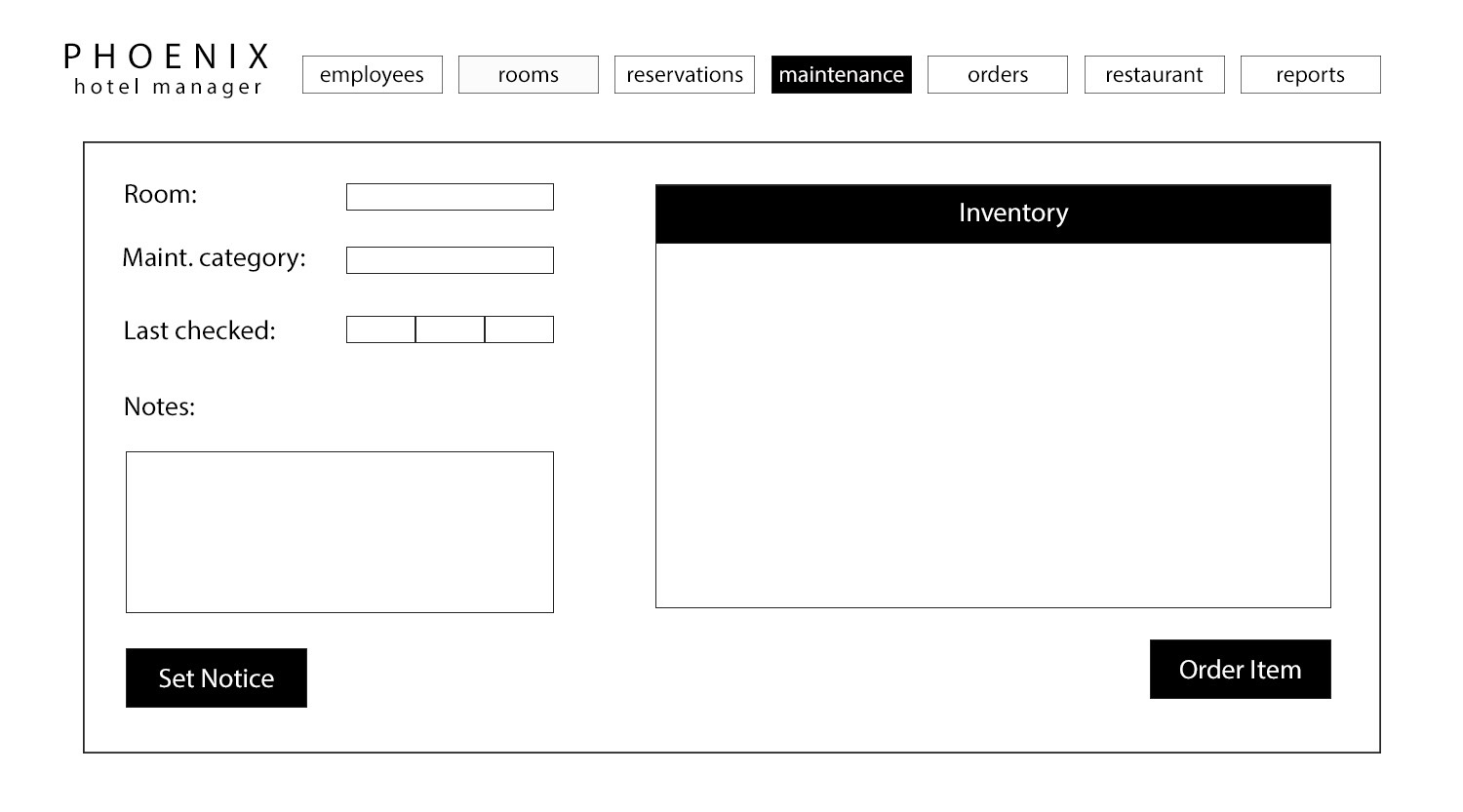
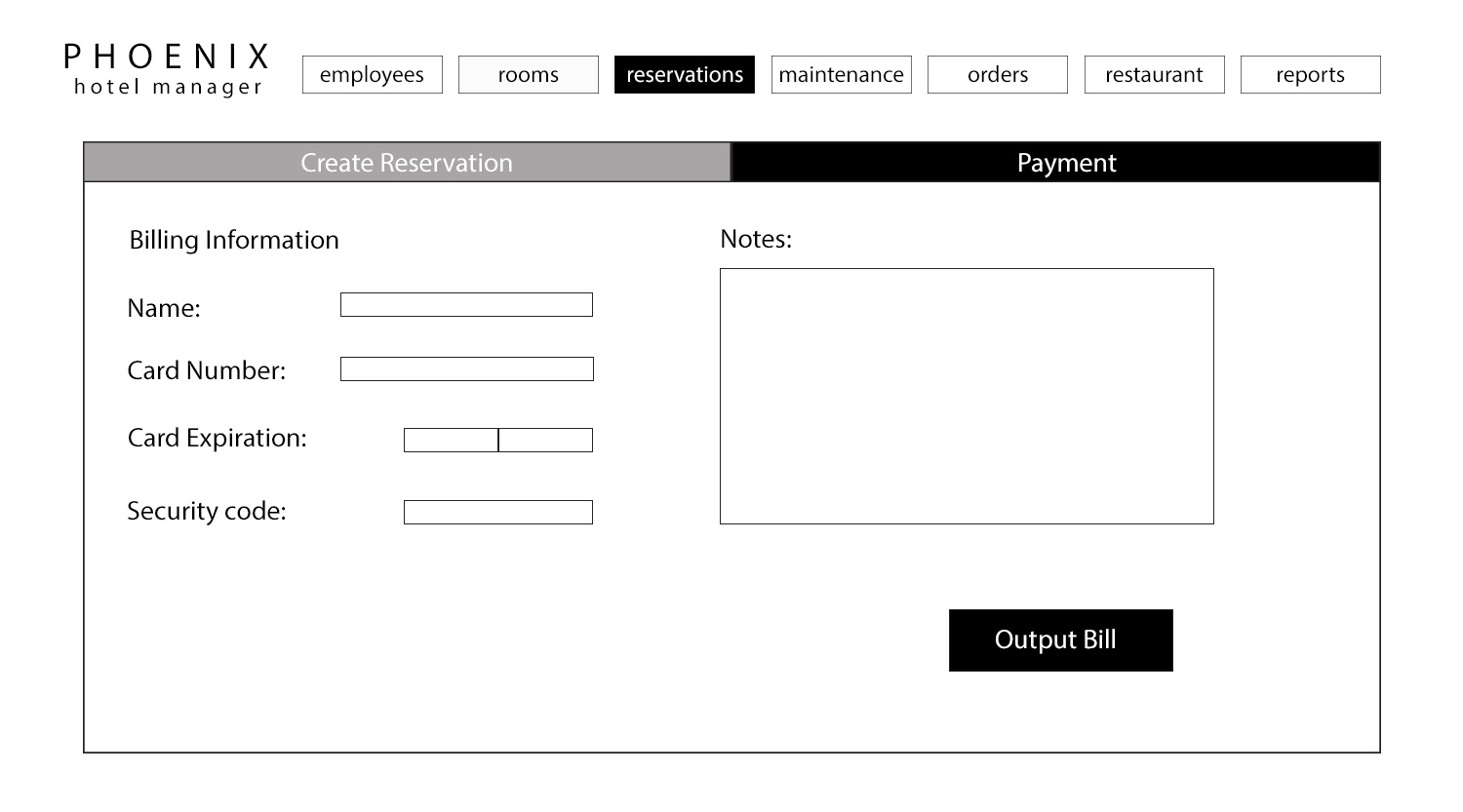
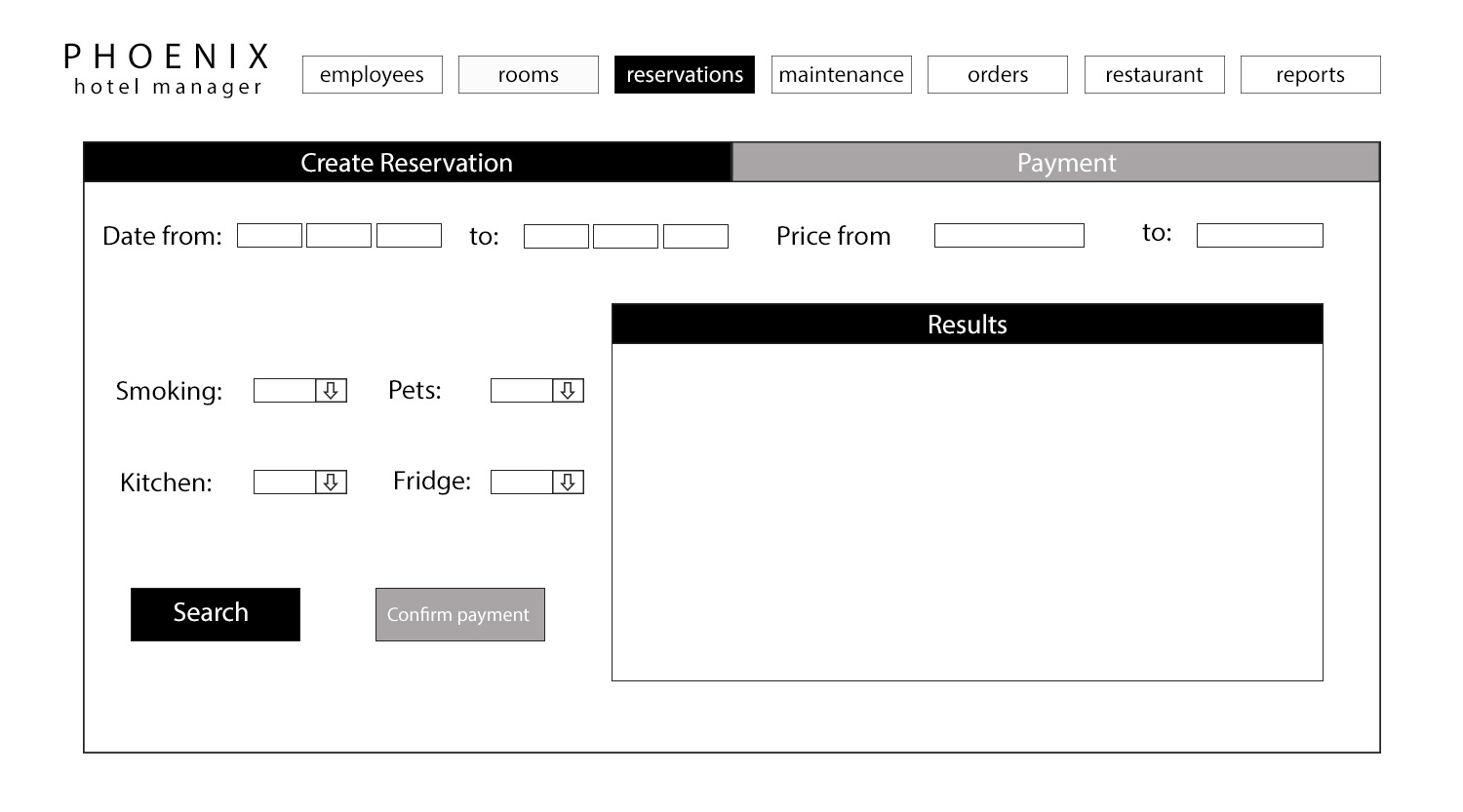
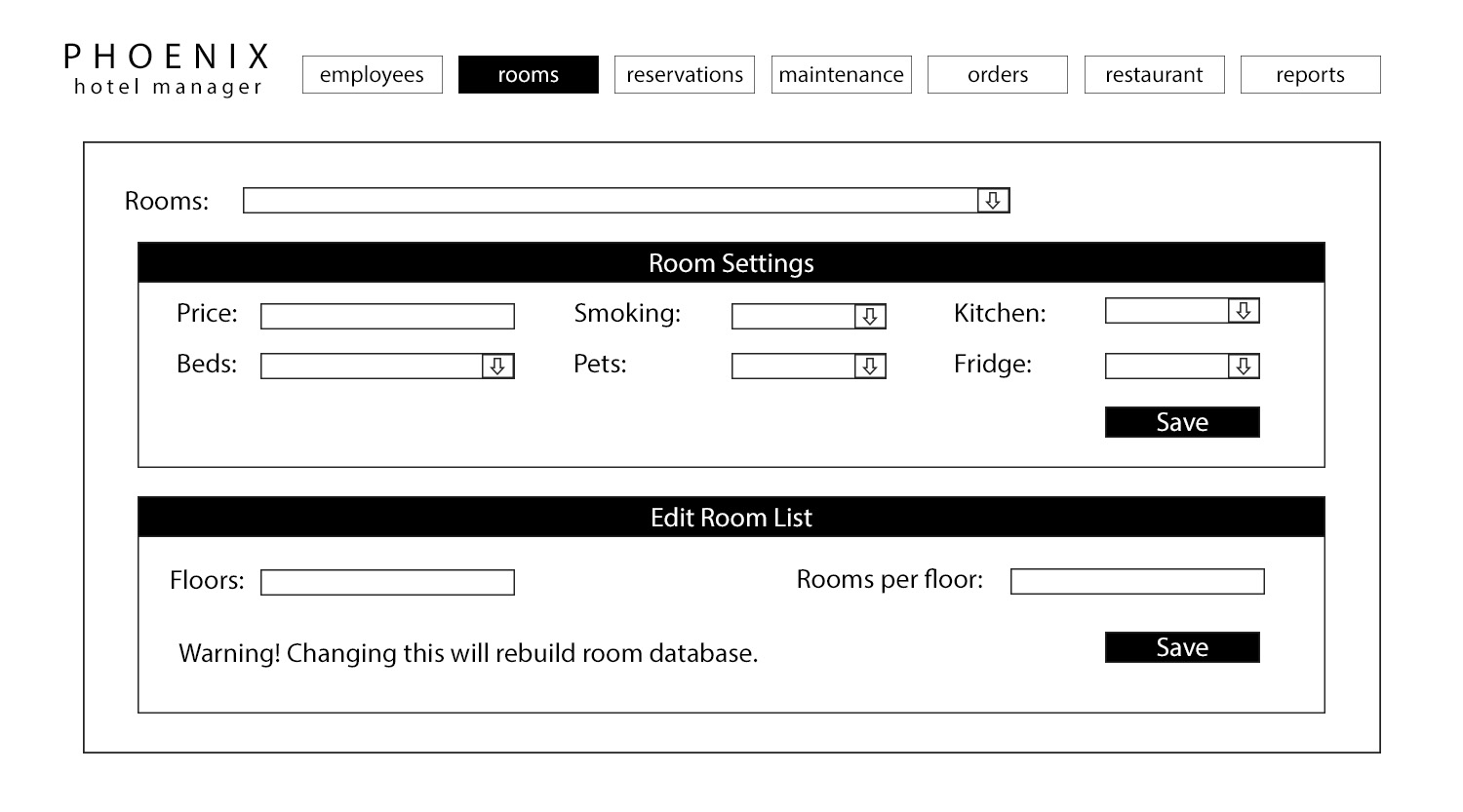
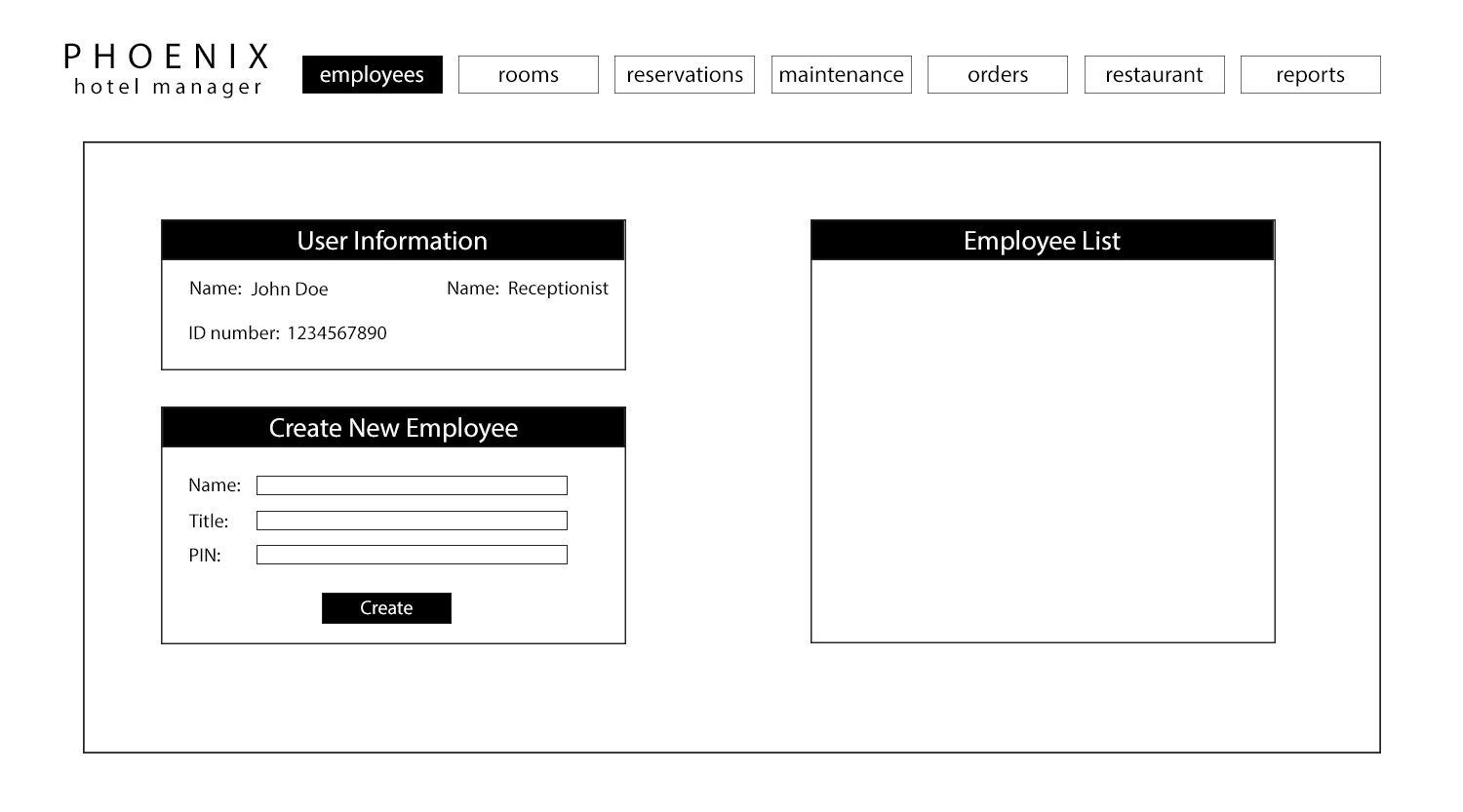
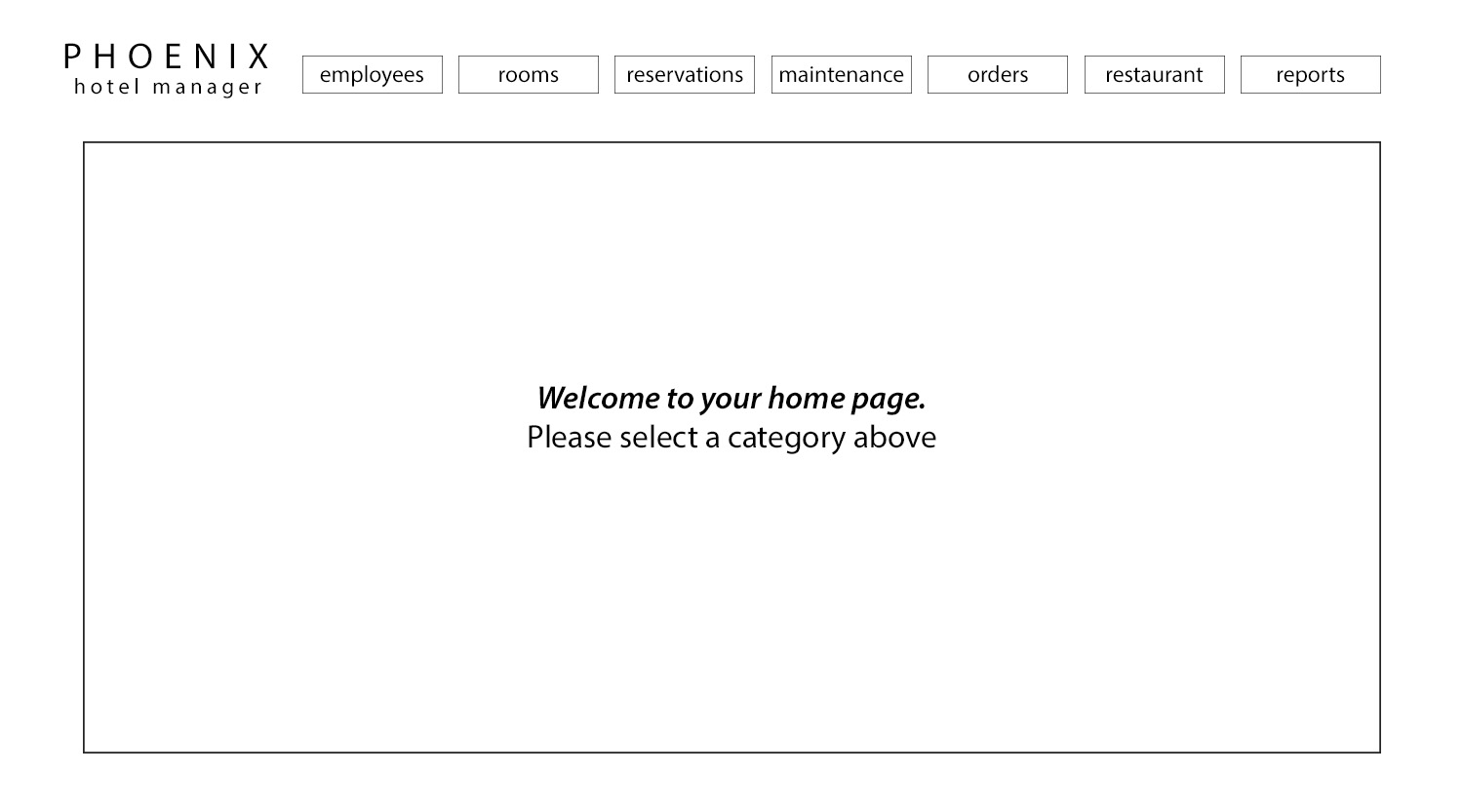
Viraj Shah

Document #3 — System Analysis and Design

September 29th, 2016

Horizontal Prototype





Requirements Traceability Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entry # | Para # | PRMS Requirements Traceability Matrix | Type | Use Case |
| 1 | 2.0 | The PRMS shall provide a graphic user interface (GUI) with tabs | SW |  |
| 2 | 2.0 | Each tab of the interface shall profile functionality for employees to interact with the PRMS | SW |  |
| 3 | 2.0 | When the employee clicks on a tab, the PRMS shall display a new interface with the appropriate functionality for that tab. | SW |  |
| 4 | 3.0 | The PRMS shall provide a tab of the interface for creating and managing employee profiles | SW |  |
| 5 | 3.0 | The employee profiles shall be stored in a local database. | SW |  |
| 6 | 3.0 | Each profile shall contain information about the employee. | SW |  |
| 7 | 3.0 | Each profile will contain information about the employee’s access to the PRMS (authentication). | NTH |  |
| 8 | 3.1 | The PRMS shall provide an interface where the new employee profiles are created. | SW |  |
| 9 | 3.1 | Each employee profile shall contain the employee’s name, the employee’s title, and the employee’s unique ID number. | SW |  |
| 10 | 3.1 | The PRMS shall allow the entry of the employee’s name and title into the new profile window. | SW | UC2 |
| 11 | 3.1 | The PRMS shall create the employee’s unique ID number with cannot be changed. | SW |  |
| 12 | 3.1 | The PRMS will provide functionality for suitably authorized employees to modify the information contained in already existing profiles. | NTH | UC2 |
| 13 | 3.2 | The PRMS will limit access by way of a 4-digit access PIN. | NTH | UC1 |
| 14 | 3.2 | The access PIN will be created when the employee profile is created. | NTH |  |
| 15 | 3.2 | The PRMS will provide functionality for suitably authorized employees to modify the access PIN contained in already existing profiles. | NTH | UC2 |
| 16 | 3.2 | The employee profile will contain information about which interface tabs are available to each employee based on the employee’s title. | NTH |  |
| 17 | 3.2 | The PRMS will log employee interactions with the PRMS in a text file. | NTH |  |
| 18 | 4.0 | The PRMS shall allow the management and reservation of the resort’s hotel rooms. | SW | UC3 |
| 19 | 4.1 | The PRMS shall provide a tab of the user interface for managing the resort’s hotel rooms. | SW | UC3 |
| 20 | 4.1 | The PRMS shall provide functionality for describing the number of floors of the hotel, and the number of rooms on each floor. | SW | UC3 |
| 21 | 4.1 | The PRMS shall assign each room a number based on which floor of the hotel the room is located. | SW |  |
| 22 | 4.1 | The PRMS shall provide functionality to enter and modify default attributes about each room. | SW | UC4 |
| 23 | 4.1 | Each room profile shall have attributes. | SW |  |
| 24 | 4.1 | The rooms inventory shall contain the following fields: Number of pillows, Number of sheets, Number of towels, Number of soaps, Number of shampoos, Number of toilet paper rolls | SW |  |
| 25 | 4.1 | The room profiles shall be stored in a local database. | SWC |  |
| 26 | 4.2 | The PRMS shall provide a tab of the user interface that provides functionality for employees to create and modify reservations for each room. | SW | UC5 |
| 27 | 4.2 | The room reservations shall be stored in a local database. | SWC |  |
| 28 | 4.2 | Each reservation shall contain attributes. | SW |  |
| 29 | 4.2 | The reservation’s billing information shall contain fields. | SW |  |
| 30 | 4.2 | The PRMS shall allow employees to query the list of rooms by the room’s attributes and whether a reservation exists for a given timespan. | SW | UC5 |
| 31 | 4.2 | The PRMS shall allow employees to create reservations for any room that does not have a reservation for a given time span. | SW | UC5 |
| 32 | 4.2 | The PRMS shall require the employee to enter valid information for all the fields in the room reservation information. | SW |  |
| 33 | 4.2 | The PRMS shall allow employees to modify a reservation’s billing information by adding or removing additional charges. | SW | UC6 |
| 34 | 4.2 | The PRMS shall allow employees to modify a reservations timespan to extend a guests stay. | SW | UC5 |
| 35 | 4.2 | The PRMS shall allow employees to output an itemized bill containing all information about the room, including the calculated cost of the reservation based on the price per night and timespan. | SW | UC6 |
| 36 | 4.3 | The PRMS shall provide a tab of the user interface that provides functionality for employees to manage room maintenance and inventory. | SW | UC7 |
| 37 | 4.3 | The PRMS shall allow the employee to note modify the number of items in the room inventory to indicate whether replacements are needed. | SW | UC7 |
| 38 | 4.3 | The PRMS shall provide an interface to note when the room has been cleaned or the inventory has been replaced and update the necessary fields in the room profile. | SW | UC7 |
| 39 | 4.3 | The PRMS shall allow the employee to output a report of all rooms which haven’t been cleaned in a given time period, rooms which have maintenance notes, or rooms with insufficient inventory. | SW | UC7 |
| 40 | 5.0 | The PRMS shall allow the management and reservation of the resort’s conference rooms. | SW |  |
| 41 | 5.1 | The PRMS shall provide a tab of the user interface for managing the resort’s conference rooms. | SW | UC8 |
| 42 | 5.1 | The PRMS shall provide functionality for describing the number of conference rooms in the resort. | SW | UC8 |
| 43 | 5.1 | Each room shall be identified by a number prefaced by the letter “C” to differentiate the conference rooms from the hotel rooms. | SW |  |
| 44 | 5.1 | The PRMS shall provide functionality to enter and modify default attributes about each conference room. | SW |  |
| 45 | 5.1 | Each conference room shall have attributes. | SW |  |
| 46 | 5.1 | The conference room profiles shall be stored in a local database. | SWC |  |
| 47 | 5.2 | The PRMS shall provide a tab of the user interface that provides functionality for employees to create and modify bookings for conference rooms. | SW | UC9, UC10 |
| 48 | 5.2 | Each conference room reservation shall contain attributes. | SW |  |
| 49 | 5.2 | The booking billing information shall contain fields. | SW |  |
| 50 | 5.2 | The PRMS shall allow employees to query the list of conference rooms by the room’s attributes and whether a booking exists for a given timespan. | SW | UC9 |
| 51 | 6.0 | The PRMS shall provide functionality for employees to manage the various services that are necessary to provide to the guests. | SW |  |
| 52 | 6.1 | The PRMS shall provide a tab of the user interface that provides functionality for employees to create and modify service orders. | SW | UC11, UC12 |
| 53 | 6.1 | Each order shall contain the attributes. | SW |  |
| 54 | 6.1 | The PRMS shall allow the employee to create a new order. | SW | UC11 |
| 55 | 6.1 | For a new order, the state of the order shall default to “ordered” and the date and time expected shall default to the current time. | SW |  |
| 56 | 6.1 | The PRMS shall ensure that the employee has selected a reservation or booking for billing and type of order. | SW |  |
| 57 | 6.1 | The PRMS shall display an interface for entering in the various data for the type of order the employee has selected | SW | UC11 |
| 58 | 6.1 | The PRMS shall allow employees to view a list of orders sorted by time expected and filterable by current state | SW | UC13 |
| 59 | 6.1 | The PRMS shall allow the employee to update the state of the order. | SW | UC12 |
| 60 | 6.1 | Once the state of the order is changed to “delivered,” the billing information of the reservation or booking associated with the order shall be updated with the name and cost of the order. | SW |  |
| 61 | 6.1 | Orders shall be stored in a local database. | SWC |  |
| 62 | 6.2 | The PRMS shall provide an interface that provides functionality for managing the hotel restaurant. | SW |  |
| 63 | 6.2 | The PRMS shall allow employees to create and modify a list of restaurant items available for room service and table service. | SW | UC14 |
| 64 | 6.2 | Each restaurant item shall contain attributes. | SW |  |
| 65 | 6.2 | The PRMS shall allow employees to create and modify a list of tables available for seating at the restaurant | SW | UC14 |
| 66 | 6.2 | Each table shall contain attributes. | SW |  |
| 67 | 6.2 | When creating an order for room service or table service, the PRMS shall require that the employee select one or more items and select a room number for billing from a list of currently reserved rooms. | SW | UC11 |
| 68 | 6.2 | If creating an order for table service, the PRMS shall require that the employee select a table number from the list of restaurant tables. | SW | UC11 |
| 69 | 6.3 | The PRMS shall allow employees to create and modify a list of options available for catered meals. | SW | UC14 |
| 70 | 6.3 | Each catered option shall contain attributes. | SW |  |
| 71 | 6.3 | When creating an order for catered meals, the PRMS shall require that the employee select a conference room booking for delivery and billing, as well as enter the number of servings required. | SW | UC11 |
| 72 | 6.4 | The PRMS shall provide an interface that provides functionality for creating general orders. | SW | UC11 |
| 73 | 6.4 | When creating a general service order, the PRMS shall provide a text box for entering specific information about the order. | SW | UC11 |
| 74 | 7.0 | The PRMS shall provide a tab of the user interface that provides functionality for employees to easily query the various databases used by the software. | SW | UC15 |
| 75 | 7.0 | The PRMS shall provide interfaces for returning information in any given timespan. | SW | UC15 |

Use Cases and Interaction Diagrams

Use Case 1: User Login

**Overview:**

The User Login menu enables users to login their credentials upon opening the software, allowing access to the main functionalities that allow them to carry out their majority work responsibilities and tasks

**Preconditions:**

1. There are no users currently logged in
2. The user’s profile information is registered into the system database
3. The PRMS\_Desktop\_View is displayed

**Scenario**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. Enter characters into user name box | 1. Username literals are displayed |
| 1. Enter characters into user pin box | 1. User pin field are displayed |
| 1. User selects the “login” button | 1. The Username and user pin field is updated in the profile checker object. Credentials in profile checker are compared to Database, and when found, PRMS\_Desktop\_View takes user to own profile page to access funcionalities |

**Scenario Notes:**

Items 1 and 2 can be done in any order. Thus Action item 3 successfully occurs after the procession of items 1 and 2

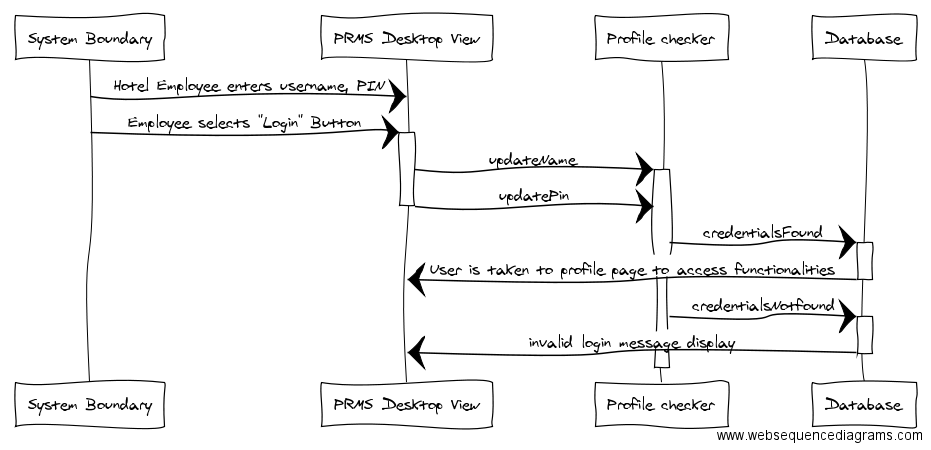
**Postconditions:**

1. User will be prompted with an error message due to an incorrect login(if “Login” button was selected)
2. User will be taken to online profile screen when credentials match(if “Login” button was selected)

|  |  |
| --- | --- |
| **Required GUI:**   1. PRMS\_Desktop\_View | **Use Cases Utilized:**  None |
| **Exceptions:**  In the case that the Database cannot be accessed, a resolution would be to copy the metadata into a master file | **Timing Constraints:**  None |

Rationale:

**Rationale:** To allow only authorized employees to access the system



Use Case 2: Create Profile

**Overview:**

Managerial and administrative users (M/A) are enabled to create profiles for new users

**Preconditions:**

1. M/A users are logged in to the main system
2. The “Create” Button is not selectable until all expected parameters have been filled
3. M/A users are on their employee profile information page of the system

**Scenario**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. M/A enters username literal | 1. Displays new username literal |
| 1. M/A enters title literal | 1. Displays new title literal |
| 1. M/A enters a 4 digit user pin literal | 1. Displays new pin literal; enables selectable “Create” button |
| 1. M/A clicks the create button | 1. New employee object is created, credentials not found, database is updated, success message appears |

**Scenario Notes:**

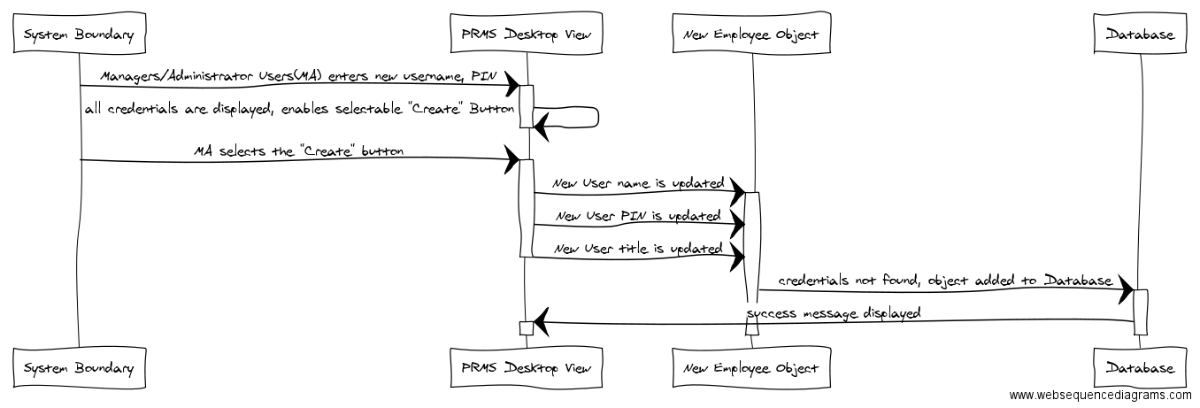
Items 1, 2, and 3 can be done in any order, but all must be initiated before proceeding to action item 4 . It is mandatory that the PIN entered in item 3 comprises of 4 integer values.

**Postconditions:**

1. Database is updated (if “Create” button was selected)

|  |  |
| --- | --- |
| **Required GUI:**  PRMS\_Desktop\_View | **Use Cases Utilized:**  None |
| **Exceptions:**   1. Clicking the Create button when all fields have not been completed. Resolved by the precondition #3 2. Entering a PIN literal that is less than or greater than 4 literals.   Resolved by the precondition #3 | **Timing Constraints:**  None |

**Rationale:** To manage employee profiles to ensure correct access and credentials

**

Use Case 3: Manage Hotel Rooms

**Overview:**

This Use Case enables the user, the employee, to manage hotel room functionality such as number of rooms and floors

**Preconditions:**

1. The user wants to manage hotel room functions
2. The database is accessible.
3. The Rooms tab is displayed.

**Scenario**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. The user clicks on a text field box for “Floors” and enter a numerical value for edit room list. | 1. The text field box will have a numerical value representing the number of floors the hotel has. |
| 1. The user clicks on a text field box for “Rooms per floor” and enter a numerical value for edit room list. | 1. The text field box will have a numerical value representing the number of rooms per floor the hotel has. |
| 1. The user clicks on save button to configure the hotel room values that were specified in the input fields. | 1. The specific hotel room values that be saved in the database. |

**Scenario Notes:**

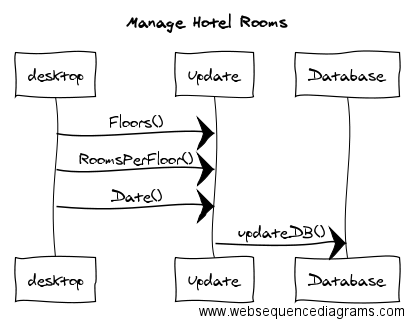
Item 1 and 2 can be done in any order, so it does not affect each other.

**Postconditions:**

1. The hotel room specifications will be saved in the database based on the specification provided.
2. The database will update to all relevant rooms that have been changed

|  |  |
| --- | --- |
| **Required GUI:**  **Mouse and keyboard** | **Use Cases Utilized:**  **none** |
| **Exceptions:**   1. The DB cannot be accessed. | **Timing Constraints:**  **none** |

**Rationale:** To provide flexibility in the number and type of rooms available. The software is designed to accommodate resorts of different sizes.



Use Case 4: Modify Room Profile

**Overview:**

This Use Case enables the user, the employee, to make manage specific hotel rooms values for the purpose of room management.

**Preconditions:**

1. The user needs to select a hotel room to modify.
2. The database is accessible.
3. The Rooms Room tab is displayed.

**Scenario**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. The user clicks on a drop box of “Rooms” and selects a room value from the drop menu. | 1. The menu item from “Rooms” will have numerical value representing the room number. |
| 1. The user enters the price for the hotel room in the price text fields | 1. The text field for “Price” will represent a numeral value representing the price of the hotel room. |
| 1. The user clicks on a drop box of “Smoking” and enter a Boolean value for the hotel reservation | 1. The drop box will have a Boolean value representing whether the room is smoking or smoke free. |
| 1. The user clicks on a drop box of “Pets” and enter a Boolean value for the hotel reservation | 1. The drop box will have a Boolean value representing whether the room accommodates pets or not. |
| 1. The user clicks on a drop box of “Beds” and enter a numerical value for the hotel reservation | 1. The drop box will have a numerical value representing the number of beds the hotel accommodates. |
| 1. The user clicks on a drop box of “Kitchen” and enter a Boolean value for the hotel reservation | 1. The drop box will have a Boolean value representing whether the room has a kitchen or not. |
| 1. The user clicks on a drop box of “Fridge” and enter a Boolean value for the hotel reservation | 1. The drop box will have a Boolean value representing whether the room has a fridge or not. |
| 1. The user clicks on save button to configure the hotel room values that were specified in the input fields. | 1. The specific hotel room values that be saved in the database. |

**Scenario Notes:**

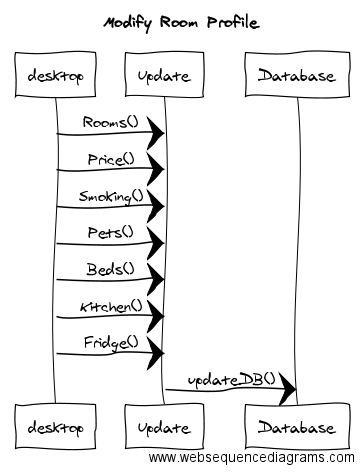
Item 1 to 7 can be done in any order, so it does not affect each other.

**Postconditions:**

1. The hotel room specifications will be saved in the database based on the specification provided.
2. The database will update to all relevant rooms that have been changed

|  |  |
| --- | --- |
| **Required GUI:**  **Mouse and keyboard** | **Use Cases Utilized:**  **None** |
| **Exceptions:**   1. The DB cannot be accessed. | **Timing Constraints:**  **None** |

**Rationale:** To provide flexibility in the types of rooms available.



Use Case 5: Create Hotel Reservation

**Overview:**

This Use Case enables the user, the employee, to make create hotel reservation functions for the customer.

**Preconditions**:

The user needs to reserve a hotel room for a customer.

The database is accessible.

The Reservation Room tab is displayed.

**Scenario**

|  |  |
| --- | --- |
| Action | Software Reaction |
| The user clicks on text field of “Date from” and “to”, and enters the dates of the requested reservation. | The two text fields of “Date from” and “to” will have numerical values representing dates. |
| The user enters the price range the customer is searching for in the price range text fields | The two text fields of the “Price Range” will represent numeral values representing the price range of available hotels |
| The user clicks on a drop box of “Smoking” and enter a Boolean value for the hotel reservation | The drop box will have a Boolean value representing whether the room is smoking or smoke free. |
| The user clicks on a drop box of “Pets” and enter a Boolean value for the hotel reservation | The drop box will have a Boolean value representing whether the room accommodates pets or not. |
| The user clicks on a drop box of “Kitchen” and enter a Boolean value for the hotel reservation | The drop box will have a Boolean value representing whether the room has a kitchen or not. |
| The user clicks on a drop box of “Fridge” and enter a Boolean value for the hotel reservation | The drop box will have a Boolean value representing whether the room has a fridge or not. |
| The user clicks on query button to search for all of the hotel room that meets the criteria. | There will be a list of available hotel room numbers that will appear on the result section. |
| The user selects any one of the hotel rooms available on the list under the result section. | The selected hotel room number will be highlighted in blue. |

**Scenario Notes:**

Item 1 to 6 can be done in any order, so it does not affect each other.

**Postconditions**:

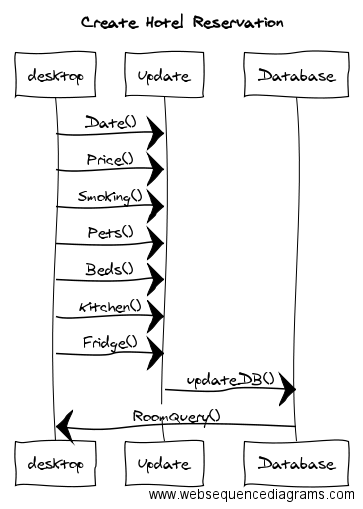
A hotel room will be reserved for the customer based on the specification provided.

The database will update to where there will be one less room.

A page will appear to let the customer input billing information.

|  |  |
| --- | --- |
| **Required GUI:**  Mouse and keyboard | **Use Cases Utilized:**  None |
| **Exceptions:**  The DB cannot be accessed.  There are no available conference rooms to be reserved. | **Timing Constraints:**  none |

**Rationale:** It is crucial to the operation of a hotel or resort to be able to efficiently reserve rooms for guests.



Use Case 6: Modify or Output Room Billing

**Overview:**

This Use Case enables the user, the employee, to modify or output room billing for reservations that have been placed.

**Preconditions:**

1. The user needs to select the reserved room for billing
2. The database is accessible.
3. The Reservation Room tab is displayed.

**Scenario**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. The user will go to the payment info section, click on the text field of name and enter the name of the customer. | 1. The text field of name will have the name of the customer represented by letters. |
| 1. The user will click on the text field of the credit card number and credit card expiration date. The user will enter the information of the customer credit card. | 1. The text field of credit card number and expiration date will have credit card information of the customer in numerical value. |
| 1. The user will click on the text field of the security code. The user will enter the information of the security code from the customer credit card. | 1. The text field of the security code will have security information of the customer as a numerical value. |
| 1. The user will click on the text field of Notes. The user will enter the additional information regarding billing. | 1. The text field of Notes will have any additional information represented by letters. |
| 1. The user will click on the Output Bill button after all of the information is filled. | 1. It will send the customer to a page that tells them the billing information. |

**Scenario Notes:**

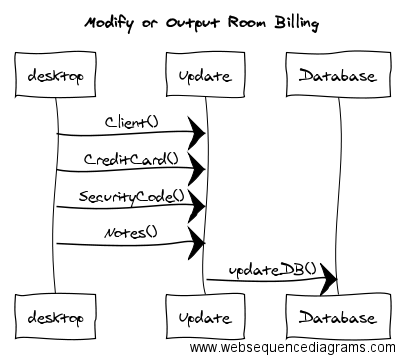
Item 1 to 4 can be done in any order, so it does not affect each other.

**Postconditions:**

1. The hotel room billing information will be saved for the customer based on the specification provided.
2. The database will update the relevant billing information.
3. A page will appear to let customer have information of their billing information room.

|  |  |
| --- | --- |
| **Required GUI:**  **Mouse and keyboard** | **Use Cases Utilized:**  **None** |
| **Exceptions:**   1. The DB cannot be accessed. 2. There are no available conference rooms to be reserved. | **Timing Constraints:**  **None** |

**Rationale:** Accurate billing is necessary for the function of any business



Use Case 7: Manage Room Maintenance

**Overview:**

This Use Case enables the user, the employee, to manage hotel room maintenance for the hotel.

**Preconditions:**

1. The user needs to select a hotel room to modify.
2. The database is accessible.
3. The Maintenance Room tab is displayed.

**Scenario**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. The user clicks on a text field box of “Room” and types in a room value. | 1. The value from “Room” will be a numerical value representing the room number. |
| 1. The user clicks on a drop box of “Maintenance Category” and selects a category from the drop down box for maintenance. | 1. The drop box item for “Maintenance Category” will be represented by letters. |
| 1. The user clicks on a text field box of “Last Checked” and enter a date value for the hotel maintenance | 1. The text field box will have a date value representing when the room was last checked. |
| 1. The user clicks on Set Notice button to configure the maintenance request that was specified in the input fields. | 1. The hotel maintenance order will be saved in the database. |
| 1. The user clicks on Order Item button to configure the maintenance request. | 1. The selected item will be placed for order. |

**Scenario Notes:**

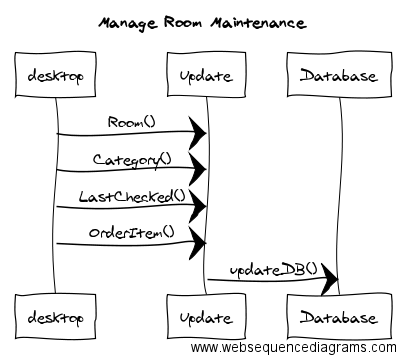
Item 1 to 3 can be done in any order, so it does not affect each other.

**Postconditions:**

1. The hotel maintenance order will be saved in the database based on the specification provided.
2. The database will update to all relevant orders that have been changed

|  |  |
| --- | --- |
| **Required GUI:**  **Mouse and keyboard** | **Use Cases Utilized:**  **None** |
| **Exceptions:**   1. The DB cannot be accessed. | **Timing Constraints:**  **none** |

**Rationale:** Employees must be access and update information about the maintenance of each room to ensure quality for guests.



Use Case 8: Create Conference Room Profiles

**Overview:**

Managers and Administrator users (M/A) can reserve available conference rooms for a certain amount of time at a certain cost. This functionality is accessible only to Manager and Administrator users.

**Preconditions:**

1. The PRMS\_Desktop\_View is displayed
2. (M/A) user is logged in into system
3. The user has clicked on the conference room tab in the interface

**Scenarior**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. User enters literal for name of the conference room | 1. Croom name field is updated |
| 1. User enters value for number of seats the conference room contains | 1. Croom seats field is updated |
| 1. User enters value for cost of reserving the conference room(s) per day | 1. Croom rate field is updated |
| 1. User marks a checkbox that determines whether or not conference room will have a stage | 1. Croom hasStage field is updated |
| 1. User marks a checkbox that determines whether or not conference room will have an audio/visual system | 1. Croom hasAV field is updated |
| 1. User clicks the “Create” button | 1. The conference room object is created, database is updated. The Croom\_created\_Pop-up\_View is created and appears |
| 1. User clicks the “Ok” button | 1. The Croom\_created\_Pop-up\_View is destroyed |

**Scenario Notes:**

Items 1,2,3,4, and 5 can be done in any order, as long as all action items are initiated in entirety before proceeding to item 6. Item 7 refers to the display of the options in The Croom\_created\_Pop-up\_View.

**Postconditions:**

1. Database is updated(if “Create” button was selected)
2. The M/A is returned to the PRMS\_Desktop\_View(if the “Ok” button was selected from Croom\_created\_Pop-up\_View)

|  |  |
| --- | --- |
| **Required GUI:**  PRMS\_Desktop\_View  Croom\_created\_Pop-up\_View | **Use Cases Utilized:**  none |
| **Exceptions:**  In the case that the Database cannot be accessed, a resolution would be to copy the metadata into a master file | **Timing Constraints:**  none |

**Rationale:** This software is designed to accommodate hotels or resorts of different sizes. This feature allows the manager to adapt the software to their needs.

Use Case 9: Conference Booking

**Overview:**

This Use Case enables the user, the employee, to make reservations for a conference room, requested by the customer, for a certain time and day.

**Preconditions:**

1. The user needs to reserve a conference room for a customer.
2. The database is accessible.
3. The conference tab is displayed.

**Scenario**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. The user clicks on text field of “Date from” and “to”, and enters the dates of the requested reservation. | 1. The two text fields of “Date from” and “to” will have numerical values representing dates. |
| 1. The user clicks on a drop box of “Number of people” and enter the number of people that will be in the conference room. | 1. The drop box will have a numerical value representing the number of people that will be in the conference room. |
| 1. The user clicks on a text field of “from” and “to” and enter the time they want to reserve. | 1. The two text fields will have numerical value representing the time they will reserve the conference room. |
| 1. The user clicks on check boxes of a list of miscellaneous requests that they want to be provided in their conference room. | 1. A list of miscellaneous requests will have some of its check boxes marked. |
| 1. The user clicks on query button to search for all of the conference room that meets the criteria. | 1. There will be a list of available conference room numbers that will appear on the result section. |
| 1. The user selects any one of the conference rooms available on the list under the result section. | 1. The selected conference room number will be highlighted in blue. |
| 1. The user will go to the billing info section, click on the text field of name and enter the name of the customer. | 1. The text field of name will have the name of the customer represented by letters. |
| 1. The user will click on the text field of the credit card number and credit card expiration date. The user will enter the information of the customer credit card. | 1. The text field of credit card number and expiration date will have credit card information of the customer in numerical value. |
| 1. The user will click on the reserve button after all of the information is filled. | 1. It will send the customer to a page that tells them the information of their conference room. |

**Scenario Notes:**

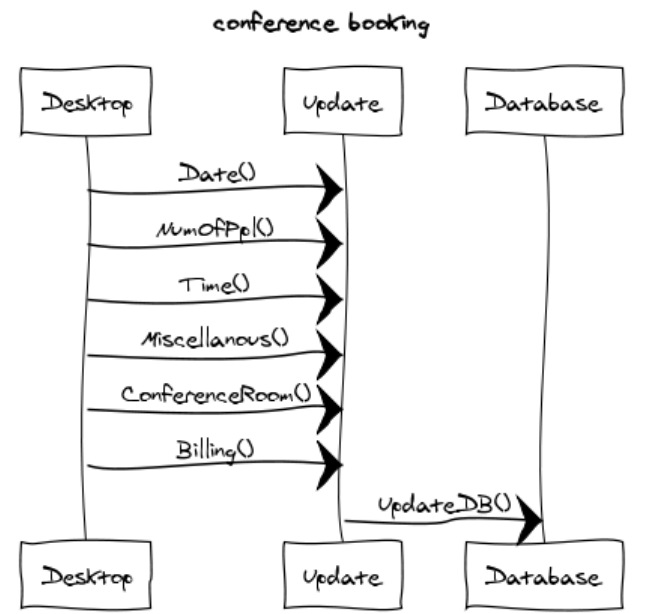
Item 1 to 4 can be done in any order, so it does not affect each other.

**Postconditions:**

1. A conference room will be reserved for the customer based on the specification provided.
2. The database will update to where there will be one less room.
3. A page will appear to let customer have information of their reserved conference room.

|  |  |
| --- | --- |
| **Required GUI:**  **Mouse and keyboard** | **Use Cases Utilized:**  **none** |
| **Exceptions:**   1. The DB cannot be accessed. 2. There are no available conference rooms to be reserved. | **Timing Constraints:**  **none** |

**Rationale:** It is necessary for conference rooms to be easily booked.



Use Case 10: Conference Billing

**Overview:**

This Use Case enables the users or employees to modify or output the conference room billing.

**Preconditions:**

1. The user needs to modify the information for of the conference room billing.
2. The user needs to send out the conference rooming billing to the customers.
3. There must be a reservation for the conference room made.
4. The conference tab is displayed.

**Scenario**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. The user will go to the payment section and clicks on the Reservation drop box and selects the current reservation. | 1. The list of current reserved rooms will drop into a list and a room will be highlighted in blue to show it is selected. |
| 1. The user will select the Payment Amount text field and enter the amount owed by customer. | 1. The text field will have numerical value representing the amount of money the customers owe. |
| 1. The user may add any notes on the Note section of the text field. | 1. The text field of the note section will be filled with numbers and letters delivering a message to the customer. |
| 1. The user clicks apply button to let the changes be permanent. | 1. The Apply button will be pressed and the information will be saved. |
| 1. The user will go to the charges section and click on reservation drop box and selects the current reservation. | 1. The list of current reserved rooms will drop into a list and a room will be highlighted in blue to show it is selected. |
| 1. The user will select the Charge Amount text field and enter the amount owed by customer. | 1. The text field will have numerical value representing the amount of money the customers owe. |
| 1. The user may add any notes on the Note section of the text field. | 1. The text field of the note section will be filled with numbers and letters delivering a message to the customer. |
| 1. The user clicks apply button to let the changes be permanent. | 1. The Apply button will be pressed and the information will be saved. |
| 1. The user clicks on the Output Bill to send to customers. | 1. It pops up wind of itemized bill. |

**Scenario Notes:**

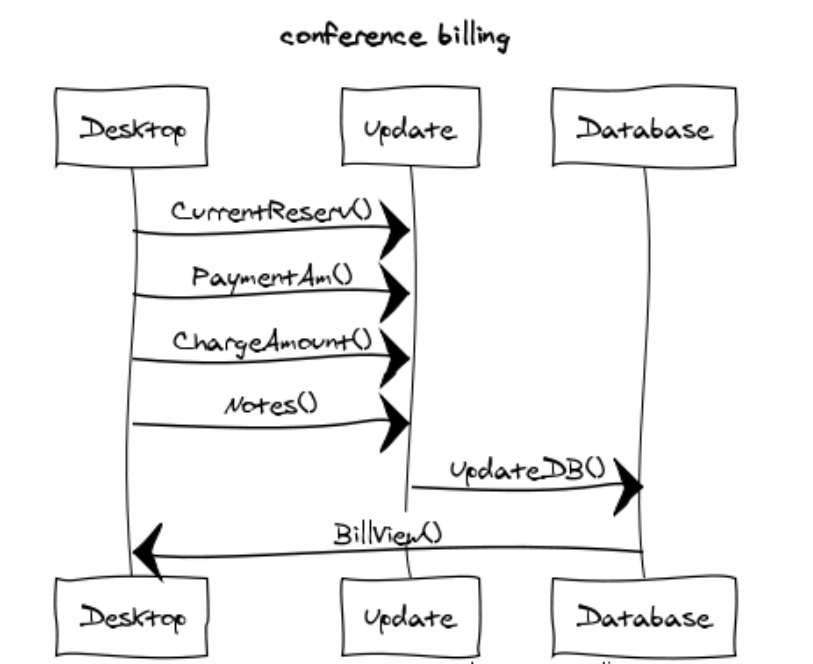
The user could have done item 1 to 4 before or after item 5 to 8.

**Postconditions:**

1. The payment information of the customer will be modified or updated.
2. The bill of customers for the conference room will be sent to the customers.

|  |  |
| --- | --- |
| **Required GUI:**  Mouse and key | **Use Cases Utilized:**  **None** |
| **Exceptions:**  **none** | **Timing Constraints:**  **none** |

**Rationale:** Conference rooms must be able to be billed effectively.



Use Case 11: Unified Ordering System

**Overview:**

The use case is established such that actors can request and order various services through the end-users. The end-users, hotel employees, respectively work with an ordering system functionality that allows them to document the requests and services in one space

**Preconditions:**

1. The “Create Order” button is not selectable until the expected parameters have been filled
2. The user is logged into the system
3. The user’s profile information is registered into the system database
4. The PRMS\_Desktop\_View is displayed

**Scenario**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. User enters name of room that requested service | 1. New Room order object created. Room order field is updated |
| 1. Users select option from the “order type” dropdown menu | 1. Order type field is updated |
| 1. Users select option from the “order state” dropdown menu | 1. Order state field is updated |
| 1. User inputs values for date and time | 1. Time field is updated |
| 1. User selects whether order is a reservation or booking for billing data | 1. Billing type field is updated |
| 1. User selects the “Create Order” button | 1. New order object added to database. Success message displayed |

**Scenario Notes:**

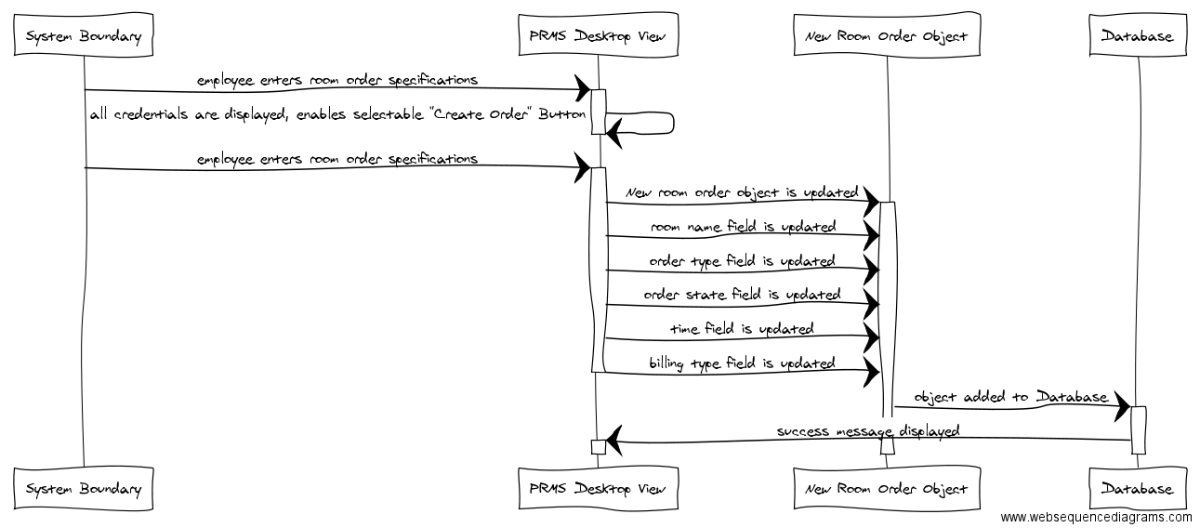
Action items 1,2,3,4, and 5 can be done in any order, as long as all action items are done completely before proceeding to action item 6.

**Postconditions:**

1. Room order field is updated(if “Create order” button was selected)
2. Order type field is updated(if “Create order” button was selected)
3. Order state field is updated (if “Create order” button was selected)
4. Time field is updated(if “Create order” button was selected)
5. Billing type field is updated (if “Create order” button was selected)
6. Order request object is added to database(if “Create order” button was selected)
7. Order\_Made\_Pop-up is created(if “Create Order” button was created)

|  |  |
| --- | --- |
| **Required GUI:**  PRMS\_Desktop\_View | **Use Cases Utilized:**  None |
| **Exceptions:**  None | **Timing Constraints:**  None |

**Rationale:** There are many customer service needs in a resort. This feature is a unified system that allows employees to provide specific orders for guests.

****

Use Case 12: Modify or Update Order

**Overview:** The User is able to look up and manage orders.

**Preconditions:**   
1. The User has to be logged in as manager.

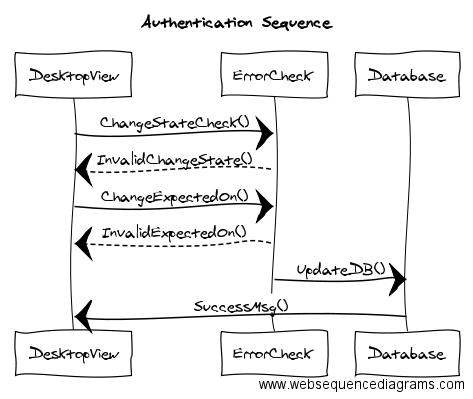
**Scenario**

|  |  |
| --- | --- |
| Action | Software Reaction |
| User clicks the Manage Orders tab in the Orders page | A “change state” dropdown menu ,an “change expected date” field, and an inactive “submit changes” button appear |
| Scenario: User changes state |  |
| User clicks on the relevant order in the Order List | Order becomes selected |
| User clicks on a new state from the “change state” dropdown menu: “ordered”, “preparing”, or “completed” | New state becomes selected |
| User chooses “completed” | the “change expected date” field becomes mandatory and now reads, “delivered on” |
| User fills the “delivered on” field | The date is checked. If User provided a date not later than today, “Submit changes” button becomes active. If the User provided a date later than today, an error message appears. |
| User chooses “ordered” or “preparing” | The “submit changes” button becomes active. |
| User clicks “submit changes” | The new state of the selected item becomes updated in the database, and a success message appears. |
| Scenario: User changes expected date |  |
| User clicks on the relevant order in the Order List | Order becomes selected |
| User changes the date within the “change expected date” field and clicks “submit changes” | The date is checked. If User provided a date not earlier than today, “Submit changes” button becomes active. If the User provided a date earlier than today, an error message appears. |
| User clicks "submit changes” | The new expected date of the selected item becomes updated in the database, and a success message appears. |

**Scenario Notes:** The User can change both state and “expected on” date before clicking “submit changes” instead of doing it one by one.

**Postconditions:** If changes are made, they are applied to the database.

|  |  |
| --- | --- |
| **Required GUI:**  **Keyboard and Mouse** | **Use Cases Utilized:**  **None** |
| **Exceptions:**  **None** | **Timing Constraints:**  **None** |

**Rationale:** I****t is essential that orders can be tracked and updated as necessary.

Use Case 13: Output Orders

**Overview:**

This Use Case enables the users or employees to output orders.

**Preconditions:**

1.The user needs to have an order that already have orders created and modified.

**Scenario**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. The user clicks on the on the Output bottom after the modifying section of the order. | 1. The order will be in a pop up window. |

**Scenario Notes:**

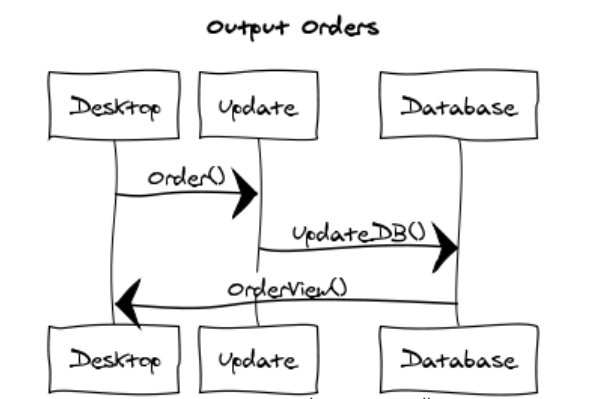
Item 1 can only be performed when an order is created or modified.

**Postconditions:**

The itemized order will be in a pop up window and sent to customers.

|  |  |
| --- | --- |
| **Required GUI:**  **Mouse and keyboard** | **Use Cases Utilized:**  **None** |
| **Exceptions:**  **None** | **Timing Constraints:**  **None** |

**Rationale:** Employees must be able to track the status and quantity of orders in order to do their jobs effectively.



Use Case 14: Manage Restaurant

**Overview:** TheUser is able to add, remove, or edit items on the restaurant menu.

**Preconditions:**   
1. User has logged in as manager

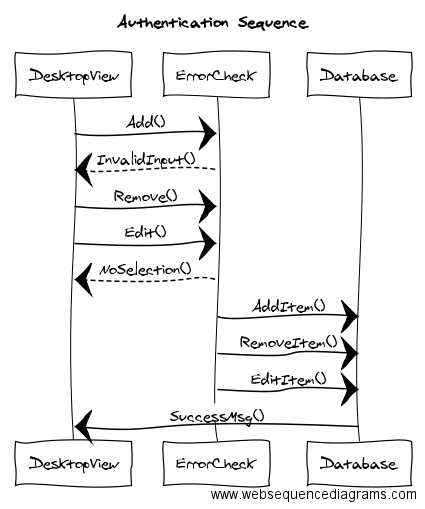
**Scenario**

|  |  |
| --- | --- |
| **Action** | **Software Reaction** |
| Scenario: Add Menu Item |  |
| User Clicks the “Add” button | The text fields are in Add Item mode |
| User fills in the name [required], cost [required], and description [optional] fields, and clicks the “save” button | The information is saved in the database and a success message is displayed. |
| User enters name but doesn’t enter cost, or vice versa | No changes to the database are applied and an error message appears |
| User doesn’t enter neither name nor cost | No changes to the database are applied and an error message appears |
| Scenario: Remove Item |  |
| User clicks on an existing menu item | The item becomes selected. |
| With an item selected, the user clicks the “delete” button | The relevant item is removed from the database and a success message is displayed. |
| Scenario: Edit Item |  |
| The user clicks the “edit” button which is beside each menu item | The name, cost, and description fields become editable for that item. |
| The user makes the relevant changes and clicks the “save” button | The changes are applied to the item in the database and a success message is displayed. |

**Scenario Notes:** If no item is selected, clicking the “save” or “delete” buttons will have no effect.

**Postconditions:** All relevant modifications, if any, are applied to the database.

|  |  |
| --- | --- |
| **Required GUI:**  **Keyboard and Mouse** | **Use Cases Utilized:**  **None** |
| **Exceptions:**  **None** | **Timing Constraints:**  **None** |

**Rationale:** A restaurant must be able to change and update its menu according to customer preferences and business needs 

Use Case 15: Generate Reports

**Overview:**

This Use Case enables the user, the employee, to generate reports for customers.

**Preconditions:**

1. The user needs to select a customer in order to generate a relevant report.
2. The database is accessible.
3. The Report tab is displayed.

**Scenario**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. The user clicks on a text field box of “Client” and types in the customer name. | 1. The value from “Client” will be represented by letters unique to the customer name. |
| 1. The user clicks on a text field box of “Room” and types in a room value. | 1. The value from “Room” will be a numerical value representing the room number. |
| 1. The user clicks on text field of “Time period” and enters the dates of the requested report. | 1. The two text fields of “Time period” will have numerical values representing dates. |
| 1. The user clicks on Generate Report button to configure the maintenance request that was specified in the input fields. | 1. The report order will be queried from the database and displayed to the user. |

**Scenario Notes:**

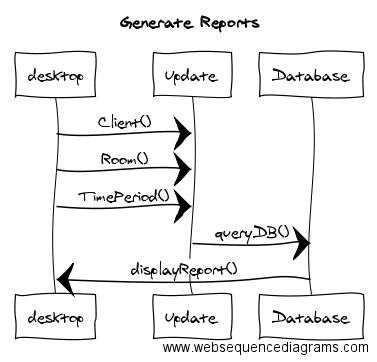
Item 1 to 3 can be done in any order, so it does not affect each other.

**Postconditions:**

1. The report order will be queried in the database based on the specification provided.
2. The database will update to all relevant orders that have been changed

|  |  |
| --- | --- |
| **Required GUI:**  Mouse and keyboard | **Use Cases Utilized:**  **None** |
| **Exceptions:**   1. The DB cannot be accessed. | **Timing Constraints:**  **none** |

**Rationale:** For a business to be efficient and profitable, real time information is crucial.



Function Point Cost Analysis

Weighting Factor Estimation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Measurement parameter** | **count** |  | **average** |  |  |
| Number of user inputs | 38 | × | 4 | = | 152 |
| Number of user outputs | 14 | × | 5 | = | 70 |
| Number of user inquires | 10 | × | 4 | = | 40 |
| Number of files | 8 | × | 10 | = | 80 |
| Number of external interfaces | 14 | × | 7 | = | 98 |
|  |  |  |  | **Total =** | **440** |

Complexity Adjustment Factor

|  |  |
| --- | --- |
| **Factor** | **Value** |
| Backup & recovery | 2 |
| Data communications | 2 |
| Distributed processing | 0 |
| Performance critical | 2 |
| Existing operating environment | 5 |
| Online data entry | 5 |
| Input transaction over multiple screens | 4 |
| Master files updated online | 3 |
| Information domain values complex | 2 |
| Internal processing complex | 1 |
| Code designed for reuse | 3 |
| Conversion/installation in design | 0 |
| Multiple installations | 5 |
| Application designed for change | 5 |
| Complexity adjustment factor | 1.17 |
| **Total** | **40.17** |

Function Point Calculation

**FP =** 440 × (0.65 + 0.01 × 40.17) = 440 × 1.0517 = 462.748

At 6.5 FP per month and $8000.00 per month, the total cost of the software would be

( 463 / 6.5 ) \* 8000 = **$569,536**

Database

We plan to use SQLite as an embedded database in our application.

Work Schedule Diagram

Gantt Chart

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Start Date:  August 22, 2016** | **Wk 1** | |  |  |  | **Wk 2** | |  |  |  | **Wk 3** | |  |  |  | **Wk 4** | |  |  |  | **Wk 5** | |  |  |  | **Wk 6** | |  |  |  |
|  | **M** | **T** | **W** | **T** | **F** | **M** | **T** | **W** | **T** | **F** | **M** | **T** | **W** | **T** | **F** | **M** | **T** | **W** | **T** | **F** | **M** | **T** | **W** | **T** | **F** | **M** | **T** | **W** | **T** | **F** |
| **Document 1** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Fergus |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Andrew |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Ryan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Devidas |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Viraj |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Document 2** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Fergus |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Andrew |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Ryan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Devidas |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Viraj |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Document 3** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Fergus: Gantt chart, FP, RTM |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Andrew: Use cases and diagrams |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Ryan: Use cases and diagrams, horizontal prototype |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Devidas: Use cases and diagrams |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Viraj: Use cases and diagrams |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Dictionary

*Embedded database*: a database that is used within a single application. This is opposed to a client-server database which involve two different software applications communicating.

*Graphical User Interface* (abbreviated *GUI*): A program that allows the user to interact with the computer using icons and other visual indicators.

*Local database*: a collection of digital indexed information that can be searched, referenced, changed, compared or otherwise manipulated. A local database is stored on the same computer that is used to access it.

Change Log

**Date and Time:** September 21, 2016- 4:13 PM

**Team member:** Ryan Ocampo

**Description:** Created Use Case documentation template

**Version:** 1 — Current version accumulating changes

**Date and Time:** September 21, 2016 — 4:27 PM

**Team member:** Ryan Ocampo

**Description:** Initial creation of document

**Version:** 1 — Current version accumulating changes

**Date and Time:** September 25, 2016 — 1:22 PM

**Team member:** Fergus Kelley

**Description:** Added database section and updated WSD

**Version:** 1 — Current version accumulating changes

**Date and Time:** September 25, 2016 —2:01 PM

**Team member:** Fergus Kelley

**Description:** Added *embedded database* to the dictionary section.

**Version:** 1 — Current version accumulating changes

**Date and Time:** September 26, 2016 — 4:45 PM

**Team member:** Ryan Ocampo

**Description:** Added use cases 1, 2, 8, 11

**Version:** 1 — Current version accumulating changes

**Date and Time:** September 26, 2016 — 5:01 PM

**Team member:** Viraj Shah

**Description:** Added use cases 3, 4, 5, 6, 7, 15

**Version:** 1 — Current version accumulating changes

**Date and Time:** September 26, 2016 — 5:21 PM

**Team member:** Devidas Rutkauskas

**Description:** Added use cases 12, 14

**Version:** 1 — Current version accumulating changes

**Date and Time:** September 26, 2016 — 5:43 PM

**Team member:** Andrew Truong

**Description:** Added use cases 9, 10, 13

**Version:** 1 — Current version accumulating changes

**Date and Time:** September 27, 2016 — 4:47 PM

**Team member:** Ryan Ocampo

**Description:** Added interaction diagrams for use cases 1, 2, 8, 11

**Version:** 1 — Current version accumulating changes

**Date and Time:** September 26, 2016 — 5:25 PM

**Team member:** Devidas Rutkauskas

**Description:** Added interaction diagrams for use cases 12, 14

**Version:** 1 — Current version accumulating changes

**Date and Time:** September 26, 2016 — 5:54 PM

**Team member:** Andrew Truong

**Description:** Added interaction diagrams for use cases 9, 10, 13

**Version:** 1 — Current version accumulating changes

**Date and Time:** September 26, 2016 — 6:20 PM

**Team member:** Viraj Shah

**Description:** Added interaction diagrams for use cases 3, 4, 5, 6, 7, 15

**Version:** 1 — Current version accumulating changes

**Date and Time:** September 27, 2016 — 9:55 PM

**Team member:** Fergus Kelley

**Description:** Added rationale for each use case

**Version:** 1 — Current version accumulating changes

**Date and Time:** September 28, 2016 — 11:05 PM

**Team member:** Fergus Kelley

**Description:** Added updated Gantt chart

**Version:** 1 — Current version accumulating changes

**Date and Time:** September 28, 2016 — 11:45 PM

**Team member:** Fergus Kelley

**Description:** Replaced Gantt chart with better design

**Version:** 1 — Current version accumulating changes

**Date and Time:** September 29, 2016 — 10:45 AM

**Team member:** Fergus Kelley

**Description:** Fixed some typography

**Version:** 1 — Current version accumulating changes